

**Machine Learning over Thoroughly Unstructured Data**

by

M. Hidayath Ansari

A dissertation submitted in partial fulfillment of  
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN-MADISON

2015

Date of final oral examination: 08/10/2015

The dissertation is approved by the following members of the Final Oral Committee:

Michael H. Coen, Assistant Professor, Biostatistics and Medical Informatics

C. David Page, Professor, Biostatistics and Medical Informatics

Barbara B. Bendlin, Assistant Professor, Medicine

Charles R. Dyer, Professor, Computer Sciences

Jude W. Shavlik, Professor, Computer Sciences

© Copyright by M. Hidayath Ansari 2015  
All Rights Reserved

*To my mother Masarrath and her father Mujtaba.*

## ACKNOWLEDGMENTS

---

*Al-ḥamdu li-L-lāhi wa kafā  
wa aṣ-ṣalātu wa as-salāmu ‘alā anbiyāi-L-lāhi al-hudāti al-muhtadīn.*

*The ultimate gratitude and praise is due to the Creator alone.*

*Peace be upon all the Messengers.*

I am extremely fortunate to have been able to study with wonderful learned people gifted with the ability to teach. From proving basic geometry theorems with Mr. Hovde in 8th grade to learning applied calculus with Mr. Surendranath to functional programming in Scheme with Prof. Sanyal at IIT to thinking algorithmically with Prof. Diwan, and with other professors too many to name, I’ve grown to appreciate the fundamental beauty of mathematics in general and discrete structures in particular. Later in graduate school, when I saw how the same ideas could be used to find patterns in all kinds of real world data, I was even more fascinated.

The seeds for this work grew out of early conversations with my advisor Prof. Michael Coen. Over the years as we worked on other projects I learned from him valuable lessons on how to decide, frame, break down, and attack research questions. I will always appreciate the amount of time he spent with me in my years in graduate school, always available for a chat next door about any topic under the sun, and even getting down in the trenches and coding with me. Among the many things I’ve learnt, some stand out because I was so severely lacking in them when starting out. These include presentation skills, both oral and in writing, and the value of good, simple, and minimal examples in communicating complex ideas. I also learned from him how to think outside my box more broadly and how to view research within a wider and larger context of applicability. His high standards and critical eye made me push myself harder and contributed significantly to my maturity. I appreciate and acknowledge the confidence Prof. Coen and the department have had in me.

Prof. Barbara Bendlin was a terrific resource when I started to work on neuroscience problems. She dedicated so much of her time to discussing the area with me and ramping me up to a brand new area of research, even in the

early months when research output was not exciting yet. I had so many basic questions, many probably repeated, which she always patiently went over with me. Suffice it to say that a large portion of this thesis would not have been possible without her and I am immensely grateful to her. To members of her lab, Alex Birdsill, Jennifer Oh, and Chuck Illingsworth especially, many thanks for helping me with data access and processing.

I would like to thank my committee members Prof. David Page, Prof. Jude Shavlik, and Prof. Chuck Dyer for their time, encouragement, and insightful conversations. Thanks also to Kevin Eliceiri and LOCI for taking me on as an RA for my first year of graduate school. Curtis Rueden and Mike Smith were very helpful throughout my time there. I am sincerely grateful to the University and State of Wisconsin, as well as the people of this State, for supporting my graduate studies.

At UW-Madison I would like to thank my officemates over the years for their help, company, and conversation: Yue Pan, Bess Berg, Tim Chang, Nathanael Fillmore, Marissa Phillips, and the unknown benefactor of the couch upon which I spent many a night. Nate and I were very productive together and he and I had many stimulating conversations. Even though we met infrequently, it was always good to catch up with Chaitanya. Special thanks to Siti and Nate for proofreading my thesis and offering comprehensive feedback.

I was very fortunate to make very close friends during my stay in Madison: Tarek, Moeed, Sabih, Abdul Aziz, Faizan, Imran, and Adnan in particular whom I spent a lot of time with. I've learned a lot from Imam Alhagie and am grateful for the time I was able to spend with him peppering him with questions.

My foundational computer science skills are the result of a rigorous curriculum taught by top-notch faculty at IIT Bombay in India. It was a pleasure to learn from and interact with them. An IIT education is truly unique and I am grateful to the government and taxpayers of India for having had access to it. There too I was very fortunate to make a set of close friends to weather the at times intense pressures of projects, exams, and assignments.

Last and most certainly not least, my family has been extremely supportive throughout my academic pursuits. My mother who sacrificed so much of her life so that I and my siblings would get a good education deserves special

mention. My late grandfather taught me integrity by living it. My uncles and aunt — Khalil, Jaleel, and Tooba — and their families always had a warm and welcoming home for me whenever I needed a change in environment.

CONTENTS

---

Contents v

List of Tables viii

List of Figures x

Abstract xiii

**1 Introduction 1**

- 1.1 *Applications and Results* 4
- 1.2 *The Point Set Representation* 8
- 1.3 *The Value of Spatial Information* 11
- 1.4 *Thesis Roadmap* 13

**2 Point Sets 15**

- 2.1 *Representation and Definitions* 16
- 2.2 *Spatial Overlap as Similarity* 24
- 2.3 *Related Work* 25
- 2.4 *Transportation Distance-Based Methods* 31
- 2.5 *Density Overlap Kernel* 45
- 2.6 *Lift Kernel* 53
- 2.7 *Examples and Comparisons* 56

**3 Clustering Comparison 61**

- 3.1 *Spatial Information in Clustering* 64
- 3.2 *Related Work* 66
- 3.3 *CDISTANCE : A Spatially Aware Distance Between Clusterings* 75
- 3.4 *Stability* 83
- 3.5 *Conclusion* 85

**4 Ensemble Clustering 87**

- 4.1 *Ensemble Clustering Overview* 89
- 4.2 *Related Work* 90
- 4.3 *Algorithm* 93
- 4.4 *Experiments* 101
- 4.5 *Conclusion* 105

- 5 Neuroimaging 106
  - 5.1 *Framework* 106
  - 5.2 *Alzheimer's Disease* 111
  - 5.3 *White Matter* 114
  - 5.4 *Literature Review* 121
  - 5.5 *Study Data* 124
  - 5.6 *Before vs. After* 127
  - 5.7 *APOE Status Classification* 136
  - 5.8 *Predicting Cognitive Changes* 138
  - 5.9 *Discussion* 143
  
- 6 Characteristic Numbers: Revealing New Properties of Classical Probability Distributions 146
  - 6.1 *Characteristic Numbers for Distributions* 147
  - 6.2 *Computing  $d_{AT}$*  151
  - 6.3 *Analytic Results for Self- $d_{AS}$*  156
  - 6.4 *Empirical Results* 165
  - 6.5 *Conclusion* 167
  
- 7 Goodness-of-Fit Testing 168
  - 7.1 *Background* 168
  - 7.2 *Viewing as Hypothesis Tests* 170
  - 7.3 *Related Work* 172
  - 7.4 *Building a Hypothesis Test for Normality* 176
  - 7.5 *Conclusion* 179
  
- 8 Evaluations 180
  - 8.1 *Classification of Samples from Probability Distributions* 180
  - 8.2 *Ensemble Clustering* 184
  - 8.3 *Neuroimaging* 187
  - 8.4 *Goodness-of-Fit Tests* 193
  - 8.5 *Document Classification* 196
  - 8.6 *Object Classification in Images* 199
  - 8.7 *Protein Structure Similarity* 200
  
- 9 Conclusion 205
  - 9.1 *Summary* 205
  - 9.2 *Contributions* 208



9.3 *Closing Remarks* 210

References 211

**A** Characteristic numbers using  $\ell_2^2$  231

A.1 *Uniform Distributions* 231

A.2 *Normal Distributions* 232

A.3 *Exponential Distributions* 234

## LIST OF TABLES

---

2.1	Accuracy results for synthetic 1-D point set classification . . . . .	58
2.2	Accuracy results for synthetic 2-D point set classification . . . . .	58
3.1	Features of various clustering comparison measures . . . . .	72
3.2	Distances to modified clusterings according to an assortment of measures (1) . . . . .	73
3.3	Distances to modified clusterings according to an assortment of measures (2) . . . . .	74
4.1	Characteristics of data sets used in evaluation of ensemble clustering algorithm SEC . . . . .	101
4.2	Results of ensemble clustering algorithms . . . . .	102
5.1	Labels corresponding to regions of interest . . . . .	133
5.2	Classification results in the before-after experiment . . . . .	133
5.3	Classification results for predicting <i>Speed and Flexibility</i> from voxels	141
5.4	Classification results for predicting <i>Speed and Flexibility</i> from 30 clus- ters of voxels . . . . .	142
6.1	Characteristic numbers for three distributions . . . . .	146
6.2	Characteristic numbers for three distributions . . . . .	165
7.1	Upper bounds for the <i>AS</i> test statistic . . . . .	177
7.2	Comparison of <i>AS</i> with other normality tests . . . . .	178
8.1	Accuracy results for synthetic 1-D point set classification . . . . .	182
8.2	Accuracy results for synthetic 2-D point set classification . . . . .	183
8.3	Characteristics of data sets used in evaluation of ensemble clustering algorithm SEC . . . . .	184
8.4	Results of ensemble clustering algorithms . . . . .	185
8.5	Labels corresponding to regions of interest . . . . .	190
8.6	Classification results in the before-after experiment . . . . .	190
8.7	Classification results for predicting <i>Speed and Flexibility</i> from voxels	192
8.8	Classification results for predicting <i>Speed and Flexibility</i> from 30 clus- ters of voxels . . . . .	193
8.9	Upper bounds for the <i>AS</i> test statistic . . . . .	194

8.10 Comparison of <i>AS</i> with other normality tests . . . . .	195
8.11 Results of document classification experiment . . . . .	198
8.12 Results on ETH-80 data set using all 128 features . . . . .	201
8.13 Results on ETH-80 data set using 10 features derived via principal components analysis . . . . .	201

## LIST OF FIGURES

---

1.1	Classifying Point Sets . . . . .	2
1.2	Regions of the splenium of the corpus callosum showing a consistent decrease or increase in fractional anisotropy . . . . .	4
1.3	Pairs of different clusterings . . . . .	6
1.4	Characteristic numbers for three distributions . . . . .	7
1.5	Point set representations of two documents in a semantic space . . . . .	9
1.6	A didactic example showing the value of spatial information . . . . .	12
2.1	An example showing the unsuitability of using Euclidean distance between means to measure distances between point sets . . . . .	19
2.2	An example of a multiple instance learning setup . . . . .	22
2.3	Examining examples of overlap in solid shapes and point sets. . . . .	23
2.4	Point set examples where reductive methods fail to detect shape information . . . . .	26
2.5	Point set examples where probability divergence measures fail to detect shape information . . . . .	28
2.6	Showing the behavior of $S_{IM}$ with respect to scale and overlap . . . . .	31
2.7	Illustration of the transportation problem on a sphere . . . . .	33
2.8	A view of the Transportation Problem (1) . . . . .	34
2.9	Comparing $S_{IM}$ with normalizations of $d_{KW}$ . . . . .	40
2.10	Plot of $S_{IM}$ between two point sets as a function of separation distance and rotation angle . . . . .	41
2.11	Error in $S_{IM}$ when approximated by hyperclustering . . . . .	44
2.12	Visualization of the density overlap kernel . . . . .	46
2.13	Plot of the density overlap kernel between two point sets as a function of separation distance . . . . .	49
2.14	An illustration of the lift kernel for three point sets . . . . .	54
2.15	Plot of the lift kernel and $S_{IM}$ between two point sets as a function of separation distance . . . . .	55
2.16	Probability density function plots for distributions in point set classification experiment 1 . . . . .	57
2.17	Probability density function plots for distributions in point set classification experiment 2 . . . . .	59
3.1	What is a clustering . . . . .	62

3.2	Pairs of different clusterings that are indistinguishable by non-spatially aware comparison measures . . . . .	63
3.3	Demonstration of steps involved in computing CDISTANCE . . . . .	77
3.4	Values of CDISTANCE between example data sets and clusterings . . . . .	81
3.5	Examining stability of CDISTANCE with respect to aggregations of small changes in partitioning . . . . .	82
3.6	Examining the variation of ADCO and CDISTANCE with respect to small movements of points in the data set . . . . .	83
3.7	CDISTANCE values for partitions generated from subsets of a data set . . . . .	85
4.1	Steps in an Ensemble Framework . . . . .	88
4.2	Contrast between output of spatially aware and unaware ensemble clustering algorithms on a sample data set . . . . .	92
4.3	Behavior of un-normalized version of $d_{KW}$ . . . . .	94
4.4	Demonstration of a cut on a cluster graph generated from an ensemble of clusterings . . . . .	98
5.1	Depiction of a healthy brain and one with advanced Alzheimer's disease . . . . .	107
5.2	Modern understanding of the stages of Alzheimer's disease . . . . .	112
5.3	Axial cross-sectional view of gray and white matter regions in the brain . . . . .	115
5.4	Isotropic and anisotropic diffusion of water . . . . .	116
5.5	FA values for an axial cross-section of a brain . . . . .	119
5.6	Histogram of ages of study participants . . . . .	125
5.7	3-D views of the splenium of the corpus callosum . . . . .	128
5.8	Two axial slices from DTI scans of the same participant . . . . .	129
5.9	Gram matrices represented as images for two kernels . . . . .	134
5.10	Variation of classification accuracy with CONS . . . . .	135
5.11	Portions of the splenium of the corpus callosum with high CONS value . . . . .	135
5.12	Views of 6 white matter regions and voxels exhibiting significant and consistent changes . . . . .	137
5.13	Regions in the corpus callosum most predictive of APOE status . . . . .	138
5.14	Plot of <i>Speed and Flexibility</i> scores for subjects . . . . .	140
5.15	Corpus Callosum voxels clustered by $Q$ value. . . . .	141
6.1	A view of the Transportation Problem (2) . . . . .	149
6.2	Empirically calculated $d_{AS}$ values for five distributions . . . . .	166

7.1	Plotting the power of a normality test against dimensionality and sample size . . . . .	179
8.1	Probability density function plots for distributions in point set classification experiment 1 . . . . .	181
8.2	Probability density function plots for distributions in point set classification experiment 2 . . . . .	183
8.3	Gram matrices represented as images for two kernels . . . . .	189
8.4	Point set representations of two documents in a semantic space . . . . .	199
8.5	Images from ETH-80 data set . . . . .	202
8.6	Alignment of proteins 1BCT and 2IFO . . . . .	202
8.7	Alignment of proteins 1ABA and 1GRX . . . . .	203

ABSTRACT

---

This thesis examines a class of problems in which the *spatial layout* (shape) of data points enables inductive inference. We (1) introduce novel mathematical and computational tools that are inherently sensitive to shape and (2) formulate spatially sensitive transformations that simplify application of pre-existing methodologies, such as support vector machines. Our choice of representation, point sets, enable fuller yet lower-dimensional descriptions of data. This representation closely models many real-world knowledge representation needs that benefit from its flexibility. We solve problems in classification, clustering, and regression for many of which spatial knowledge is crucial for obtaining a solution. Furthermore, we demonstrate that previous approaches sometimes ignore the basic most informative aspects of data and in retrospect provide counter-intuitive solutions.

We explore novel and existing measures of similarity between point sets based on exploiting the geometric **spatial relationships** in the underlying domain between data points. Many of these techniques are built upon innovative ways of extending an intuitive notion of “spatial overlap” between solids to rigorous definitions for *sets of points* that by definition are zero-dimensional and thus have no overlap. In addition to a study of theoretical aspects of the point set representation we also show extensive demonstrations of its diverse applicability.

In the neuroscience domain we introduce a new framework using these techniques that allows us to reason about individuals, as opposed to populations. We study the problem of detecting minute, short-term changes in white matter structure in the brain and relating them to changes in cognitive test scores and genetic biomarkers. Our results present the first evidence demonstrating that very small changes in white matter structure over a two year period can predict change in cognitive function in healthy adults.

In other domains we present new results and techniques in clustering comparison, natural language processing, object recognition in images, goodness-of-fit testing, and multivariate point set classification.

## 1 INTRODUCTION

---

In this dissertation we address machine learning over *thoroughly unstructured* data, naturally leading one to ask exactly what "thoroughly unstructured" means. It is perhaps easier to say what it isn't. In machine learning, problems are often posed via input-output pairs  $(x, y)$  to enable *training*, where the goal is to learn a function that properly assigns an unencountered input  $x$  to an appropriate value of  $y$  according to some inductive bias. This is, of course, an entirely reasonable and ubiquitous paradigm.

However, what exactly do we assume about these  $x$ 's? It is extraordinarily common to use *fixed-length, ordered vectors* to represent data. For example, we might make weather predictions based on a set of predetermined and serialized features describing climatic conditions. Here, there is an explicit assumption that each individual  $x$  is internally consistent, meaning the features are not independent of one another. Surely, we'd be surprised to find it was simultaneously 95°F and snowing.

Alternatively, the  $x$  inputs may be more complex. For example, they could be drawn from structured (database) records, where each  $x$  contains labels and associated values. It needn't be the case that every record has an identical set of labels. If each  $x$  corresponds to climate, the *humidity* information (label) might be missing for particular days. Nonetheless, we assume a fair degree of label overlap, enabling inferences to be made.

It may also be the case that each  $x$  is composed of largely independent components. This is common when each  $x$  itself is a set of inputs, of which only a few are thought to be significant to the outcome  $y$ . For example, the  $x$  might be individual stock prices for the goal of predicting the Dow Jones Industrial Index or groups of medications given to an individual patient, where  $y$  represents



that patient’s clinical outcome. In this case, we make no commitment regarding the independence among the components of each input data  $x$ .

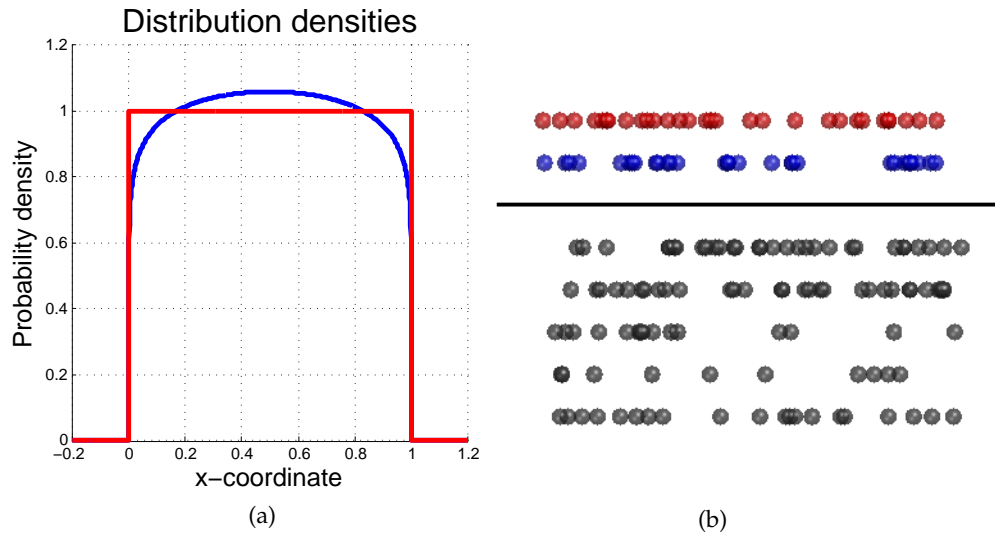


Figure 1.1: Panel (a) shows plots of two probability distributions, one in red and the other in blue. The distributions have identical means and almost identical variance. Panel (b) shows finite point samples — of different sizes — drawn from each distribution. The top two samples are colored according to their originating distribution. Our goal is to classify the grey *samples* by determining which source distribution they are most likely drawn from. This is very challenging to do for distributions that are very similar to each other and differ in only minute and subtle ways, such as those shown in (a).

Now, let us consider an alternative type of input  $x$  not easily covered by the above cases and which poses specific challenges for many approaches in the machine learning toolbox. Consider a scenario where data consist of bags of varying numbers of points — i.e. *point sets* — with all the points in each bag being drawn from one of two distinct distributions. An example is shown in Figure 1.1. The task is to learn a classifier for these two kinds of bags. This can be a challenging task for many real world domains; consider a domain in this thesis — neuroscience — where a data set may consist of selected voxels from brain scans. The voxels may be selected by setting a threshold on their intensities, such

that the scans for older people in the study consist of fewer voxels than those of younger subjects. The differences in the underlying distributions of voxels for healthy and pathological subjects may be extremely subtle and difficult to detect. What kind of representation and learning algorithm may be used in this case to build a classifier? This type of question has received sparing attention in the literature. As we will show in the remainder of this thesis, conventional ML methods in many cases are unable to satisfactorily tackle this type of problem.

Similar scenarios can arise in the cases of webpages, proteins, photos, samples from probability distributions, and data clusters. Data in these domains may be naturally seen as being composed of sets of elements; in the examples above, these elements are words, atoms, edges, and points. In all these cases, classification tasks at their core involve differentiating between sets of points originating from different distributions.

In this dissertation I propose new techniques and discuss existing ones for solving these types of tasks based on exploiting the geometric **spatial relationships** in the underlying domain between data points. Many of these techniques are built upon innovative ways of extending an intuitive notion of “spatial overlap” between solids to rigorous definitions for *sets of points* that by definition are zero-dimensional and thus have no overlap. The word “spatial” refers to the fact that these techniques make direct use of the locations of the points themselves and the geometric relationships between them (as opposed to discrete binning or statistical reductions such as mean, variance, kurtosis, moments, and skewness). Moreover, these techniques make no additional assumptions on the form of the probability distributions the point sets are drawn from. I then demonstrate the value of this type of spatial analysis in formulating and solving new problems as well as increasing prediction accuracies on tasks in neuroimaging and other diverse scientific domains.

## 1.1 Applications and Results

Chapters 3, 4, 5, 7, and 8 elaborate on how ideas from point set representations can be adapted to a number of machine learning applications, both new and existing. For existing applications, we were able to obtain higher accuracies and for new applications these methods paved the way for new tasks to be formulated and solved using spatial analysis. A brief summary of each follows, with more details in following chapters.

**Neuroimaging:** Our goal with this application was to use spatially sensitive analysis techniques to understand the different pathways of neural evolution inside the brain corresponding to natural aging as well as the development of pathological conditions such as Alzheimer’s disease. The experiments were therefore framed with the intent of producing interpretable results: (1) Prediction of the direction of change in two scans from the same subject taken approximately 2 years apart. This experiment was designed to detect patterns of subtle changes in the brains of the subject population (shown in Figure 1.2) that enabled the following two experiments. (2) Prediction of the presence

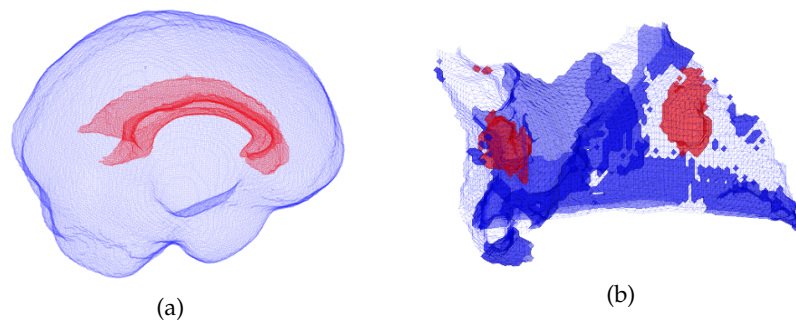


Figure 1.2: (a) An image indicating the position of the largest white matter body in the brain — the corpus callosum. (b) Regions of the splenium (rear) of the corpus callosum that showed an overall consistent decrease or increase in fractional anisotropy (white matter integrity) over the whole population. Red regions indicate a consistent increase and blue regions a decrease. Identification of these regions enabled prediction of cognitive scores.

of a gene form solely based on brain scans. We were able to identify regions that change differently based on the presence or absence of the  $\epsilon 4$  allele of the APOE gene. (3) Prediction of the direction of change in performance on neuropsychological cognitive tests based on scans. We were able to predict an increase or decrease in performance on certain neuropsychological tests with 75% accuracy solely based on the change in a subject's scans over a two year period. The goal of these experiments is to advance current understanding of the progress of neural decay and regeneration in the brain and to find imaging markers for early diagnosis of Alzheimer's disease (Ansari et al., 2014; Coen et al., 2013).

**Clustering Comparison and Ensemble Clustering:** We introduced spatial awareness into the process of comparing clusterings and constructing a consensus clustering from an ensemble. The measure we formulated is able to detect differences in clusterings that extant methods cannot. It combines both partitional and geometric information contained in the clusterings and is unique in enabling comparisons between clusterings that differ in their underlying data sets, number of points, and number of clusters. An example is shown in Figure 1.3.

This idea was then extended to formulate an end-to-end ensemble clustering algorithm that was evaluated on six different data sets (with varying numbers of features and instances) with ground truth. Our algorithm arrived at a consensus clustering with the least error (as measured against given labels) on all data sets when compared with competing algorithms. In the case of one large data set, our consensus method was able to achieve an error not only significantly lower than the most accurate clustering in the ensemble, but also significantly lower than two state-of-the-art ensemble clustering algorithms (Coen et al., 2010; Ansari et al., 2010).

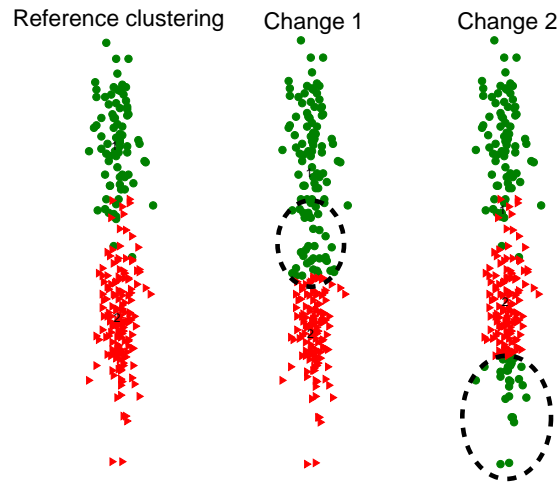


Figure 1.3: **Comparing Clusterings.** This figure displays three clusterings. Each one contains two clusters, whose members are indicated by green circles and red triangles. In the two changed clusterings, the circled points have been reassigned from the red to the green cluster. We might expect that the Reference Clustering is more similar to Change 1 than Change 2 because the modified points are *closer* to it. The vast majority of extant techniques are incapable of distinguishing between the two changes. This figure is taken from Coen, Ansari, and Fillmore (2010).

**Characteristic Numbers:** Drawing upon ideas from point set comparison we landed upon theoretically interesting and surprising results related to classical probability distributions. We defined a new non-trivial quantity called a “characteristic number” over probability distributions that is constant for certain families of distributions irrespective of their parameters. This leads to being able to characterize and differentiate between families of distributions without knowing them a priori. Characteristic numbers for normal, uniform, and exponential distributions are shown in Figure 1.4.

**Goodness-of-fit testing:** The quantity described above can be adapted to discrete point sets and used in powerful new goodness-of-fit tests, i.e., tests that determine how likely it is that a sample originated from a particular probability distribution. Our test is unique in that it uses the coordinates of each point

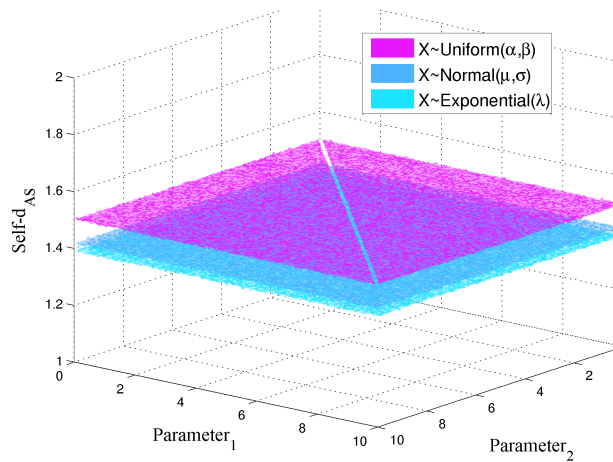


Figure 1.4: This plot shows empirically computed (discrete equivalents of) characteristic numbers for samples from three families of distributions: normal, uniform, and exponential. For each parameter setting, the average value of this quantity over 500 samples is shown. As the planes show, characteristic numbers are constant for distributions from a particular family regardless of their parameters.

directly, rather than relying on summary statistics such as mean, variance, kurtosis, moments, or skewness. Our method performs exceptionally well in the comparison of its statistical power (being able to differentiate between samples from distinct probability distributions) with other state-of-the-art techniques (Coen et al., 2012).

**Text classification:** We constructed point sets for documents by treating each word as a member of a semantic space we defined and conducted classification experiments with this representation. For a 2-class binary classification task on the popular 20 Newsgroups data set (Lang, 1995), our method was able to achieve a higher accuracy (92.75%) than standard bag-of-word techniques (83.33%) (Coen et al., 2011).

**Image classification:** We applied a spatially sensitive similarity kernel to an image classification task on a subset of the publicly available ETH-80 dataset (Leibe

and Schiele, 2003), using the data and experimental setup of Grauman and Darrell (2007). We trained a classifier using a variety of kernels on the problem of predicting the category of an object in an image, and were able to achieve a higher accuracy (94%) than a state of the art algorithm (90%) (Coen et al., 2011).

## 1.2 The Point Set Representation

The typical representation of data in machine learning algorithms is a vector of features that each describe an aspect of an instance (Liu and Motoda, 1998). To borrow an example from Mitchell (1997), features that describe the climatic attributes of a day may include the condition of the skies, air temperature, humidity level, and wind. Measurements of these attributes in a vector constitute one instance of a day, which may then be input to a learning algorithm. In this scheme, every data point (consisting of weather information for one day) is reduced to a single fixed-length feature vector without regard to the complexity of climatic variation during the day. This method of representing data has been wildly successful in a wide variety of domains from text classification to image retrieval to protein classification (Joachims, 1998; Liu et al., 2003; Ambroise and McLachlan, 2002; Camastra and Verri, 2005; Cai et al., 2003).

To take another example, let us consider some different ways in which image data may be represented. A common, if primitive, method in computer vision is to construct feature vectors for images from color values of pixels (Roobaert and Van Hulle, 1999; Pontil and Verri, 1998; Ishii et al., 2005). This method treats each pixel as an independent dimension (in fact, it treats each descriptor at every pixel as an independent dimension). This of course leads to a very high-dimensional representation even for modest-sized images. When two images in this representation are compared, the pixel-to-pixel correspondences remain but any information about spatial locality of neighboring pixels and





Consider a third representation consisting of *unordered* and *variable-sized* feature sets: an image as a *set* of low-dimensional local descriptors corresponding to image elements (whether pixels or regions of interest). In this case, instead of one very long vector corresponding to each image, it now corresponds to a *set* of short vectors <sup>1</sup> (see Figure 1.5 for a visual demonstration of two documents represented as point sets instead of vectors). Instead of guessing which global aspects of an instance best describe it we let the elements within an instance each have an independent contribution in the final representation.

Notice that this makes available an additional design choice: the ability to represent individual pixels or regions in a space of their own via features that capture meaningful aspects of each one, such as location and color profile. For a text document represented as a set of words, this space may be constructed of features that measure the syntactic and semantic content of each word. We will call this a “point set” or a “spatial” representation — alluding to the fact that data elements are kept in their own space (as opposed to a constructed space consisting of more global summary features) during the machine learning process.

It follows directly that another benefit of this representation is that it is “lossless”: all components forming a data instance can be represented in full, no matter how many there are, or even if their number differs from instance to instance. Yet another benefit is that we are able to perform inference in lower-dimensional spaces, avoiding the problems that come with sparse high-dimensional representations.

This dissertation is devoted to examining the benefits and costs of this representation at a theoretical level and a demonstrating its application to diverse scientific domains. Chapter 2 consists of a discussion of point sets, methods of

---

<sup>1</sup>For a discussion of how this relates to multiple instance learning the reader is referred to Section 2.1.4

defining similarity between them, and how to compute them. The focus will be on methods that incorporate learning biases — through the similarity function — that take into account the spatial relationships between constituent elements of data instances. In doing so we wish to make the most use of the inherent structure already in the data. The methods discussed draw on ideas from optimization theory, nonparametric density estimation, and random Fourier features.

### 1.3 The Value of Spatial Information

Many machine learning algorithms such as nearest neighbors (Cover and Hart, 1967), support vector machines (Shawe-Taylor and Cristianini, 2000), and k-means clustering (MacQueen et al., 1967) require the computation of a similarity (or distance <sup>2</sup>) between data instances. With the fixed-length vector representation this similarity is typically computed as a function of the differences in the values of the features (descriptors) of each instance. When calculating similarity between two instances, it is often useful to consider not only the changes in the values of each feature, but also the relative *locations* of those features, wherever those features are embeddable in a metric space. The choice of representation plays a big role in enabling this, as does the choice of the comparison function. The following example brings out the value of encoding and using spatial information in the distance function, whether one uses the fixed-length vector representation or the point set representation or another representation altogether.

Figure 1.6 shows a simple example where a naïve choice of representation and comparison function can omit potentially valuable information. A fixed-length 9-tuple representation is chosen to represent the configuration of a  $3 \times 3$

---

<sup>2</sup>See Section 2.1.3 for a discussion on the duality between similarity and distance functions

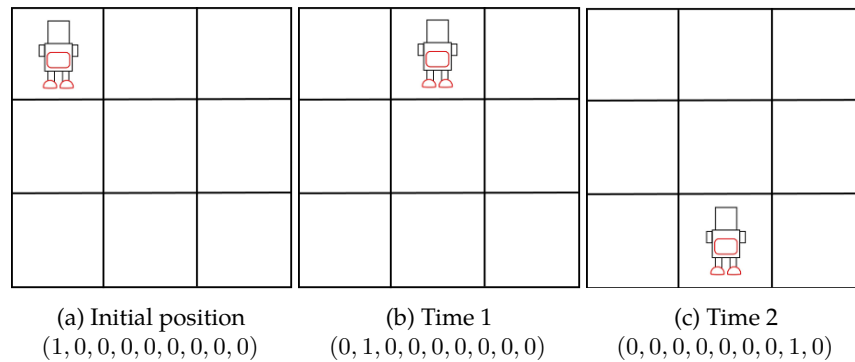


Figure 1.6: Consider the setup shown in the panels above: a robot occupies one square on a  $3 \times 3$  grid. We want to compute how different two positions are. Each possible position is one configuration; three such configurations are shown in the panels above. One possible representation for configurations is shown below each panel, wherein the squares of the grid are serialized as numbers in a feature vector. The presence of the robot in a square is encoded as 1 and its absence as 0. A typical way of computing distance between feature vectors is using Euclidean distance. However, this combination of representation and distance function omits all spatial information. If the configuration in panel (a) is considered to be the initial position, and panels (b) and (c) the positions at two subsequent times, the “distance” between panels (a) and (b) is the same as the distance between (a) and (c) when it is clear that the robot is further away in (c) from the origin than in (b). This representation and distance function combination is thus not useful for an application that is sensitive to the actual distance the robot has moved. It is straightforward to modify either the representation or distance function to accommodate this requirement for this example but may not be as simple in other cases.

board on which one square is occupied by a robot capable of moving between squares. As the robot moves, the distance between configurations is computed using Euclidean distance between the 9-tuples. The end effect of this distance is simply to measure whether two board configurations are identical or different; it does not convey *how* different they are, or *how far* the robot is in them.

In this particular case, either the representation or comparison function or both can be easily modified so that the distance between two boards more accurately reflects the distance moved by the robot. However this is not as simple or straightforward to do in many other situations such as in a medical imaging domain where continuous quantities are associated with locations and

we are interested in not just the differences between two images but also *where* those differences are. This is because organs like brains and lungs come in different shapes and sizes and the process of aligning them to a single template can be imperfect. We may wish to have similar imaging markers in nearby regions of two brain scans correspond to higher similarity between them. The choice of representation and distance function should be able to encode the relevant spatial information and deliver such a result.

## 1.4 Thesis Roadmap

In the rest of this dissertation I will demonstrate that it is often beneficial to keep data in their original space rather than abstracting it away. In other words, by using information pertaining to geometric relations between the data that is faithful to their native topology we can improve machine learning outcomes. The following chapters elaborate on concepts and results mentioned above. All work was done in close collaboration with my advisor Prof. Michael Coen. Chapter 2 consists of a discussion of point sets and methods of defining similarity between them. These methods will be used throughout the rest of the thesis. Parts of this chapter build upon tools and ideas developed in Prof. Coen's PhD thesis (Coen, 2006). This chapter also includes joint work with Nathanael Fillmore. Chapters 3 and 4 detail a new clustering comparison measure and a novel spatially aware ensemble clustering algorithm respectively. Chapter 5 presents an adaptation and application of our framework to the problem of identifying subtle changes in the brain using longitudinal neuroimaging data and was done with extensive guidance from Prof. Barbara Bendlin. Chapters 6 and 7 present a new result on the characterization of classical probability distributions based on measures from optimization theory and an application to goodness-of-fit testing. This work also was done in collaboration with Marissa

Phillips. Chapter 8 contains details on the applications of point set representation to the domains of text classification, image classification, and protein structure similarity, and is the result of joint work with Nathanael Fillmore and Layla Oesper. Finally, I conclude in Chapter 9.

## 2 POINT SETS

---

The point set representation — as introduced in the previous chapter — is a lossless way of representing variable-sized, unstructured data. Each data instance is represented as a *set* of points corresponding to its components, rather than a single point. In this chapter we will formally define this representation, discuss its costs and benefits, present new and existing ways of defining a similarity function between point sets, and discuss their strengths and shortcomings.

The term “spatially sensitive” as used in this chapter and the rest of this dissertation will refer to the ability of an algorithm to work directly with the coordinates of *all* points in a point set in order to leverage information about the (pairwise) spatial relationships between those points. We elaborate on this idea in Section 2.2 and discuss three such algorithms in detail:

- **Similarity (SIM)** (Section 2.4) uses optimization theory to define a measure of spatial overlap between point sets.
- **Density Overlap Kernel** (Section 2.5) is the dot product of two functions constructed from density profiles corresponding to each point set.
- **Lift Kernel** (Section 2.6) uses random Fourier features (Rahimi and Recht, 2007) to construct a high-dimensional vector representation of a point set that encapsulates pairwise spatial relationships between its points.

At the end of this chapter we show results from classification experiments using spatially sensitive and insensitive methods on synthetic samples generated from 1-dimensional and 2-dimensional distributions. The following chapters detail how these techniques are applied to machine learning tasks in various domains.

## 2.1 Representation and Definitions

Recall that in the previous chapter we identified domains where instances of data could be seen as being composed of a variable number of unordered elements of the same type. Fixed-length vector representations for each of these elements in an instance can be put together in a set; the result is called the point set representation for that instance. Because these vectors each only need to describe one element of an instance fully, they are typically lower-dimensional than a fixed-length representation describing an entire instance. Moreover, the contribution of each element of the instance is preserved in the final representation. In contrast, fixed-length representations are either unable to deal with instances consisting of varying numbers of elements or will require the computation of summary statistics through lossy aggregation over multiple elements. A *weighted point set* is defined as follows:

**Definition 2.1.** *Weighted Point Set:* A finite collection of pairs  $P = \{(p_1, \omega_1), \dots, (p_n, \omega_n)\}$  where  $n \in \mathbb{N}$  and each  $p_i \in \mathcal{X}$  for some associated metric space  $\mathcal{X}$  endowed with a distance  $d_{\mathcal{X}}$ , called the “ground” distance. Each point  $p_i$  has an associated weight  $\omega_i \in [0, 1]$ , such that  $\sum_{i=0}^n \omega_i = 1$ .

In the definition above,  $\omega$  corresponds to a discrete probability distribution over the domain  $\mathcal{X}$ , for example,  $\mathbb{R}^d$ . In the case where the  $\omega_i$ ’s are all equal (to  $\frac{1}{|P|}$ ) we will omit them altogether and consider a point set as simply being the set  $\{p_1, \dots, p_n\}$ .

### 2.1.1 Benefits

The design decision of representing data in the form of point sets rather than single fixed-length vectors has a number of immediate benefits, some of which are outlined below:

- We are able to keep vectors in low-dimensional non-sparse spaces, thus avoiding sparsity issues and the curse of dimensionality (Duda et al., 2012; Beyer et al., 1999). This neatly sidesteps the entire problem of working in extremely high dimensions and obviates the need to mitigate its effect during the learning process. For example, it is common to represent data in spaces with tens of thousands of dimensions in text and image analysis. With the point set representation, one only needs as many dimensions as necessary to describe *one element* of the entire instance.
- In many domains, due to the constraint that each instance must have the same number of features, the conventional representation is often lossy in the sense that it is constructed from the extraction of summary features over an entire instance (e.g. histograms over image pixels). Instead, the point set representation allows for a richer representation without needing to settle for summary statistics.
- Following from above, **the point set representation allows for variable cardinality sets that describe each instance**. Rather than each instance being constrained to have a fixed number of features, this representation allows for the number of vectors in its set to vary according to its specifics. For example, if every word is mapped to a vector encoding its semantic meaning, a 100-word document will be represented as a point set with 100 points, and a 1000-word document as one with 1000 points, each containing just the amount of information necessary to describe it without needing to resort to lossy representations. This has the added benefit of allowing for the flexibility of different dimensions of inputs. For example, many computer vision algorithms require all input images to have a certain fixed input size (or be preprocessed to that size). This no longer need be the case.



- This representation makes the construction of domain-specific custom kernels for machine learning easier. The scientist can focus on designing similarity measures for low-dimensional vectors that directly represent elements of an instance. In fact, if there is already a natural way of computing similarity between the elements of an instance, this can be used as is without the need for any other kernel function. This stands in contrast to either working with general nonspecific kernels like polynomial or Gaussian kernels, or designing kernels for extremely high-dimensional representations.

While there are a number of clear benefits of the point set representation, it is not a panacea for all problems in machine learning. It is suited for problems where data instances are composed of multiple independent (but possibly related) entities, e.g. words in a document or pixels/regions in an image. Sometimes the spatial angle for the representation is obvious (e.g. if the data already come from a domain where its elements are embedded in a metric space), while at other times it has to be artificially constructed (such as with words in text documents). Regardless, the benefit of remaining in a lower-dimensional space as opposed to a very high-dimensional one is applicable in both situations.

### 2.1.2 The Similarity Question

Having defined the representation, we will now discuss ways of using it in a learning setting. A core problem for any representation, if it is to be used in a comparison-based learning algorithm such as nearest-neighbors (Cover and Hart, 1967), support vector machines (Shawe-Taylor and Cristianini, 2000), or k-means clustering (MacQueen et al., 1967), is to define a meaningful similarity (or distance<sup>1</sup>) function between data instances. In the case of point sets, this

---

<sup>1</sup>See Section 2.1.3 for a discussion on the duality between similarity and distance functions.

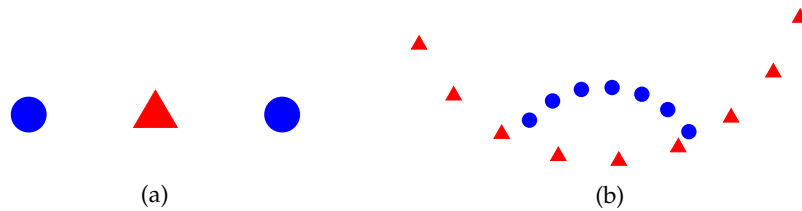


Figure 2.1: These two panels demonstrate the unsuitability of Euclidean distance as a way to measure distance between sets of points. In the panel to the left, one point set consists of the two blue circles, and another point set the single red triangle. These two point sets are distinct, and yet they would be indistinguishable if each one was reduced to its mean. Similarly in the right panel the blue point set and red point sets have different geometric shapes and have almost no overlap, and yet the Euclidean distance between their means is zero.

translates to the question of how point sets are to be compared to one another. This problem is not as straightforward as defining ways to compare two points. The following sections present a thorough investigation of this problem and approaches to solving it.

Note that a set-to-set similarity function by itself doesn't necessarily impose a restriction on how similarity is defined between single points. Put another way, set-to-set similarity functions can be parameterized by the point-to-point similarity function. The point-to-point function (also called a "ground distance" function) can be complex and domain specific, and it will fit in just as easily as Gaussian similarity into the set-to-set function. In domains that are amenable to a point set representation it is frequently the case that there are natural and simple similarity functions between *elements* of an instance. For example, given a word-to-word similarity function the methods in this chapter may be used to compute similarity between text documents.

### Euclidean Distance

A natural question at this juncture is "Why not just use Euclidean distance?" This question however is ill-posed; Euclidean distance is defined from a single

point to another point, not between unordered variable-sized point sets. A simple tweak of Euclidean distance such as reducing point sets to a summary statistic (e.g. their means) and setting the distance between point sets to the distance between these statistics does not lead to the desired result. The two panels in Figure 2.1 demonstrate the unsuitability of Euclidean distance as a way to measure distance between sets of points, if in fact one wishes such a distance to reflect information pertaining to the geometric similarity or dissimilarity of the point sets. In the panel to the left, consider one point set consisting of the two blue circles, and another point set consisting of the single red triangle. These two point sets are distinct, and yet they would be indistinguishable if we reduced each one to its mean and found the Euclidean distance between them. Similarly in the right panel the blue point set and red point sets have different geometric shapes and have almost no overlap, and yet the Euclidean distance between their means is zero. Euclidean distance between means is therefore not an appropriate distance function if we wish to measure meaningful spatial distance between point sets. Doing this in fact voids the entire point of using a richer representation that is able to encode more aspects of each instance.

### 2.1.3 Distance and Similarity

A numerical measure of comparison between two items yields a value that either describes how *similar* they are or how *different* they are. In other words, comparison measures are either similarity measures or distance measures. We note that these two are duals of each other; given a similarity measure  $s(\cdot, \cdot)$  it induces a distance  $\delta(A, B) = s(A, A) + s(B, B) - 2s(A, B)$ . In the other direction, a distance measure  $\delta(\cdot, \cdot)$  may be converted to a similarity function via any order inverting function  $f$ . Examples of such functions include:

- $s = f(\delta) = \delta_{max} - \delta$

where  $\delta_{max}$  is the maximum possible value of the distance (this transformation only applies to bounded distance measures).

- $s = f(\delta) = e^{-\gamma\delta^2}, \gamma \in \mathbb{R}^+$

- $s = f(\delta) = \frac{1}{1+\delta}$

- $s = f(\delta) = -\delta$

For a deeper discussion of substituting distance metrics in kernel functions, see Haasdonk and Bahlmann (2004). In the discussion of point set comparison frameworks below we will present each one as a similarity measure.

#### 2.1.4 Point Set Representations and Multiple Instance Learning

Multiple instance (MI) learning is a form of supervised learning where training examples are received in groups rather than individually (Dietterich et al., 1997). Target labels are only available for each group as a whole and not necessarily for each individual training example. A group is labeled positive if it contains at least one positive training example and negative otherwise. An example of a MI learning setup is shown in Figure 2.2.

There is a superficial similarity between the point set representation and the organization of input data for the multiple instance learning setup. In both cases each input to the learning algorithm consists of an unordered variable-sized set of vectors. This, however, is where the similarities end. Some major differences are as follows:

1. In multiple instance learning a “bag” consists of one or more separate instances of data; each one is a complete instance by itself and the only thing they have in common is that they are grouped together in one bag. With the point set representation of data, a point set (counterpart to a

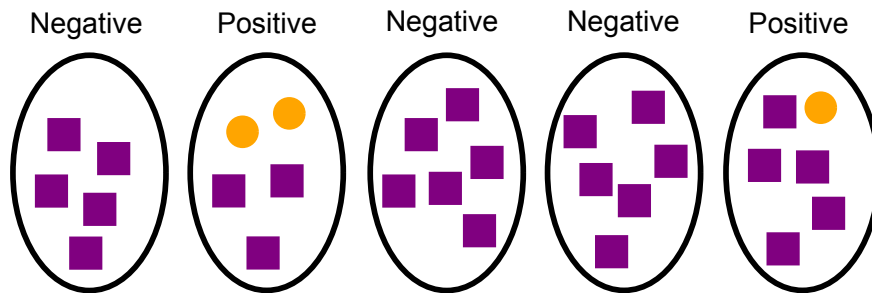


Figure 2.2: **Multiple Instance Learning.** An example of instances in a multiple instance (MI) learning setup. A training example consists of “bags” of instances. If any instance in a bag is labeled positive, the entire bag is labeled positive, otherwise it is labeled negative. The learning algorithm is only able to access labels at the level of bags, not individual instances.

bag in MI learning) consists of vectors that are different components of a *single* instance of data. In other words, all points in a set contribute to the specification of an instance.

2. Point set representations may be used for supervised or unsupervised learning. In the supervised case, the target label is a property of the instance as a whole and not of a single point within the set. With MI learning however, the label of the bag is positive if any of the (independent) instances within it are positive and negative otherwise.
3. In the case of point set representations, it is not just the presence of one vector that may be the cause of a particular label for the entire point set, but could instead be a function of a combination of constituent vectors. With MI learning on the other hand, one positively labeled example is enough to label the entire bag positive.

It is therefore a category error to treat the two as competing methods.

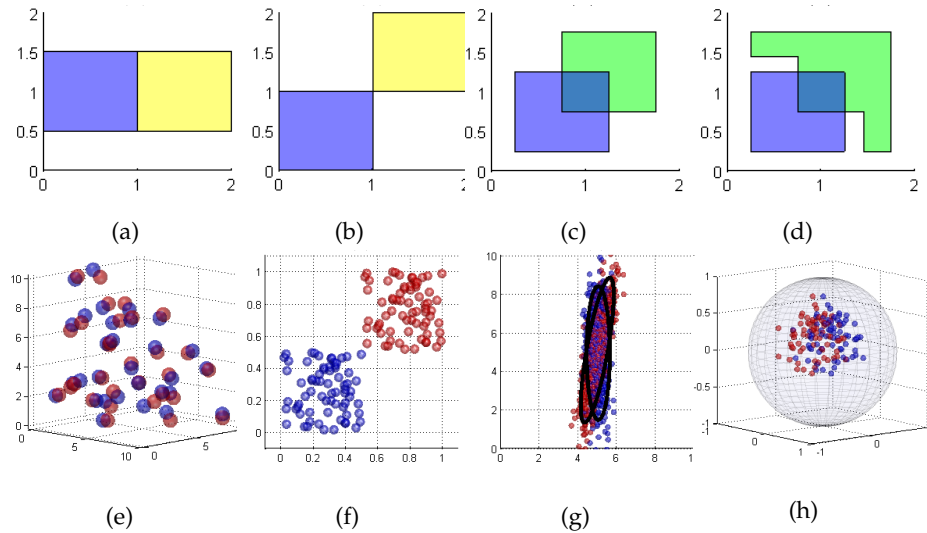


Figure 2.3: **Overlap in solids.** Consider the design decisions involved in defining similarity between solid shapes (at fixed locations) by quantifying the degree of their overlap in space. In both panels (a) and (b) the blue and yellow squares have zero overlap area, and yet in (b) the squares are further apart. One choice involved in the design of a similarity measure is whether to simply consider the overlap area alone or also include the distance between the shapes. In panel (c) the squares' overlap area is a quarter of the area of each respective square. In panel (d) however, the absolute area of overlap is the same as in panel (c) but the overlapping shapes have different areas. Another design decision is whether to simply measure the shared area or consider the shared area in relation to the areas of both shapes, and if so, how. A third design decision concerns the areas of non-overlap: should it matter *where* the area of non-overlap is in relation to the area of overlap? The similarity measures in Sections 2.3-2.6 adapt these questions to point sets and each answer them differently. **Overlap in point sets.** In the bottom row of panels, we examine examples of spatial overlap in point sets: (e) 3-dimensional “distributed” point sets that strongly overlap; (f) “Localized” non-overlapping point sets; (g) Point sets drawn from 2-dimensional Gaussian distributions that occupy a large area of space, and yet have significant overlap; (h) Moderately overlapping sets on the surface of a manifold – here, a sphere. A design decision for a point set similarity measure is how to trade off scale and amount of overlap (or whether there should be a tradeoff at all). This figure is adapted from Coen (2010).

## 2.2 Spatial Overlap as Similarity

Putting aside point sets for the moment and thinking in terms of solid shapes the abstract question of similarity between two objects becomes simpler to reason about: solid shapes have area (non-zero Lebesgue measure) and a measure of similarity for two shapes can be derived from the unambiguous amount of shared area between them. For the purposes of the following discussion we will consider shapes to have locations in space that are fixed. Consider the series of shapes in the top row of Figure 2.3 shown in various degrees of overlap. The caption contains a discussion of how similarity may be measured between different solid shapes.

If we accept that similarity is to be measured by the degree to which two shapes occupy the same region of space, and this notion then is to be extended to defining similarity between point sets, we are posing the question: “To what degree do two point sets occupy the same region in a metric space?” In beginning to formulate an answer to this question however, we confront an immediate problem: point sets have zero measure, and therefore by definition any two point sets – even if their points overlap exactly – have zero overlap if the amount of shared space is measured. Moreover, the individual points constituting each point set are unlikely to coincide with one another in real data sets. For example, if points are sampled from a continuous distribution such as the normal distribution, the probability that two points coincide is vanishingly small:

$$P(x = y | x, y \sim Normal(\mu, \sigma)) = 0$$

A strict definition of overlap will not be useful in answering this question. We therefore adopt a broader notion of spatial overlap and frame the question not in terms of the points themselves but rather in terms of the *shapes* of the

point clouds formed by each point set. In other words, one way of quantifying similarity between point sets is by developing a measure for the similarity in the shapes of the point clouds they form. The bottom row in Figure 2.3 presents examples of overlap in point sets, along with a discussion of desirable behaviors of any method of overlap quantification. We examine three different approaches for defining and measuring this notion of overlap in Sections 2.4-2.6.

## 2.3 Related Work

The earliest work in comparing point sets was in the context of comparing clusterings (the subject of Chapter 3) and relied purely on set-theoretic operations to define distance between point sets. Subsequent methods are more sophisticated, yet none of them were developed specifically with a view to quantifying spatial overlap. Most of these measures are also very sensitive to their parameters, which often requires extensive search for a given problem. This makes their use problematic in unsupervised learning problems. In the subsections below we give brief descriptions of the different families of solutions this problem has inspired in the literature.

### 2.3.1 Set Reduction Methods

The first set of approaches are inspired from point set and Hausdorff distances (Munkres, 2000) wherein a point set is reduced to a well-chosen member point. Point set distance  $d_{PS}$  is defined between a single point  $x$  and a point set  $A$  in terms of the ground distance  $d(\cdot, \cdot)$  as

$$d_{PS}(x, A) = \inf_{y \in A} d(x, y).$$

Hausdorff distance is an extension of this concept; the *directed* Hausdorff distance  $D_{Haus}(A, B)$  between two sets of points  $A$  and  $B$  is



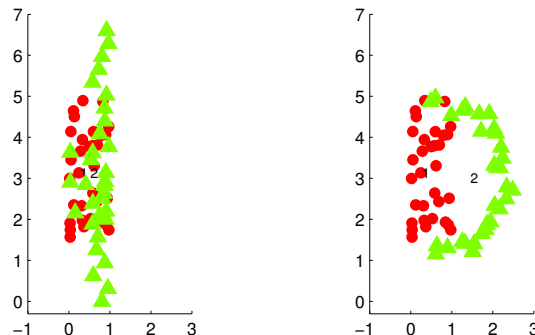


Figure 2.4: The two pairs of point sets shown here are clearly different. There is no overlap in the second case, and yet the Hausdorff distance and Procrustes distance between the two are almost identical.

$$D_{\text{Haus}}(A, B) = \sup_{x \in A} \inf_{y \in B} d(x, y).$$

The Hausdorff distance between  $A$  and  $B$  defined as the larger of  $D_{\text{Haus}}(A, B)$  and  $D_{\text{Haus}}(B, A)$ . Other metrics inspired from point-set distance are the modified Hausdorff metric and Busemann metric (Deza and Deza, 2009). We discuss this class of distances further in section 2.3.5.

In Figure 2.4 we compute Hausdorff distance between example point sets. Note that the point sets on the left overlap more with each other than the ones on the right and are more alike in their shape. However Hausdorff distance is unable to differentiate between them, reporting a distance of 1.75 in both cases.

### 2.3.2 Procrustes Distance and Variations

Another method of computing distances between point sets is to assume an order between the points in them, align them using an algorithm such as Kabsch (1976) or Procrustes (Goodall, 1991). Once an alignment is found, a distortion measure (such as least root mean square distance) can be calculated by summing up distances between corresponding pairs of points. Clearly this method can only work for point sets of the same cardinality and is susceptible

to disproportionate influence by outlying points. While modifications exist to mitigate these problems, these general methods of summing distances between pairs of points yield little information about similarity or shape congruence.

For the point sets in Figure 2.4 Procrustes returns almost equal distances of 1.87 and 1.91, unable to tell the two pairs of point sets apart.

### 2.3.3 Modified statistical distance measures

Metrics for comparing probability distributions – such as Mallows distance (Levina and Bickel, 2001) – can sometimes be modified to measure distance between point sets. Because Mallows distance computes the infimum of the expected value of functions on random variables, we can transform this into a discrete minimization problem (Levina and Bickel, 2001) suitable for discrete point sets.

Other metrics compute differences between probability mass or density functions, which have no immediate applicability in the discrete point set case without an intermediate step. It is possible to view coordinates of points as being values taken by discrete random variables but it is rarely the case that multiple points have precisely the same coordinates, making probabilities for each location degenerate into zeros or multiples of  $\frac{1}{n}$ , where  $n$  is the number of points.

For sets with large numbers of points we can bin them into regions, treat each region as having a probability value proportionate to the number of points lying within it, and apply any of a number of probability divergence measures such as Bhattacharya distance, KL-divergence, Hellinger distance or any of the family of such measures (Deza and Deza, 2009). We note that this is an approximation that degrades with point sets of low density.

The final comparisons are with probability divergence measures. Each data set is processed into a set of Voronoi regions using k-means clustering, and each region is treated as a value of a random variable, whose probability is

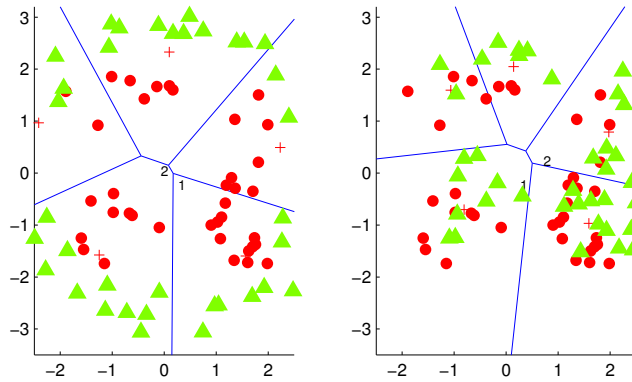


Figure 2.5: The two pairs of point sets shown here are clearly different. There is no overlap in the first case, and yet the distance according to probability divergence measures is 0.37 and 0.34 respectively.

equal to the fraction of points of that point set lying within that region. In this way we make sure to operate over the same domain, which allows the use of these measures. We chose Hellinger distance as a representative of the family of measures since it is most representative of this family of distances in terms of its form. In Figure 2.5(a) with the number of Voronoi regions  $k = 5$ , the Hellinger distance is 0.37 and in Figure 2.5(b) it is 0.34. To put these values in perspective, note that the Hellinger distances between the point sets in Figure 2.4 are 0.64 and 1.47 respectively.

A conceptual drawback of using this technique is that it is approximate – the points may lie anywhere within their Voronoi region and the divergence measure would return the same value. There is also no clear way of choosing the regions or even their number. Other divergence measures such as Bhattacharya distance, chi-squared distance, and Jeffrey divergence result in similar values and behavior. None are suited to the measure of similarity between point sets.

### 2.3.4 Kernel Methods

There are a number of methods in the literature that define kernels between point sets. A common approach is to map each point set to a single fixed-

length “supervector” and apply a standard kernel to them. This can be done by simply concatenating (also known as stacking), sampling, or averaging points together to form a new higher-dimensional point (Szummer and Picard, 1998; Campbell et al., 2006; Roobaert and Van Hulle, 1999; Kinnunen and Li, 2010). A more sophisticated approach is to construct a “vocabulary” by clustering all points in all point sets, picking a representative from each cluster, and then representing each instance as a linear combination of contributions from each cluster by computing similarities of each point in the instance’s point set to the cluster representatives. This is exemplified by quantization approaches in image retrieval tasks (Nister and Stewenius, 2006; Philbin et al., 2007). A disadvantage of these approaches is that a new mapping needs to be constructed separately for each new problem.

The Bhattacharyya point set kernel (Kondor and Jebara, 2003) is a kernel between point sets that takes into account the density of point sets in its computation. It requires a parametric model to be fit to each point set (for example a single Gaussian distribution) and defines a kernel based on the probabilistic divergence measure Bhattacharyya distance. This approach is further kernelized by mapping the elements of each point set to a new Hilbert space before fitting the parametric model. The two main issues with this approach are that it assumes a fixed distribution for the points and is quite computationally expensive due to extensive matrix multiplications, inverses and determinants.

Pyramid match (Grauman and Darrell, 2007) and other match kernels have been developed as efficient ways to determine similarity between point sets especially with vision applications in mind. Pyramid match computes histograms at different resolutions based on the input points and computes a kernel from the weighted histogram intersections. The focus in match kernels however is to find similarity *while not penalizing non-similarity*. These kernels find closest pairs among between individual points and only take into account these pairs

for the kernel computation. Thus such methods do not fully capture a notion of the “shape” of the point sets, but only their intersection, regardless of the importance of their non-overlap. This can have practical negative consequences if the “shape” or “density” of the point sets really is the important characteristic of the point sets.

### 2.3.5 Other Techniques

Many of the approaches above are lossy in the sense that they reduce two point sets to pairs of points or a single pair of points. The final distance measure thus only relies on some of the pairwise relationships between constituent points. In many domains this appears to work suitably, especially image matching. However, ignoring all points except for one pair (or a restricted set of pairs) yields no information about how similar the overall shapes of the entire point sets are. It collapses all information down to a single distance (or the sum of a few distances), stripping away all information about the internal layout and structure of each point set, as well as the relationship of points within each point set. They are also neither bounded nor scale-invariant, making absolute judgements of similarity difficult.

Finding similarity between multi-dimensional point sets is a core problem in image matching. The driving concern in that domain however is to locate objects similar to each other but transformed in some simple way, such as being rotated, reflected or translated in one of the two images (Hubo et al., 2008). The focus is on preserving distance across transformations and so the distance measures used are very primitive, e.g. minimal symmetric set difference across all translations (Cho and Mount, 2008).

A related popular approach in image retrieval and shape matching applications (Hubo et al., 2008; Osada et al., 2001) is to apply functions that sample

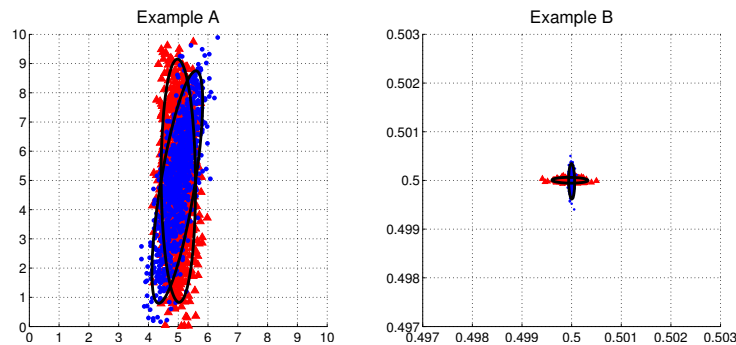


Figure 2.6: **SIM behavior.** The point set similarity measure  $S_{IM}$  considers the two point sets in Example A to be far more similar to one another than those in Example B. This is the case even though they occupy far more area in absolute terms and would be deemed less similar by many other similarity measures. This figure is taken from Coen (2006).

points and compute histograms over point sets. Probability divergence measures are applied to the binned histograms to compute final distances.

## 2.4 Transportation Distance-Based Methods

We begin discussion of spatially sensitive methods of comparing point sets in this section and continue in Sections 2.5 and 2.6. Three methods are discussed in depth, each of which takes a different approach to measuring and quantifying spatial overlap. The first method is discussed in this section and employs a distance from optimization theory to directly measure the extent of overlap of two point sets.

Originally introduced as a distance by Coen (2006), we present this method here as a similarity instead and call it  $S_{IM}$  (for Similarity).  $S_{IM}$  is designed to explicitly measure spatial overlap between point sets, without regard to their “scale.” An image is useful for illustrating this idea; consider the two pairs of point sets shown in Figure 2.6.  $S_{IM}$  is designed so that the point sets in Example A are judged to be much more similar than those in Example B, based on their

degree of spatial overlap, even though the points in Example A cover orders of magnitude more area than those in Example B.

In the following subsections we present the components that will be used to construct  $S_{IM}$ , describe how the measure works, define it formally, discuss how to compute it, and examine why this problem defies a number of standard normalization techniques.

### 2.4.1 Transportation Distances

Similarity ( $S_{IM}$ ) is derived from the Kantorovich-Wasserstein distance metric ( $d_{KW}$ ) (Kantorovich, 2006; Deza and Deza, 2009), which proposed a solution to the Transportation Problem posed by Monge (1781). This problem may be stated as follows:

*What is the optimal way to move a set of masses from suppliers to receivers, who are some distance away?*

Optimal in this definition means minimizing the amount of total work performed, where work is defined as  $mass \times distance$ . For example, we might imagine a set of factories that stock a set of warehouses, and we would like to situate them to minimize the amount of driving necessary between the two. This problem has been rediscovered in many guises, most recently in modified form as the Earth Mover's Distance (Rubner et al., 2000), which has become popular in computer vision.

We can visualize the problem solved in the computation of  $d_{KW}$  in Figure 2.7. Imagine the red squares are factories located around the world delivering identical goods to the blue triangles, which represent warehouses, also located around the world. We assume the amount of goods to be shipped is equal to the amount of goods being received, reflecting the fact that these objects represent probability distributions; they therefore have equal masses of one.  $d_{KW}$  is the

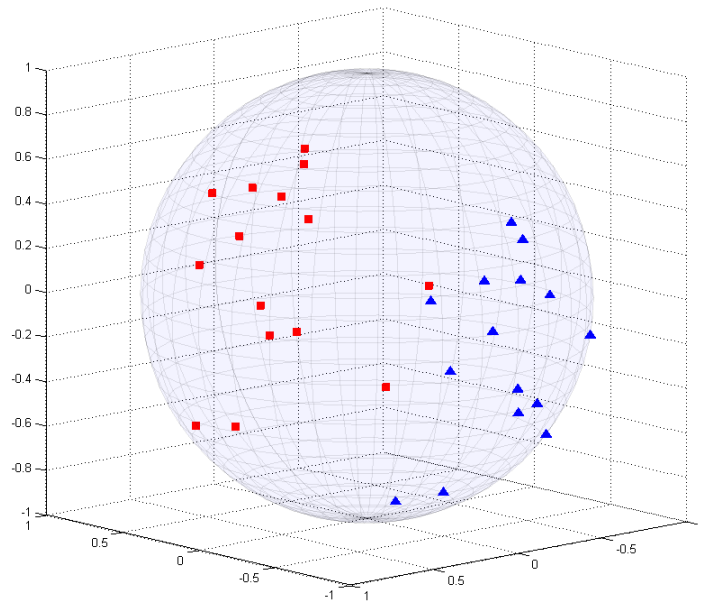


Figure 2.7: **The Transportation Problem on a sphere.** In this figure, the red squares represent factories (sources) with differing degrees of production and the blue triangles represent warehouse (sinks) with different storage capacities, on the surface of a sphere. The Kantorovich-Wasserstein distance measures the most efficient amount of work necessary to transport from the red squares to the blue triangles. We note the amount of mass being “produced” must be equivalent to the amount of mass being “consumed.” This figure is taken from Coen (2010).

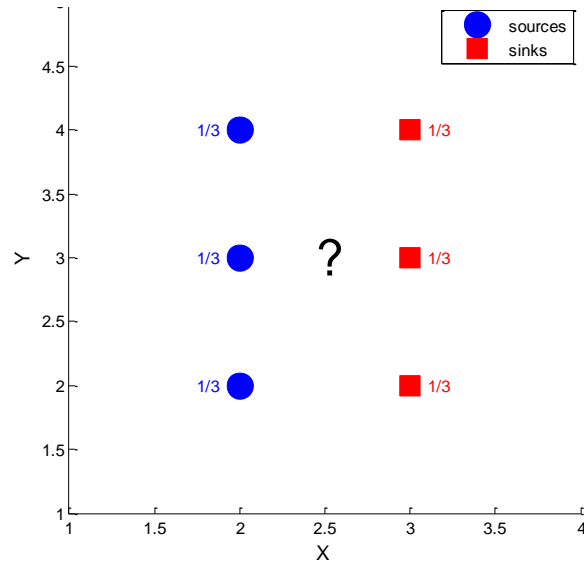
least amount of work that is required to move the masses contained in the red squares onto the blue triangles.

It is useful to view the Kantorovich-Wasserstein distance as the *maximally cooperative* way to transport masses between sources and sinks. Here, cooperative means that the sources “agree” to transport their masses with a globally minimal cost<sup>2</sup>. In other words, they communicate to determine how to minimize the amount of shipping required by exchanging delivery obligations.

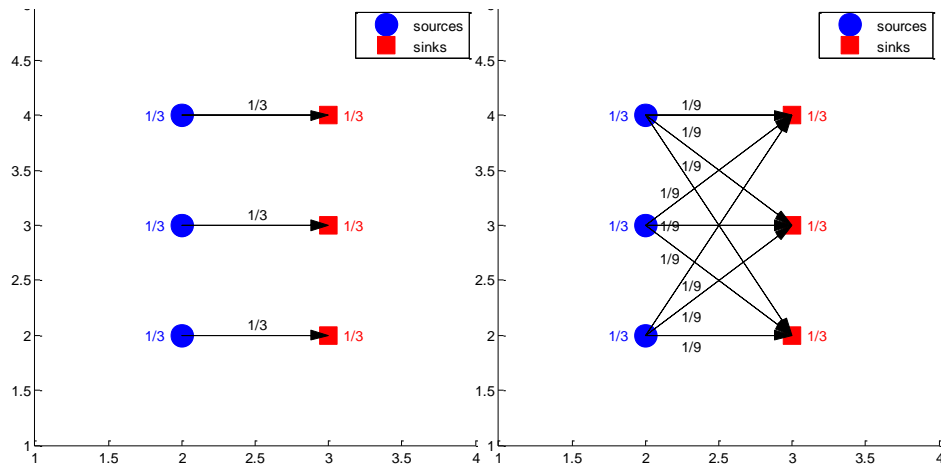
Let us contrast this optimal view with the notion that each source delivers its mass to all sinks independently of any other sources, in proportion to its

<sup>2</sup>We will use “cost” interchangeably with “work” in this context.





(a) The source (blue) and sink (red) masses



(b) The optimal solution, minimizing the amount of work.

(c) The naive solution; each source distributes its mass independently.

Figure 2.8: **A view of the Transportation Problem.** The goal is to transport the mass in the sources to the sinks, minimizing the amount of *work* performed, where work is  $\sum mass \times distance$ . This figure is taken from Coen (2010).

production. We will call this *naive transportation distance* ( $d_{NT}$ ). In other words, the sources do not communicate. Each simply makes its own deliveries to every sink proportionally. Note this is *not* the worst (i.e., most inefficient) transportation schema. It is simply what occurs if the sources are oblivious to one another — when they do not take advantage of the potential savings that could be gained by cooperation. Figure 2.8 shows the contrast between the optimal, maximally cooperative way of transferring mass and the naive, uncooperative way for a set of three sources and three sinks.

The similarity  $\text{SIM}(A, B)$  between two point sets  $A$  and  $B$  defined below uses the ratio between the two quantities  $d_{KW}(A, B)$  and  $d_{NT}(A, B)$  to quantify spatial overlap. The intuition behind this is that point sets with little overlap will have little difference between  $d_{KW}$  and  $d_{NT}$  since there is little to be gained by cooperation. On the other hand, with higher spatial overlap between  $A$  and  $B$ , the benefit of cooperation will make  $d_{KW}$  smaller than  $d_{NT}$ . The ratio therefore measures the benefit gained through cooperation, and by proxy, their spatial overlap.

## 2.4.2 Formal Definitions

### Kantorovich-Wasserstein Distance

The Kantorovich-Wasserstein distance ( $d_{KW}$ ) is a solution to the transportation problem defined above. Although Kantorovich was the first to solve this problem, the solution has since been rediscovered numerous times, most recently in Rubner et al. (2000). It is commonly presented in a form due to Vasershtein (Wasserstein, 1969) and in this instantiation known as the Wasserstein distance<sup>3</sup>. Since  $d_{KW}$  computes the solution to the transportation problem with minimum distance, we will also call it the “optimal transportation distance.”  $d_{KW}$  is a

---

<sup>3</sup>The change in name spelling is due to historical reasons.

metric over expected distance between two probability distributions. We follow the presentation in Gibbs and Su (2002). Let  $\mu$  and  $\nu$  be two probability distributions on a metric space  $\Omega$  with associated distance metric  $d_\Omega$ . Define  $d_{\text{KW}}$ :

$$d_{\text{KW}}(\mu, \nu; d_\Omega) = \inf_J \{E(d_\Omega(x, y)) : \mathcal{L}(x) = \mu, \mathcal{L}(y) = \nu\} \quad (2.1)$$

where the marginals  $\mathcal{L}$  are  $\mu$  and  $\nu$  respectively, and the infimum is taken over all joint distributions  $J$  on  $\mu$  and  $\nu$  (which are in  $\Omega \times \Omega$ ). Here,  $d_\Omega$  is the distance metric for  $\Omega$  and  $d_\Omega(x, y)$  represents work required to move a unit amount of mass from  $x$  to  $y$ . By taking the infimum,  $d_{\text{KW}}$  seeks to minimize the expected amount of work in transferring mass from one distribution to the other. This corresponds to Monge's Transportation Problem, where a mass of one is being moved between two probability distributions, one corresponding to the suppliers and the other corresponding to the receivers.

Let us now consider two discrete distributions  $A$  and  $B$ , described by weighted point sets as follows:

$$\begin{aligned} A &= \{(a_1, p_1), \dots, (a_m, p_m)\} \\ B &= \{(b_1, q_1), \dots, (b_n, q_n)\}. \end{aligned}$$

The discrete formulation of  $d_{\text{KW}}$  is obtained by reduction from Equation 2.1 (Levina and Bickel, 2001), and transforms into a minimization problem as follows. Treating  $A$  and  $B$  as random variables taking values  $\{a_i\}$  and  $\{b_j\}$  with probabilities  $\{p_i\}$  and  $\{q_j\}$  respectively,  $d_{\text{KW}}$  is obtained by minimizing the expected distance between  $A$  and  $B$  over all joint distributions  $F = (f_{ij})$  of  $A$  and  $B$ :

$$E_F \|A - B\| = \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_\Omega(a_i, b_j) \quad (2.2)$$

where  $F$  is subject to:

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n \quad (2.3)$$

$$\sum_{j=1}^n f_{ij} = p_i, 1 \leq i \leq m \quad (2.4)$$

$$\sum_{i=1}^m f_{ij} = q_j, 1 \leq j \leq n \quad (2.5)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \sum_{i=1}^m p_i = \sum_{j=1}^n q_j = 1 \quad (2.6)$$

Once so formulated this optimization problem may be solved using the transportation simplex algorithm. Although this algorithm is known to have exponential worst case runtime (Klee and Minty, 1972), it is remarkably efficient on most inputs and therefore widely used. We discuss runtime complexity and an approximation technique for enormous point sets in sections 2.4.4 and 2.4.5.

### Naive Transportation Distance

We now define a “naive” solution to the transportation problem. Here, each “supply” point is responsible for delivering its mass proportionally to each “receiving” point. In this instance, none of the shippers or receivers communicate to exchange transport obligations, leading to inefficiency in shipping the mass from one probability distribution to the other. Note however that this does not correspond to the *least* efficient shipping. (We could obtain this by switching the infimum in equation 2.1 to a supremum. This quantity has a number of interesting properties that are explored in Chapter 6.) Rather, the naive transportation distance corresponds to each supplier acting individually, without concern for anything other than shipping its own mass.

Let  $f$  and  $g$  be the density functions of our distributions  $\mu$  and  $\nu$ . We define the naive transportation distance:

$$d_{\text{NT}}(\mu, \nu; d_{\Omega}) = \int_{\mu} f(x) d_{\text{KW}}(x, \nu; d_{\Omega}) dx \quad (2.7)$$

$$= \int_{\mu} \int_{\nu} f(x) g(y) E[d_{\Omega}(x, y)] dx dy \quad (2.8)$$

$$= \int_{\nu} g(y) d_{\text{KW}}(\mu, y; d_{\Omega}) dy = d_{\text{NT}}(\nu, \mu; d_{\Omega}) \quad (2.9)$$

Note that this definition employs a degenerate case of  $d_{\text{KW}}$ , namely where one of the distributions is a single point ( $x$  or  $y$ ). In this case,  $d_{\text{KW}} = d_{\text{NT}}$ , as no optimization is possible and the naive distance is the best one can obtain.

The discrete form of naive transportation distance  $d_{\text{NT}}$  over weighted point sets corresponding to discrete distributions is defined as:

$$d_{\text{NT}}(A, B; d_{\Omega}) = \sum_{i=1}^m \sum_{j=1}^n p_i q_j d_{\Omega}(a_i, b_j) = d_{\text{NT}}(B, A; d_{\Omega}) \quad (2.10)$$

The naive distance is the weighted sum of the ground distances between each individual point and the entirety of the other sample. It is straightforward to directly calculate this quantity, requiring  $O(k^2)$  time, where  $k = \max(m, n)$ .

### Point Set Similarity $\text{SIM}$

The Similarity  $\text{SIM}$  between two distributions  $\mu$  and  $\nu$  is defined simply as the ratio of two metrics above subtracted from 1:

$$\text{SIM}(\mu, \nu; d_{\Omega}) = 1 - \frac{d_{\text{KW}}(\mu, \nu; d_{\Omega})}{d_{\text{NT}}(\mu, \nu; d_{\Omega})} \quad (2.11)$$

Over weighted point sets  $A$  and  $B$ , the discrete form follows as:

$$\text{SIM}(A, B; d_{\Omega}) = 1 - \frac{d_{\text{KW}}(A, B; d_{\Omega})}{d_{\text{NT}}(A, B; d_{\Omega})} \quad (2.12)$$

For notational convenience we will omit  $d_\Omega$  when it is clear which distance metric is being used as the ground distance. It should be kept in mind however that  $d_{KW}$ ,  $d_{NT}$ , and  $SIM$  all depend on  $d_\Omega$ .

We will call the ratio  $\frac{d_{KW}(A,B)}{d_{NT}(A,B)}$  itself similarity *distance*, or  $d_{SIM}$ . The subtraction from 1 for  $SIM$  is done so that the final measure is one of similarity, not distance. The ratio  $\frac{d_{KW}}{d_{NT}}$  is low for high overlap (due to high benefits gained by cooperation), and high for low overlap. Similarity between the point sets therefore is the inverse of the optimization gained through cooperation. We discuss below some properties of  $SIM$ .

### Is This Normalization?

$SIM$  measures the amount of optimization provided by cooperative vs. independent, naive transportation; intuitively, it measures the spatial overlap between two weighted point sets. One might ask how else similarity might be computed from  $d_{KW}$ . A number of schemes have been devised to rescale data in order to normalize it (see Stolcke et al. (2008) for overviews and empirical evaluations). We compared  $1 - SIM$  with linear scaling; sample mean and variance normalization; sample mean normalization; sample variance normalization; Gaussianization, and Distribution matching. It was straightforward to find examples (see Figure 2.9) for all of these where they did not capture any notion of spatial overlap.

### 2.4.3 Properties and Behavior

The ratio  $\frac{d_{KW}(A,B)}{d_{NT}(A,B)}$  measures *the optimization gained by adding cooperation* when moving the source  $A$  onto the sink  $B$ . Thus, it is a dimensionless quantity that ranges between zero and one. This is because  $d_{KW}$  can never exceed  $d_{NT}$ ; even if there is no benefit to cooperation, it will be equal to  $d_{NT}$  at worst. For the upper bound, note that  $d_{KW}$  is a nonnegative distance, so that  $SIM$  can never exceed

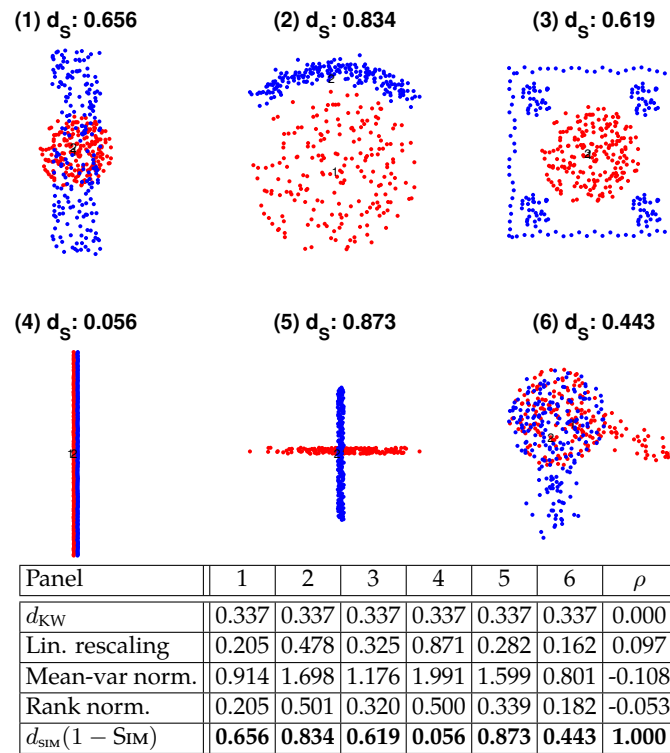


Figure 2.9: **S<sub>IM</sub> is not normalization.** All six examples in this figure were constructed to have the same  $d_{KW}$  and Earth Mover's Distance (= .337), between the blue and red point sets, while having markedly different spatial properties from each other. This is reflected in their similarities as measured by  $S_{IM}$ , as shown above each example. Here, since we are comparing normalizations of *distance*, we turn  $S_{IM}$  into a distance measure by subtracting it from one and denoting it  $d_{SIM}$  (similarity distance). The table further illustrates that one cannot simply normalize  $d_{KW}$  to obtain the measure provided by  $S_{IM}$ . The final column shows Pearson correlation coefficients of each normalization with similarity distance, demonstrating that none of them capture the notion of spatial overlap.

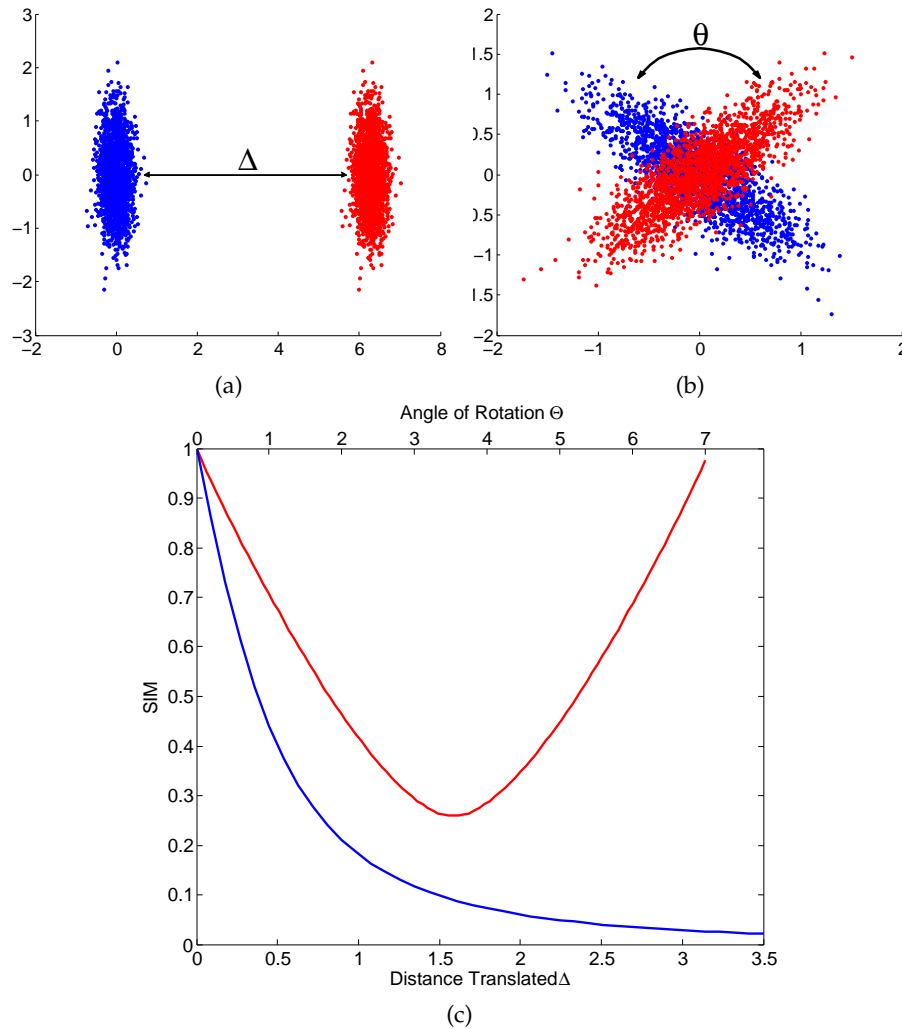


Figure 2.10: **SIM behavior.** Panels (a) and (b) contain plots of two point sets in varying degrees of separation. In panel (a), the point sets are separated in space due to translation and in panel (b) due to rotation. The graph at the bottom plots  $S_{IM}$  as a function of separation distance (a) and rotation angle (b) between the two point sets shown at the top. As can be seen,  $S_{IM}$  shrinks non-linearly as the distance between the point sets or the angle between them increases and then quickly approaches its asymptotic limit of 0. This figure is adapted from Coen (2006).



1. For clarity, let us examine  $S_{\text{IM}}$  at its two extremes. If  $S_{\text{IM}}(A, B) = 1$ , then  $d_{\text{KW}}(A, B) = 0$ , implying the maximally cooperative distance between  $A$  and  $B$  is zero. This can occur only when  $A = B$ ; i.e. when they perfectly overlap. This means each source is co-located with a sink expecting precisely as much mass as it produces.

In contrast, suppose  $S_{\text{IM}}(A, B) \rightarrow 0$ . This tells us that cooperation does not help during transportation. This occurs when  $A$  and  $B$  are so far apart that the points in  $A$  are much closer to other points in  $A$  than those in  $B$  and vice-versa. Thus, cooperation does not yield any significant benefit. In this case,  $d_{\text{KW}}(A, B) \rightarrow d_{\text{NT}}(A, B)$ , implying  $S_{\text{IM}}(A, B) \rightarrow 0$ . As  $d_{\text{NT}}(A, B) \geq d_{\text{KW}}(A, B)$  by definition, this provides the lower bound for  $S_{\text{IM}}(A, B)$  of 0. We see this in Figure 2.10, where  $S_{\text{IM}}$  between the two illustrated point sets quickly approaches 0 as they are separated. Conversely, as the point sets increasingly overlap, the similarity measure  $S_{\text{IM}}$  approaches 1 rapidly.

$S_{\text{IM}}$  is also scale invariant: it only measures the optimization gained, not the amount of work that must be performed. Thus, the actual size of the distributions makes no difference, as illustrated in Figure 2.6. Finally, we note for completeness that  $d_{\text{NT}} \neq 0$  except in the pathological case where both distributions are identical single points. In that case we will set  $S_{\text{IM}}$  to be 1 by definition.

#### **Note on using $S_{\text{IM}}$ as a kernel**

$S_{\text{IM}}$  is not an inner product in any space; in fact, the space of point sets themselves is not even a vector space (for example, if set union is taken to be the analog of the addition operation, then there can be no additive inverse). However, this is not necessarily a problem for the use of  $S_{\text{IM}}$  in downstream optimization functions since as noted in Burges (1998) it may be the case that a positive semidefinite Hessian matrix results for a given training set. This has indeed been the case for all experiments run during the course of this research. The

eigenvalues of every Gram matrix were checked for negative values before proceeding to the learning step, with an error to be triggered if there were any. This error was not triggered in any of the extensive experiments conducted with  $S_{IM}$ .

#### 2.4.4 Computational complexity

The complexity of computing  $S_{IM}$  is dominated by the Kantorovich-Wasserstein distance  $d_{KW}$ , which is a well-studied problem; using the Hungarian method has worst case complexity  $O(n^3)$  (Li, 2010) in unrestricted metric spaces. Recently a number of linear or sublinear time approximation algorithms have been developed for this problem and several variations, e.g., Li (2010); Do Ba et al. (2011); Pele and Werman (2009); Andoni et al. (2009). We have tested our implementation, which uses the transportation simplex algorithm, over several hundred thousand pairs of point sets drawn from standard statistical distributions and real world data sets. The runtime has expected time complexity of  $(1.38 \times 10^{-7})n^{2.6}$  seconds, fit with an  $R^2$  value of 1, where  $n$  is the size of the larger of the two point sets being compared. (We are particular to provide the quadratic coefficient, rather than describe the runtime using order notation, as its small value is what allows this approach to be used on larger scale problems.)

#### 2.4.5 Hyperclustering

Because  $S_{IM}$  measures the relative density overlap between point sets, it is not overly sensitive to their exact numbers or locations. We may use this intuition to approximate  $S_{IM}$  by grouping nearby points into a single weighted point. We call these groups of nearby points “hyperclusters” and construct them by recursively splitting the original point sets via k-means clustering (MacQueen et al., 1967) until the maximum interpoint distance within each hypercluster is

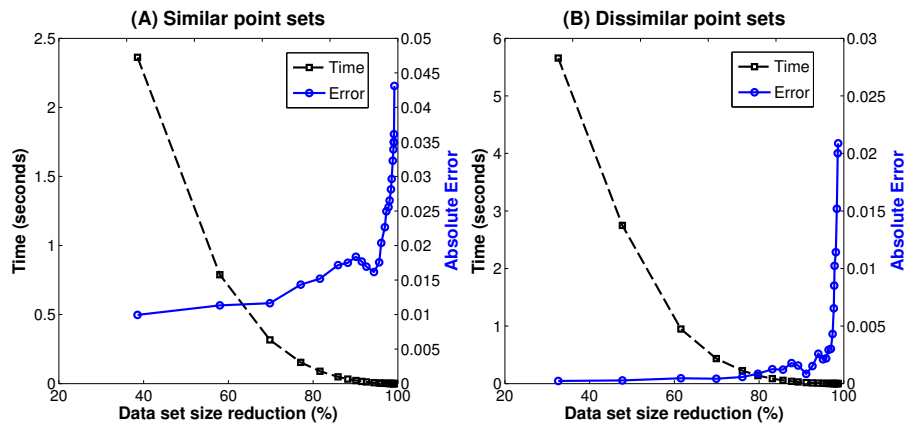


Figure 2.11: **Hyperclustering.** Error in  $S_{IM}$  when approximated by hyperclustering, averaged over 30 runs. In (A), we sample two sets of size 1000 from the same distribution. The exact  $S_{IM}$  value between them is 0.892, which takes 14.3 seconds to compute precisely. We vary the number of hyperclusters, corresponding to a reduction in problem size, and plot the error and overall computation time. In (B), we sample two sets of size 100 from poorly-overlapping distributions. The actual  $S_{IM}$  is 0.121, which takes 16.12 seconds to compute precisely. Note in both cases there is negligible loss in accuracy even when the point set size is reduced by up to 80%. This figure is taken from Coen, Ansari, and Fillmore (2011).

less than a specified threshold. This approximation method was first proposed by Coen (2006). In Figure 2.11, we show how the error and runtime change for a pair of point sets as the number of hyperclusters change. Empirically, this technique allows  $S_{IM}$  to be approximated closely for sets of millions of points. For example, precisely computing  $S_{IM}$  for point sets of size 100,000 would take almost 16 days, but an approximate answer can be computed in 46.9 seconds to within 0.01 of the true value. In extensive experimentation with this approximate form of  $S_{IM}$ , errors of up to 0.05 have little effect and correspond to natural variation in samples drawn from the same distribution.<sup>4</sup>

<sup>4</sup>We are able to determine this by having solved for the value of  $S_{IM}$  analytically for samples drawn from several common statistical distributions, thereby providing a way to evaluate approximations.

## 2.5 Density Overlap Kernel

The *density overlap kernel* is a novel kernel I developed to quantify the spatial overlap of two point sets. It is based on modelling the point densities of both point sets as a continuous distribution and defining a value for their “overlap” via an inner product. It is a provable Mercer kernel, so that this kernel may be used in downstream optimization algorithms with a guarantee of a bounded dual objective function. This kernel (1) has moderate computational cost, (2) makes no distributional assumptions, and (3) is simple in both concept and implementation. It also permits the user to trade simplicity for speed in a controlled and principled way. Given two point sets, we first construct a *non-parametric* density estimate from each point set. The kernel between the point sets is then defined as the inner product in  $L^2$  between the density estimates. The intuition behind this definition can be seen in Figure 2.12. Similar point sets, like those in panel (A), lead to similar density estimates, with a large inner product, while dissimilar point sets like those in panel (B) lead to dissimilar density estimates with a small inner product.

Some advantages that this method offers over other kernel methods discussed in Section 2.3 are as follows:

- Due to the fact that this method compares point sets using continuous density estimates as surrogates, we avoid the need to explicitly compute a matching (or a flow) between points from one set to points in the other set in order to compute the kernel value between two point sets.
- Using Gaussian kernels as nonparametric density estimates the overall kernel between point sets can be computed via a closed form which can be evaluated as a simple sum of Gaussians, leading to moderate computation cost.

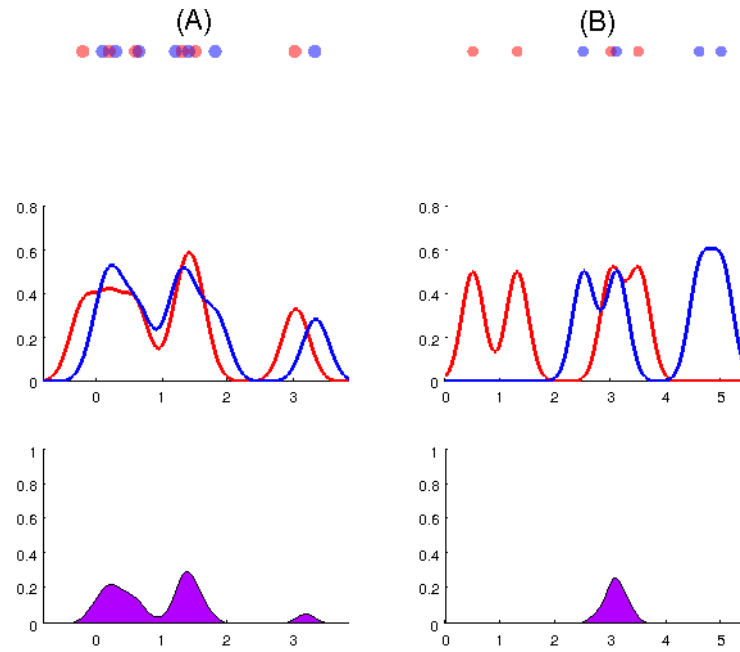


Figure 2.12: **Density overlap kernel.** (A) The top panel shows two point sets, represented in blue and red respectively. The middle panel shows their individual kernel density functions, while the bottom panel shows the value of the dot product of their functions. The density overlap kernel is the area covered by the dot product in the bottom kernel. (B) A similar set of panels as (A), this time shown for two point sets that have much lesser overlap. Note that the areas of the bottom panels capture the intuition that the two pairs of point sets at the top are relatively similar and dissimilar respectively.

- There is no need to make strong distributional assumptions about the data. This method is thus able to detect even subtle differences between point sets.

### 2.5.1 Formal Definition

The density overlap kernel defines an inner product  $K(A, B)$  on the space of point sets (each containing points in  $\mathbb{R}^d$ ) as follows:

#### Step 1: Mapping point sets to continuous functions

We use kernel density estimation to construct a representation for a given point set. In this way we move from a discrete representation of a point set that has support only at the points it contains to a continuous representation that captures its density and has support in the whole space, and therefore is comparable to representations of other point sets.

Specifically, to each finite point set  $A = \{\mathbf{a}_i\}_{i=1}^{n_A} \subset \mathbb{R}^d$  we associate the kernel density estimate  $f_A$ , defined as

$$f_A(\mathbf{z}) = \frac{1}{n_A} \sum_{i=1}^{n_A} e^{-\frac{\|\mathbf{z} - \mathbf{a}_i\|_2^2}{2\sigma^2}}.$$

Here we use the unnormalized Gaussian kernel  $e^{-\frac{\|\mathbf{z} - \mathbf{a}_i\|_2^2}{2\sigma^2}}$  where  $\sigma$  is bandwidth, and can be chosen either based on knowledge of the data set, cross-validation, or by an automated method such as Botev et al. (2010) that chooses the bandwidth minimizing mean integrated square error in a purely nonparametric and data-driven way.

#### Step 2: Kernel Computation

A point set  $A$  is now represented in the form of a kernel density function  $f_A$  whose value at any point  $\mathbf{z}$  in  $\mathbb{R}^d$  is given by the sum of its kernels with respect to all points in  $A$ . We define the *density overlap kernel*  $K$  between finite point sets  $A, B \subset \mathbb{R}^d$  to be the inner product in  $L^2(\mathbb{R}^d)$  of the associated density estimates:

$$K(A, B) = \int_{\mathbb{R}^d} f_A(\mathbf{z}) f_B(\mathbf{z}) d\mathbf{z}$$

To compute  $K(A, B)$  for any point sets  $A$  and  $B$  of cardinalities  $n$  and  $m$  respectively, let  $A = \{\mathbf{a}_i\}_{i=1}^n$ ,  $B = \{\mathbf{b}_j\}_{j=1}^m$ . A basic algorithm for computing  $K(A, B)$  is obtained by transforming the functional inner product  $\int f_A(\mathbf{z})f_B(\mathbf{z})d\mathbf{z}$  into a closed-form discrete summation as follows:

$$\begin{aligned}
\langle f_A, f_B \rangle &= \int_{\mathbb{R}^d} \left( \frac{1}{n} \sum_{i=1}^n e^{-\frac{\|\mathbf{z}-\mathbf{a}_i\|_2^2}{2\sigma^2}} \right) \left( \frac{1}{m} \sum_{j=1}^m e^{-\frac{\|\mathbf{z}-\mathbf{b}_j\|_2^2}{2\sigma^2}} \right) d\mathbf{z} \\
&= \int_{\mathbb{R}^d} \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m e^{-\frac{\|\mathbf{z}-\mathbf{a}_i\|_2^2 + \|\mathbf{z}-\mathbf{b}_j\|_2^2}{2\sigma^2}} d\mathbf{z} \\
&= \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \int_{\mathbb{R}^d} e^{-\frac{\|\mathbf{z}-\mathbf{a}_i\|_2^2 + \|\mathbf{z}-\mathbf{b}_j\|_2^2}{2\sigma^2}} d\mathbf{z} \\
&= \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \int_{\mathbb{R}^d} e^{-\frac{2\|\mathbf{z}-\frac{\mathbf{a}_i+\mathbf{b}_j}{2}\|_2^2 + \frac{\|\mathbf{a}_i-\mathbf{b}_j\|_2^2}{2}}{2\sigma^2}} d\mathbf{z} \text{ (by rewriting)} \\
&= \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m e^{\frac{\|\mathbf{a}_i-\mathbf{b}_j\|_2^2}{4\sigma^2}} \int_{\mathbb{R}^d} e^{-\frac{2\|\mathbf{z}-\frac{\mathbf{a}_i+\mathbf{b}_j}{2}\|_2^2}{2\sigma^2}} d\mathbf{z} \\
&= (\sigma\sqrt{\pi})^d \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m e^{-\frac{\|\mathbf{a}_i-\mathbf{b}_j\|_2^2}{4\sigma^2}} \text{ (by the identity } \int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi})
\end{aligned} \tag{2.13}$$

This method has  $O(nmd)$  computational complexity. It is helpful to see the final sum above as being the average entry in a  $n \times m$  matrix coincidentally also consisting of kernels, albeit with a bandwidth that is scaled up by a factor of  $\sqrt{2}$ .

## 2.5.2 Properties and Behavior

The variation of the density overlap kernel for two point sets with progressively larger separations is shown in Figure 2.13. As the distance between the point sets grows, the kernel value drops sharply to zero. As the distance shrinks, the value approaches 1.

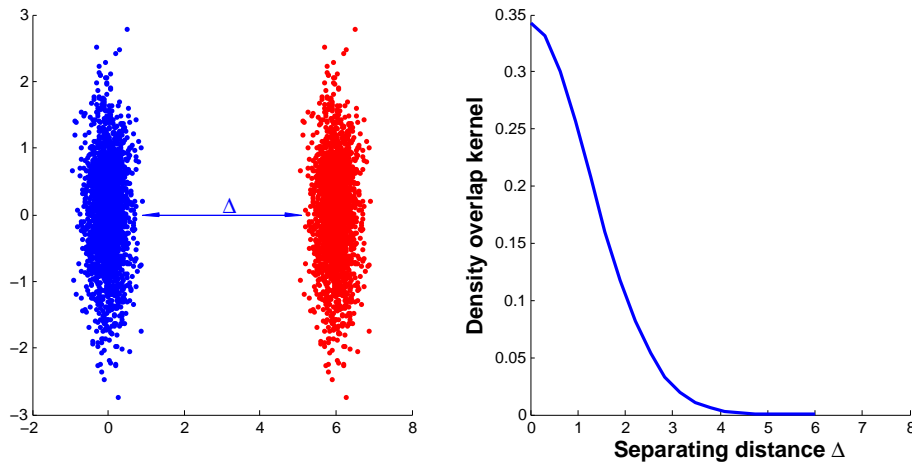


Figure 2.13: **Density overlap behavior.** The graph on the right plots the value of the density overlap kernel as a function of separation distance between the two point sets shown on the left. As can be seen, the value shrinks non-linearly as the distance between the point sets increases and then quickly approaches its asymptotic limit of 0.

A probability density  $f$  in general need not necessarily be an  $L^2$ -function, i.e., it need not be the case that  $\int_{\mathbb{R}} f^2 dx$  is finite. For example, if  $f$  is the Cauchy density function  $\int_{\mathbb{R}} f^2 dx = \infty$ . However, our kernel density estimates based on finite point sets are elements of  $L^2$ , since maps of the form  $x \mapsto e^{-\|x-y\|_2^2/2\sigma^2}$  are in  $L^2$ , and our kernel density estimates are finite sums of these. Our kernel  $K$  defines an inner product on the density estimates and therefore on the associated finite point sets. Finally, since  $K(A, B)$  is defined as an inner product in  $L^2$ , a Hilbert space, it follows that  $K$  is a valid kernel.

### 2.5.3 Approximation Schemes

In its exact form the dot product between kernel density estimates requires computing  $\Omega(n^2)$  entries and taking their average. In this section we propose one new method to approximating the value of the kernel by taking advantage of structure in the data set when present and also describe other existing approaches.



### Fast Gaussian Summation

The closed form obtained in Eq. 2.13 is a type of “Gaussian summation” of the form  $\sum_i \sum_j e^{-\frac{\|x_i - y_j\|}{2\sigma^2}}$ . During the past decade and earlier several algorithms have been proposed to approximate sums of this form quickly. These approaches can be grouped into three main categories:

- One set of methods (e.g. Gray (2003)) takes advantage of sparsity in the data by constructing k-d trees (Bentley, 1975) that partition the space  $\mathbb{R}^d$  in order to avoid having to compare all pairs of points. The tree is then walked over using a priority queue with priority based on upper and lower bounds on the Gaussian sums. When the upper and lower bounds are within a pre-specified error tolerance the walk terminates. These methods are effective when sparsity is actually present in the data and the point sets are large, but they have high overheads and in fact take longer than exact methods for relatively small point sets ( $< 1000$  points).
- Another set of methods, e.g. Lee and Gray (2006), expands sums of exponentials into a series of simpler terms, applies various algebraic transformations, and then truncates the series to reduce the number of computations in a controlled way. These methods are practical only for data of low dimensionality ( $< 3$ ), but they often achieve substantial speedups for any number of points when the dimensionality is small.
- More recently, Monte Carlo methods have found favor in the computation of these sums (Holmes et al., 2008; Lee and Gray, 2009). Through careful sampling these methods are able with high probability to approximate the sum to arbitrarily high accuracy, in time that is constant with respect to the number of points in the point sets. These methods also have high overhead and only yield gains in running time at high point set sizes ( $\geq 50000$ ).

### Direct Truncation

The approaches above are well-motivated theoretically and effective in practice in settings they are designed for (one-off computations on very large point sets in low dimensions). In the training phase of a kernelized learning algorithm with  $N$  training instances, kernel evaluations need to be performed a large number ( $\Omega(N^2)$ ) of times. Moreover in the applications discussed in following chapters the point sets have moderate to high dimensionality (10–256) and low to moderate cardinalities (1–1000). Thus constant overhead can accumulate and be a significant problem in practice. This motivates the following novel approximation scheme which also takes advantage of sparsity in the data.

Similar to Gray (2003) we observe that when there is even mild sparsity present, a number of terms in the sum (2.13) are very close to zero relative to other terms. For example, consider three points  $\mathbf{a}$ ,  $\mathbf{b}$ , and  $\mathbf{c}$  lying in  $(0, 1)$  with coordinates 0.1, 0.2 and 0.8 respectively. For  $\sigma = 0.1$ , the relative contribution to the overall value from the pair  $(\mathbf{a}, \mathbf{b})$  compared to the pair  $(\mathbf{a}, \mathbf{c})$  is

$$\frac{e^{-\frac{0.1^2}{4(0.1)^2}}}{e^{-\frac{0.7^2}{4(0.1)^2}}} = e^{12} \approx 162755.$$

Therefore instead of computing the contribution from the pair  $(\mathbf{a}, \mathbf{c})$  we can assign it a value of zero with little cost to accuracy.

In fact this can be done for all pairs of points separated by more than  $t\sigma$  for a suitable value of  $t$  given a kernel bandwidth  $\sigma$ : the ratio between the contribution to the overall kernel value of a pair of points  $t\sigma$  away from each other, and the contribution of a pair of points  $\sigma$  away, is  $e^{-\frac{t^2-1}{4}}$ . compared to a point one  $\sigma$  away. The value of  $t$  can be chosen appropriately depending on the value of  $\sigma$  and the level of accuracy desired (for example,  $t = 5$  achieves similar accuracy to the exact version in our experiments).

In the multi-dimensional case we make the following observation: if two points are separated by  $t\sigma$  in *any one dimension*, the relative contribution of that pair can similarly never exceed  $e^{-\frac{t^2-1}{4}}$ , since multiple dimensions only further reduce the value in the exponent. This suggests that dimensions can be dealt with independently and inspires our approximation.

Our approximation is as follows: We would like to compute  $K(A, B)$  for point sets  $A$  and  $B$ . First, one of the input point sets (say  $A$ ) is sorted along each dimension, keeping track of the original indices. Then for every point in set  $B$ , a binary search can be performed for each coordinate to determine its closest point (in that dimension) and a fan-out search to determine indices of neighboring points lying within  $t\sigma$ . These sets of neighboring points in each dimension are then intersected to find those points within  $t\sigma$  distance of the point in  $B$  in all dimensions.

This is summarized in the following algorithm:

```

INPUT:  $A, B, \sigma, t$ 
 $val := 0$ 
 $SortedA := sort(A)$ 
forall the  $\mathbf{b} \in B$  do
   $ClosestPts := FindClosestPoints(\mathbf{b}, SortedA, \sigma, t)$ 
  forall the  $ClosePoint \in ClosestPts$  do
     $val := val + e^{-\frac{\|\mathbf{b} - ClosePoint\|_2^2}{4\sigma^2}}$ 
  end
end
return  $val$ 

```

**Complexity Analysis** Let  $|B| = m$  and  $|A| = n$ , and let the dimensionality of points in each set be  $d$ . The initial sorting step is  $\Omega(nd \log(n))$ . Let  $c$  be the average number of neighbors in  $A$  lying in the  $t\sigma$ -neighborhood of a point in  $B$ . Then the *FindClosestPoints* procedure takes  $O(d(\log(n) + c))$  amortized time and is run  $m$  times. The total running time therefore is  $O(d(n \log(n) + m \log(n) + mc))$ . Assuming  $d < m < n$ , the running time is  $O(d(n \log(n) + mc))$ .

If the sparsity assumption is stated as “the number of points lying in the  $t\sigma$ -neighborhood of any point is at most  $O(\log(n))$ ” and  $t$  is chosen to satisfy this constraint, the final runtime complexity is  $O(nd \log(n))$ .

**Error Bounds** The final computation is the average of  $mn$  entries. The error contribution of each entry is 0 if the point pair it corresponds to are less than  $t\sigma$  apart in each dimension, since in that case the exact value is computed. If not, we approximate that entry to be 0, and the error arising out of that term is its value itself, at most  $e^{-\frac{t^2}{4}}$ . If  $c$  is defined as in the previous section, the absolute error is then upper-bounded by  $\frac{mn-mc}{mn} e^{-\frac{t^2}{4}}$ .

Obtaining a tight bound for the relative error is a harder task since there is no efficient way of calculating a reasonable lower bound estimate for the value of the function to be computed, and most prior work in estimating kernel computations has concentrated on developing absolute error bounds (see Lee and Gray (2006) for example).

## 2.6 Lift Kernel

Lifting is a recently proposed spatially sensitive metric between clusterings (Raman et al., 2011) that can be adapted for point set similarity. It applies a map  $\tilde{\Phi}$  to each data point, transforming it into an element of a  $\rho$ -dimensional approximation of a reproducing kernel Hilbert space (RKHS). The application of this map to a point is called “lifting” by the authors. A point set is represented as an element of this space by summing up the lifted representation of each data point. The summed vector is normalized to unit length to eliminate differences caused by differing set cardinalities. The similarity between two point sets  $X$  and  $Y$  is defined as the dot product between the vectors representing them. The key to the utility of this kernel is the choice of the map; it is chosen so that the lifted points preserve information about spatial relationships between

their source points. This procedure is illustrated for three sample point sets in Figure 2.14.

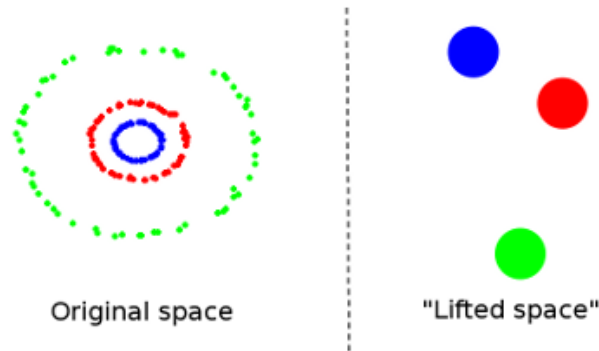


Figure 2.14: **Lift kernel.** This figure provides an illustration of the “lift” operation described in Section 2.6. The three clusters of points in the original two-dimensional space (colored blue, red, and green respectively) are transformed into singular points in a much higher  $P$ -dimensional space. An approximation of their relative positions in a two-dimensional projection is shown on the right hand side of the figure. Notice that the “distance” relationships between the clusters on the left are preserved in the new space, in that blue is closer to red than green which is furthest from the others.

The map  $\Phi$  is inspired from a similar map in random Fourier features (Rahimi and Recht, 2007), which was introduced to transform data into a form where linear operations can approximately simulate kernel evaluations for certain choices of kernels.  $\tilde{\Phi}$  (the “lifting” function) is applied to each data point in  $\mathbb{R}^d$ , transforming it into an element of  $\mathbb{R}^{2\rho}$ , a  $2\rho$ -dimensional approximation of a reproducing kernel Hilbert space (RKHS). This mapping is randomized and similarity-preserving; a shift-invariant kernel in the original space is approximately equal to the inner product in the new space:

$$K(x, y) \approx \langle \Phi(x), \Phi(y) \rangle$$

where the approximation can be made as precise as possible by varying the dimensionality ( $2\rho$ ) of the lifted space. For the kernel  $K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2}}$ , the approximate lifting map  $\hat{\Phi}_d : \mathbb{R}^d \rightarrow \mathbb{R}^{2\rho}$  is defined as follows:

$$\hat{\Phi}(\mathbf{x}) = [\cos(\omega_1 \mathbf{x}), \dots, \cos(\omega_d \mathbf{x}), \sin(\omega_1 \mathbf{x}), \dots, \sin(\omega_d \mathbf{x})]$$

for  $\mathbf{x} \in \mathbb{R}^d$  where elements of  $\omega_i$ 's are random and normally distributed, and

$$\langle \hat{\Phi}(\mathbf{x}), \hat{\Phi}(\mathbf{y}) \rangle \simeq K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2}}$$

for any  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ .

We extend this formulation of point set similarity to incorporate weights for each point, so that the final expression for similarity between two weighted point sets  $X = \{x_i, w_i\}$  and  $Y = \{y_i, v_i\}$  becomes  $\langle \frac{\hat{\Phi}(X)}{\|\hat{\Phi}(X)\|}, \frac{\hat{\Phi}(Y)}{\|\hat{\Phi}(Y)\|} \rangle$ , where  $\hat{\Phi}(X) = \sum_{\mathbf{x}_i, w_i} w_i \hat{\Phi}(\mathbf{x}_i)$ .

The complexity of this technique is  $O(nd\rho)$ , dominated by the cost of matrix multiplication. Empirically, we have observed that for best experimental results,  $\rho = O(nd)$ .

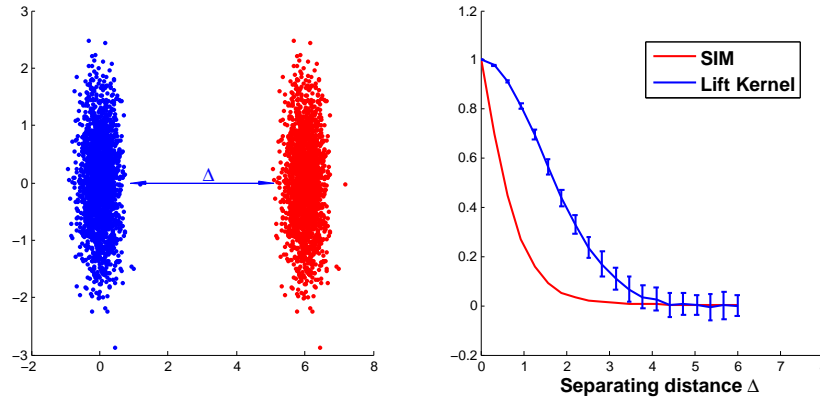


Figure 2.15: **Lift instability.** This figure shows the values of SIM and the lift kernel between two clusters as a function of their separating distance,  $\Delta$ . Due to the randomized nature of  $\tilde{\Phi}$ , the value of the lift kernel between any two clusters varies across multiple runs and can have a standard error of up to 5%.

Due to the randomized nature of  $\tilde{\Phi}$ , the distance between any two clusters varies across multiple runs and can have a standard error of up to 5%, as shown in Figure 2.15. Additionally, because a cluster is represented as the sum of  $\tilde{\Phi}$ -maps of points contained within it, it is no longer one-to-one, and

it is possible for multiple clusters to share representations despite being very different from one another. If  $\rho < nd$  where  $n$  is the number of points in a data set and  $d$  is its dimensionality, the mapped space is not rich enough to have unique representations for every possible cluster. In fact, for certain choices of  $\rho$ , one can construct two very different clusters with identical or nearly identical representations. To a certain extent, this problem can be alleviated by choosing high values for  $\rho$ , which in turn leads to higher run-times but still leaves the problem of variation in returned values across multiple runs even for these high values of  $\rho$ .

## 2.7 Examples and Comparisons

Having detailed a number of point set comparison measures we return to a motivating problem at the beginning of this work: classification of samples from probability distributions. One of the common assumptions in machine learning is that data from different classes originate from different underlying distributions. When data are represented in the form of point sets, the classification problem becomes one of being able to differentiate between *samples* of points from differing distributions. Below we conduct experiments wherein we sample sets of points from two different multivariate probability distributions and evaluate the utility of various spatial and non-spatial point set comparison methods in classifying these point sets.

### Experiment 1

Consider the pair of 1-dimensional distributions shown in Figure 2.16 where the densities of both distributions are plotted along the  $y$ -axis; one in blue and the other in red. The distribution shown in blue is a beta distribution (parameters  $\alpha = 0.8$  and  $\beta = 1.4$ ) and the one in red is a truncated gamma distribution (shape  $k = 1.83$  and rate  $\theta = 0.19$ ). The parameters for these distributions

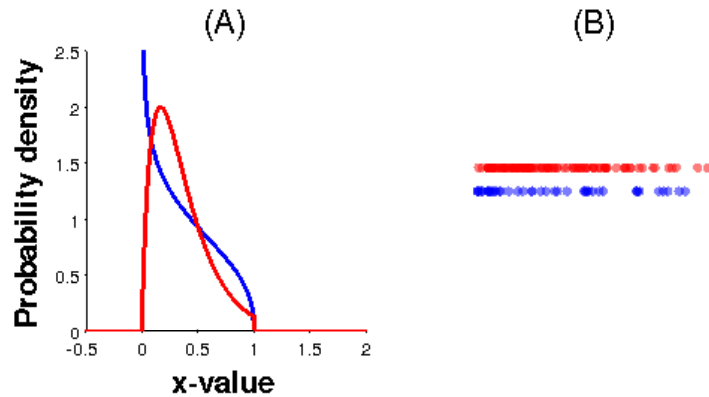


Figure 2.16: (A) Probability density function plots for Experiment 1 in Section 2.7. The plot in blue is a beta distribution with parameters 0.8 and 1.4, and the one in red is a truncated gamma distribution, with shape parameter 1.83 and rate parameter 0.19. (B) Two examples of point sets that are sampled from distributions shown in (A). This figure is adapted from Coen, Ansari, and Fillmore (2011).

were chosen so as to make both the means and variances of these distributions identical to one another respectively.

We sampled 100 point sets from each distribution, each containing a varying numbers of points between 30 and 60. We then trained a support vector machine (Shawe-Taylor and Cristianini, 2000) to separate between point sets originating from these two distributions, using  $S_{IM}$ , density overlap, lift kernel, Bhattacharyya kernel (Kondor and Jebara, 2003), and pyramid match (Grauman and Darrell, 2007) as similarity functions. Classification accuracy results on a holdout set of another 100 point sets from each distribution are tabulated in Table 2.1.

## Experiment 2

We replicate the experiment above with the two-dimensional distributions shown in Figure 2.17. The density functions in one dimension of both distribu-



	Accuracy	Time
Density overlap ( $\sigma = 0.05$ )	<b>93.5%</b>	9.51s
SIM	87%	33.4s
Pyramid match	85.8%	4.84s
Lift kernel ( $\rho = 400$ )	74%	45.25s
Bhattacharyya kernel	83%	120.9s

Table 2.1: Accuracy results for five point set similarity measures on classifying synthetic data in 1 dimension sampled from the distributions shown in Figure 2.16. 100 point sets of varying cardinality between 30 and 60 were sampled from each distribution and used to train a support vector machine classifier. This classifier was then tested on another 100 samples; the classification accuracies are shown in the table above. The time shown for density overlap includes a beam search for the  $\sigma$  parameter.

	Accuracy	Time
Density overlap ( $\sigma = 0.05$ )	<b>77.5%</b>	11.34s
SIM	76%	107.52s
Pyramid match	59.3%	5.3s
Bhattacharyya kernel	63%	116.79s
Lift kernel ( $\rho = 1000$ )	54.5%	79.48s

Table 2.2: Accuracy results for five point set similarity measures on classifying synthetic data in 2 dimensions sampled from the distributions shown in Figure 2.17. 100 point sets of varying cardinality between 30 and 60 were sampled from each distribution and used to train a support vector machine classifier. This classifier was then tested on another 100 samples; the classification accuracies are shown in the table above. The time shown for density overlap includes a beam search for the  $\sigma$  parameter.

tions are plotted along the  $x$ -axis, and along the other dimension on the  $y$ -axis. One distribution (in blue) follows a beta distribution with parameters (1.3, 1.3) in the  $x$ -coordinate and a normal distribution with mean 0.5 and variance 0.04 in the  $y$ -coordinate. The other distribution (in red) follows a uniform distribution with parameters (0, 1) in the  $x$ -coordinate and a beta distribution with parameters  $\alpha = 2.4$  and  $\beta = 2.4$  in the  $y$ -coordinate.

Similar to Experiment 1 above, we sampled 100 point sets from each distribution, each containing a varying numbers of points between 30 and 60. We

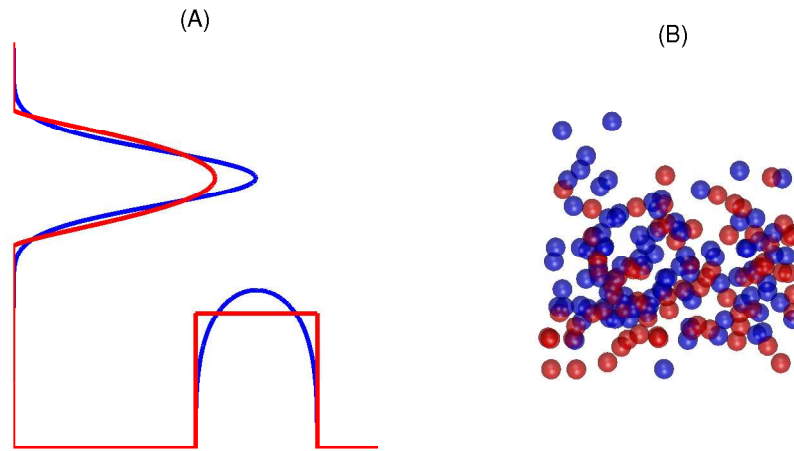


Figure 2.17: (A) Probability density function plots for Experiment 2 in Section 2.7. The plots in blue represent the density in each dimension of one distribution; it follows a beta distribution with parameters 1.3 and 1.3 in the  $x$ -dimension and a normal distribution with mean 0.5 and variance 0.04 in the  $y$ -dimension. The plots in red represent the per-dimension densities of the second distribution; along the  $x$ -dimension is a uniform distribution on  $[0, 1]$  and along the  $y$ -dimension a beta distribution with parameters 2.4 and 2.4. (B) Two examples of point sets that are sampled from distributions shown in (A). This figure is adapted from Coen, Ansari, and Fillmore (2011).

then trained a support vector machine to separate between point sets originating from these two distributions, using  $S_{IM}$ , density overlap, lift kernel, Bhattacharyya kernel (Kondor and Jebara, 2003), and pyramid match (Grauman and Darrell, 2007) as similarity functions. Classification accuracy results on a holdout set of another 100 point sets from each distribution are tabulated in Table 2.2.

### 2.7.1 Discussion

The distributions in the experiments above were designed so as to be extremely difficult to distinguish between. The means and variances along each dimension were calculated to be identical or almost identical. The differences between samples of points from these distributions are therefore likely to be subtle and

difficult to detect. As the results in Tables 2.1 and 2.2 show, it is the spatially aware comparison algorithms that are able to detect a classifying boundary with any reasonable degree of accuracy. It is the relative densities at different points in space that are the separating characteristic of point sets from different distributions; this is precisely what spatially sensitive measures aim to capture. We will see a similar pattern in the relative efficacies of spatial and non-spatial methods in the applications in following chapters.

### 3 CLUSTERING COMPARISON

---

Data clustering, the task of organizing data into groups or clusters based on similarity (Jain et al., 1999), is a classic means of discovery in science. The idea is that data within a cluster should be more similar to one another than to data in a different cluster (Raghavan, 1982). Figure 3.1 shows an example of a data set and an intuitive grouping of its points. These groupings — called “clusterings” or “partitions” — are constructed by algorithms implicitly or explicitly relying on measures of similarity or dissimilarity between data points. Surprisingly however, most extant ways of *comparing* clusterings — the subject of this chapter — entirely ignore the very measure that were instrumental in creating them, many times leading to very counter-intuitive results.

Clustering does not require the use of class labels associated with each point. Clustering algorithms therefore are a useful investigative tool when given large amounts of data that are unlabeled and/or are very costly to label. Due to the fact that data are unlabeled in this scenario, clustering is a form of *unsupervised* learning. Clustering can provide insight into the nature of the data set itself by identifying distinct structures and patterns in the data; this information can then be used in the design of a future supervised learning phase (Duda et al., 2012).

There are dozens of clustering algorithms (Jain et al., 1999) that employ diverse methods to discover clusters in data. The fundamental question of evaluating the output of clustering algorithms however is not quite as settled as in the world of supervised learning, where there exists a veritable suite of evaluation metrics such as accuracy, precision, recall,  $F_\beta$  scores, and ROC curves. It is difficult therefore to make absolute judgments about the value of a given clustering. Is it informative? Does it capture some intrinsic property present in a data set? And perhaps most vexing of all, how many clusters are appropriate?

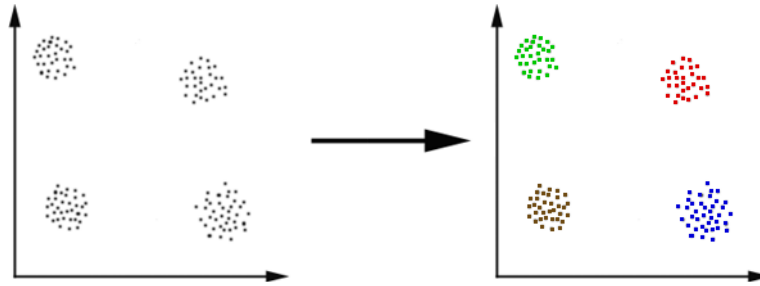


Figure 3.1: **Clustering.** This figure shows an example of a simple two-dimensional data set consisting of four well-separated groups of points. There are no labels associated with instances in this example; the figure on the left therefore colors each point identically. On the right is one possible clustering (among many) of these data. The points in different clusters are now differentiated by their color.

In the absence of labels, the only way of inducing an order preference for clusterings is by the inductive bias of the algorithm that produced them. There is no notion of a “best” clustering algorithm; the appropriate algorithm for a given problem depends upon some assumed inductive bias (Mitchell, 1980), in the absence of which all clusterings are equally valid, as formally described by Watanabe (1969) and Wolpert (1996).

Thus, without some a priori preference, we have no reason to prefer one clustering over another. However, this point immediately raises a basic question: how should one compare different clusterings? Rather than address the ill-defined evaluation of a given clustering head on, we turn to the more tractable question of how *similar* two clusterings are. Numerous approaches to this problem have been proposed (detailed in Section 3.2). Most of them however address the cluster assignments of the points alone without taking into account their spatial locations. In other words, these clustering comparison methods depend entirely on the *partitional* information in the clusterings and not on the spatial information of the points they contain. These comparison measures require that the clusterings are over identical data sets and sometimes even that the number of clusters in each output is identical.

As a corollary to the point above about there not being a “best” clustering algorithm, it is similarly not meaningful to state that one framework for comparing clusterings is objectively superior to another (Meila, 2005). Our goal then in this chapter is to present a more sophisticated view of comparing clusterings that incorporates information beyond partitional knowledge. Exploiting spatial information can provide a deeper and more nuanced understanding of the underlying structure in the data set.

In the remainder of this chapter I present a novel, principled, method called  $CDISTANCE$  for comparing clusterings both geometrically and partitionally, discuss related work, and provide comparisons with results on experiments that demonstrate the value added by spatial information, followed by an application of  $CDISTANCE$  to evaluating the stability of a clustering algorithm with respect

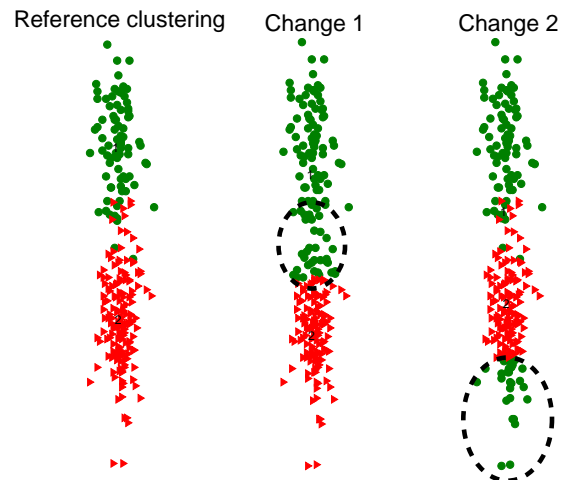


Figure 3.2: **Comparing Clusterings.** This figure displays three clusterings. Each one contains two clusters, whose members are indicated by green circles and red triangles. In the two changed clusterings, the circled points have been reassigned from the red to the green cluster. We might expect that the Reference Clustering is more similar to Change 1 than Change 2 because the modified points are *closer* to it. However, all methods from Rand (1971); Hubert and Arabie (1985); Dongen (2000); Fowlkes and Mallows (1983); Meila (2005) and Zhou et al. (2005) are incapable of distinguishing between the two changes. This figure is taken from Coen, Ansari, and Fillmore (2010).

to a particular data set. Finally, I present a new ensemble clustering algorithm enabled by a spatially aware analysis of clusters that **significantly outperforms** existing algorithms as measured by accuracy on labeled data sets. The following definition of a clustering<sup>1</sup> will be used throughout the remainder of this chapter:

**Definition 3.1.** *Clustering: A (hard) clustering  $\mathcal{A}$  is a partition of a data set  $X$  into a finite collection of  $K$  (point) sets  $X_1, X_2, \dots, X_K$  called clusters such that*

$$X_i \cap X_j = \emptyset \text{ for all } i \neq j, \text{ and } \bigcup_{k=1}^K X_k = X,$$

where  $X, X_1, \dots, X_K$ , are finite subsets of a metric space  $\Omega$  that has an associated distance metric  $d_\Omega$ .

We note it is unusual for a clustering comparison algorithm to utilize  $d_\Omega$ . While it is essential for clustering these points, the vast majority of algorithms compare clusterings solely in the space of cluster assignments rather than in  $\Omega$  directly.

### 3.1 Spatial Information in Clustering

Popular clustering algorithms, e.g.  $k$ -means clustering (MacQueen et al., 1967), spectral clustering (Ng et al., 2002), affinity propagation (Dueck and Frey, 2007), etc., take as input not only a collection of points to be clustered, but also a distance function on the space in which the points lie. This distance function may be specified implicitly and it may be transformed by a kernel, but it must be defined between all points in the data set and its properties are crucial to the clustering algorithm's result. The information provided by this function

---

<sup>1</sup>In situations where the verb and noun can be confused for each other, we will substitute the word "partition" in place of "clustering."

enables the clustering algorithm to apply its inductive bias and arrive at its preferred output.

In contrast, almost all existing clustering *comparison* techniques ignore the distances between points, treating clusterings as partitions of disembodied atoms. While this approach has merit under some circumstances, it seems surprising to ignore the distance function that was used to construct the clusterings. Doing so seems to discard what is in some sense the most basic information we have about the clusterings. Indeed, in Section 3.2, we exhibit a number of clusterings that have substantially different spatial properties but are indistinguishable by almost all previous clustering comparison techniques. One such example is presented in Figure 3.2. We have found only one other clustering comparison technique published prior to `CDISTANCE` that can distinguish between the leftmost reference clustering and its two modifications to the right.

### 3.1.1 Added Benefits

By incorporating spatial information into the clustering comparison measure, `CDISTANCE` provides several additional benefits. First, we are able to compare clusterings that are considered incomparable by many other techniques; specifically, we can compare clusterings:

1. over different sets of points
2. over different numbers of points, and
3. over different number of clusters

Only one other clustering comparison technique, published after `CDISTANCE` and borrowing its key idea, allows comparison under all such conditions, particularly conditions (1) and (2), which are largely unaddressed in the literature. In contrast to some other approaches, our approach also extends in a



straightforward way to soft (non-partitional) clustering. We briefly review some applications in which a distance between clusterings is useful:

- Because clustering is an unsupervised learning technique, comparing the output of clustering algorithms is difficult. In some cases, there may be a gold standard with which we would like our algorithm to agree. Measuring the distance between the gold standard partition and an algorithm's output provides important insight into whether the algorithm is suitable for a given domain.
- We can explore the stability of a clustering algorithm's results on a data set by repeatedly subsampling the data set and comparing the algorithm's results against each other. (Condition (1) in the previous paragraph is particularly useful here.)
- If the outputs of two clustering algorithms tend to agree on certain kinds of data, we may prefer to use the algorithm with lower computational complexity; comparing the algorithms' outputs helps us make this determination.
- Finally, for ensemble methods in clustering, we may employ a variety of clustering algorithms that exploit different mathematical properties of the data. Asking if their outputs are both partitionally and geometrically compatible adds an extra dimension for comparison.

### 3.2 Related Work

The question of comparing quality of clusterings has inspired many solutions over the past several decades, drawing from diverse mathematical tools from set theory to information theory to statistics. Below we detail classes of methods for comparing clusterings in the literature with representatives of each. In the

presentation of the similarity measures below, let  $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^D$  be the data set being clustered, and  $\mathcal{A} = \{A_1, \dots, A_n\}$  and  $\mathcal{B} = \{B_1, \dots, B_m\}$  two partitions of  $X$ .

### Set-theoretic methods

The majority of clustering comparisons and the most popular in terms of usage are methods that are based on constructing sets corresponding to cluster memberships of points or pairs of points and calculating their cardinalities. We call these methods “set-theoretic” in reference to their discrete natures and because they are the result of union and intersection operations on sets derived from membership information. Among the earliest of such methods is the Jaccard index (1901) (Ben-Hur et al., 2002), which measures the fraction of pair-wise cluster assignments on which the two partitions agree. It counts the numbers of pairs of points  $x_i$  and  $x_j$  that are clustered together in both clusterings<sup>2</sup>:

$$Jaccard(\mathcal{A}, \mathcal{B}) = \frac{\sum_{i,j} [x_i \text{ and } x_j \text{ are clustered together in both } \mathcal{A} \text{ and } \mathcal{B}]}{\sum_{i,j} [x_i \text{ and } x_j \text{ are clustered together in } \mathcal{A} \text{ or } \mathcal{B}]} \quad (3.1)$$

For the presentation of the following measures it is useful to predefine the following quantities:

---

<sup>2</sup>The formula uses Iverson brackets; it denotes a number that is 1 if the condition in square brackets is satisfied, and 0 otherwise.

$$\begin{aligned}
N_{ss} &= \sum_{i,j} [x_i \text{ and } x_j \text{ are clustered together in both } \mathcal{A} \text{ and } \mathcal{B}] \\
N_{dd} &= \sum_{i,j} [x_i \text{ and } x_j \text{ are not clustered together in } \mathcal{A} \text{ nor } \mathcal{B}] \\
N_{sd} &= \sum_{i,j} [x_i \text{ and } x_j \text{ are clustered together in } \mathcal{A} \text{ but not } \mathcal{B}] \\
N_{ds} &= \sum_{i,j} [x_i \text{ and } x_j \text{ are clustered together in } \mathcal{B} \text{ but not } \mathcal{A}]
\end{aligned}$$

In terms of the above quantities, the Jaccard index can be re-written as

$$Jaccard(\mathcal{A}, \mathcal{B}) = \frac{N_{ss}}{N_{ss} + N_{ds} + N_{sd}} \quad (3.2)$$

Subsequently, Rand (1971) famously proposed a method for comparing clusterings based on both similarities and differences in point assignments (the Jaccard index above only takes similarities into account). This metric, known as the Rand index, is calculated by the fraction of points – taken pairwise – whose assignments are consistent between two clusterings:

$$Rand(\mathcal{A}, \mathcal{B}) = \frac{N_{ss} + N_{dd}}{N_{ss} + N_{ds} + N_{sd} + N_{dd}} \quad (3.3)$$

Rand also discussed the notion of partition sensitivity to data perturbation, which was among the earliest applications of these types of comparison methods.

Techniques for a more restricted problem, one of comparing the hierarchical outputs of agglomerative clustering algorithms have inspired several set-theoretic solutions as well. The measure proposed by Sokal and Rohlf (1962) was inspired by the need to compare taxonomies from fields in biology such as phylogenetic trees. It computes for each pair of elements the height of their

lowest ancestor in both dendrograms; these measures are then arranged in a vector and the final measure is a correlation coefficient between these vectors. Farris (1973) derives a measure to compare hierarchical clusterings (also called dendrograms or trees) by counting the number of fragments into which the clusters of one dendrogram are broken into in another dendrogram. Fowlkes and Mallows (1983) proposed a measure to compare dendrograms by “cutting” them at different heights to get  $k = 2, 3, \dots, N$  clusters in each partition, and defined the similarity at each  $k$  to be

$$\text{Fowlkes-Mallows}(\mathcal{A}_k, \mathcal{B}_k) = \frac{N_{ss}}{\sqrt{(N_{ss} + N_{ds})(N_{ss} + N_{sd})}} \quad (3.4)$$

The approach taken by Rand of counting pair agreements has been built upon by many others who examined similarity between non-hierarchical partitions and formulated a number of novel distance metrics in the process. A survey of these early partition metrics can be found in Day (1981). For example, Hubert and Arabie (1985) addressed the Rand index’s well-known problem of overestimating similarity on randomly clustered data sets and introduced the Adjusted Rand Index to correct for chance agreement:

$$\text{AdjustedRand}(\mathcal{A}, \mathcal{B}) = \frac{N_{ss} + N_{dd} - I_{exp}}{N_{ss} + N_{ds} + N_{sd} + N_{dd} - I_{exp}} \quad (3.5)$$

where  $I_{exp}$  is the expected count for the quantity in the numerator of the Rand Index:

$$I_{exp} = \frac{N^2(N^2 + 1) - N(N + 1)(\sum_i |A_i|^2 + \sum_j |B_j|^2) + \sum_{i,j} |A_i \cap B_j|^2}{2N(N - 1)} \quad (3.6)$$

The Jaccard, Rand, and Hubert measures above are part of a general class of clustering comparison techniques based on tuple-counting or set-based membership. Other more recent measures in this category include the Mirkin,

van Dongen, Fowlkes-Mallows and Wallace indices, a discussion of which can be found in Ben-Hur et al. (2002) and Meila (2005).

### Hamming Distance

Lange et al. (2004) proposed a clustering similarity measure for the purpose of measuring stability of clustering solutions. A clustering is considered more stable if another set of similar data produces a clustering that is similar to the first one. The similarity measure proposed is a simple Hamming distance: it is the count of data points that occur in the same cluster in both clusterings, normalized by the size of the data set. Since there are no labels and therefore no gold standard correspondence of clusters between two clusterings, the final quantity is taken to be the minimum normalizing Hamming distance over all possible permutations of cluster correspondences. Note that for this particular measure we must have  $m = n$ , i.e. the two partitions must have the same number of clusters. Let  $L_{\mathcal{A}}$  be a vector of length  $N$  constructed from the cluster assignments of points in  $X$  according to partition  $\mathcal{A}$ , and similarly  $L_{\mathcal{B}}$  corresponding to  $\mathcal{B}$ . The Hamming distance between clusterings is

$$\text{Hamming}(\mathcal{A}, \mathcal{B}) = \min_{\pi \in \mathcal{G}_n} \frac{1}{N} \sum_{i=0}^N [L_{\mathcal{A}}(i) \neq \pi(L_{\mathcal{B}}(i))] \quad (3.7)$$

where  $\pi$  is a permutation of the set  $\{1, 2, \dots, n\}$  and the minimization is over the group  $\mathcal{G}_n$  of all such permutations.

### Variation of Information

The Variation of Information approach (Meila, 2005), defines an information theoretic metric between clusterings by considering partitions as points in a lattice. Here, the distance between partitions is defined by their relationship in the lattice. This distance measures the information lost and gained in moving from one clustering to another, over the same set of points. Perhaps its most

important property is that of convex additivity, which insures locality in the metric; namely, changes within a cluster (e.g. refinement) cannot affect the similarity of the rest of the clustering. This metric is defined as follows:

$$VI(\mathcal{A}, \mathcal{B}) = H(\mathcal{A}) + H(\mathcal{B}) - 2I(\mathcal{A}, \mathcal{B}) \quad (3.8)$$

where  $H$  is the entropy function:

$$H(\mathcal{A}) = - \sum_{i=1}^n \frac{|A_i|}{N} \log\left(\frac{|A_i|}{N}\right)$$

and  $I$  the mutual information function:

$$I(\mathcal{A}, \mathcal{B}) = \sum_{i=1}^n \sum_{j=1}^m \frac{|A_i \cap B_j|}{N} \log\left(\frac{|A_i \cap B_j|}{N} \frac{|A_i|}{N} \frac{|B_j|}{N}\right)$$

### Mallows Distance

Zhou et al. (2005) defined a distance between clusterings that is based on the Mallows distance between probability distributions (equivalent to the Kantorovich-Wasserstein distance, defined in Section 2.4.1). Clusters are represented as discrete probability distributions over the data set  $X$ , where the density at each data point is defined by the probability of a point belonging to that cluster. The distance between clusters is defined as the  $\ell_1$  distance between their corresponding membership distributions, and the distance between clusterings is defined as the Mallows distance (Section 2.4.1) between them.

This distance shares our motivation of incorporating some notion of distance into comparing clusterings. However, it is computed over a space of indicator vectors for each cluster, representing whether each data point is a member of that cluster. Thus, similarity over clusters is measured by their shared points and does not make any use of the geometries of the points themselves. The primary motivation for this work was to introduce soft clustering into the standard literature of clustering comparisons.

All the techniques above can be considered as being statistics derived from analysis of cluster assignments. That is, they are dependent only on the relative fractions of points in each cluster and cluster intersections. While these measures can robustly handle a different number of clusters in the clusterings, they cannot take into account a number of other desirable features as in Table 3.1.

	Rand	VI	Mallows	<b>CDISTANCE</b>	Hamming	ADCO
Different $k$	✓	✓	✓	✓	×	×
Different data	×	×	×	✓	×	✓
Spatially aware	×	×	×	✓	×	✓

Table 3.1: **Clustering comparison measures.** This table shows the ability of different clustering comparison measures to handle different numbers of clusters in two clusterings ( $k$ ), different underlying data sets, and whether they take into account the geometric properties of points in clusters. The measures in the columns of this table correspond to the Rand index (Rand, 1971) and associated set-theoretic methods, Variation of Information (VI) (Meila, 2005), the Mallows Distance based measure from Zhou et al. (2005), our method, Hamming distance (Lange et al., 2004), and ADCO (Bae et al., 2006).

### ADCO

Bae et al. (2006) present the first and only method we are aware of at the time of developing CDISTANCE that explicitly utilizes spatial information about points to compare two clusterings. This approach – called Attribute Distribution Clustering Orthogonality (ADCO) – first bins all data points being clustered along each dimension. It then computes the counts of points in each cluster in each bin; these counts are called the cluster-densities. Finally, the distance between two clusterings is defined as the minimal sum of pairwise cluster-density dot products (derived from the binning), with the minimization taken over all possible permutations of cluster correspondences between the clusterings. We note that this is in general not a feasible computation. The number of bins grows exponentially with the dimensionality of the space, and more importantly, examining all matchings between clusters requires  $O(n!)$  time, where  $n$  is the number of clusters.

Table (a)

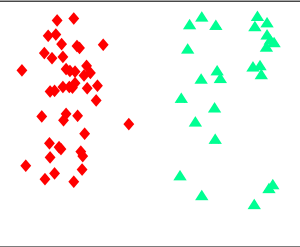
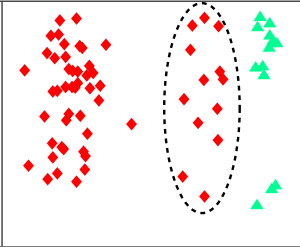
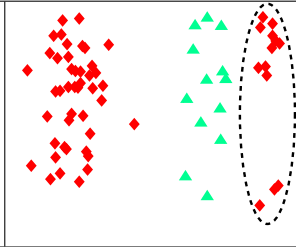
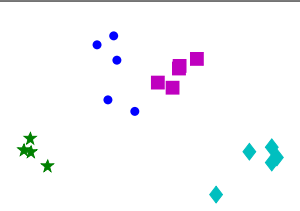
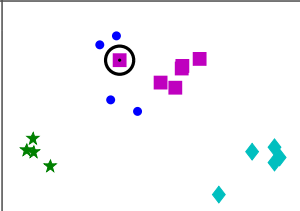
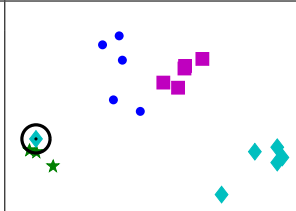
Ex.	Reference Clustering ( $\mathcal{R}$ )	Change 1	Change 2
1			
2			

Table (b)

Technique Name	Example 1			Example 2		
	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?
Hubert	0.38	0.38	×	0.04	0.04	×
1 – Rand	0.00	0.00	×	0.05	0.05	×
VI	5.22	5.22	×	11.31	11.31	×
Mallows	8.24	8.24	×	0.90	0.90	×
ADCO	0.11	0.17	✓	0.02	0.04	✓
<b>CDISTANCE</b>	<b>0.61</b>	<b>0.73</b>	✓	<b>0.06</b>	<b>0.18</b>	✓

Table 3.2: **Distances to Modified Clusterings.** Each row in Table (a) depicts a dataset with points colored according to a reference clustering  $\mathcal{R}$  (left column) and two different modifications of this clustering (center and right columns). For each example, Table (b) presents the distance between the reference clustering and each modification for the indicated clustering comparison techniques. The column labeled “?” indicates whether the technique provides sensible output.

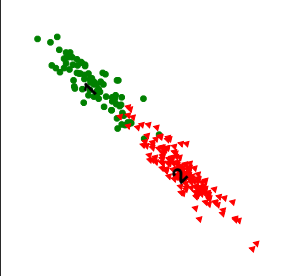
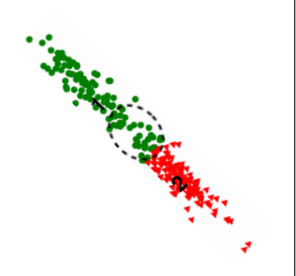
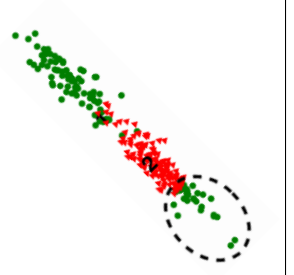
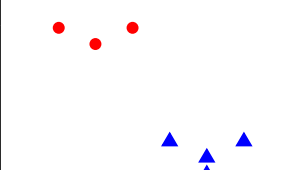
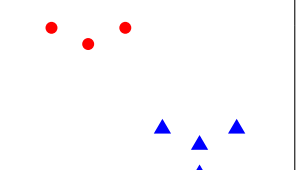
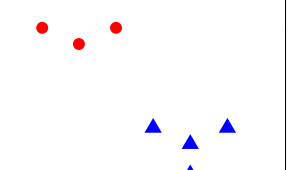
In Examples 1 and 2 in this table, as well as Example 3 in Table 3.3, we modify only the cluster assignments; the points remain stationary. Since the pairwise relationships among the points change in the same way in each modification, only ADCO and CDISTANCE detect that the modifications are not identical with respect to the reference.

### LIFTEDM

Following publication of CDISTANCE in 2010, another spatially aware clustering comparison measure was proposed by Raman et al. (2011). This measure, called LIFTEDM is similar to CDISTANCE in its structure but uses the lift kernel (detailed



*Table (a)*

Ex.	Reference Clustering ( $\mathcal{R}$ )	Change 1	Change 2
3			
4			

*Table (b)*

Technique Name	Example 3			Example 4		
	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?
Hubert	0.25	0.25	×	N/A	N/A	×
1 – Rand	0.00	0.00	×	N/A	N/A	×
VI	5.89	5.89	×	N/A	N/A	×
Mallows	10.0	10.0	×	N/A	N/A	×
ADCO	0.07	0.09	✓	0.00	0.07	×
<b>CDISTANCE</b>	<b>0.41</b>	<b>0.56</b>	✓	<b>0.08</b>	<b>0.09</b>	✓

Table 3.3: **Distances to Modified Clusterings.** (*cont'd from Table 3.2.*)

In Example 4, the reference clustering  $\mathcal{R}$  is modified by moving three points of the bottom right cluster by small amounts. In Modification 1, the points do not move across a bin boundary, whereas in Modification 2, they do. As a result, ADCO detects no change between Modification 1 and  $\mathcal{R}$  but detects a large change between Modification 2 and  $\mathcal{R}$ , even though the two modifications differ by only a small amount. CDISTANCE correctly reports a similar change between Modification 1 and  $\mathcal{R}$ , and Modification 2 and  $\mathcal{R}$ . Other clustering comparison techniques are not applicable to this example because the data sets in the modifications are different from the one in  $\mathcal{R}$ . This table, the preceding table, and all included figures are taken from Coen, Ansari, and Fillmore (2010).

in Section 2.6) in place of one of the components in CDISTANCE. LIFTEMD uses the lift kernel to assign distances between the clusters of two clusterings, treating them as point sets, and then computes the optimal transportation distance  $d_{KW}$

between them (Section 2.4.1) between the clusterings, treating each clustering as a set of points (corresponding to their clusters) in a new vector space.

Tables 3.2 and 3.3 show comparisons between the measures above and CDISTANCE on sample pairs of clusterings. Each row in the table highlights a potential failure mode of non-spatially sensitive clustering comparison methods.

### 3.3 CDISTANCE : A Spatially Aware Distance Between Clusterings

Our goal is to construct a measure of comparison between clusterings of points in a metric space that captures both spatial and partitional information about the clusterings. This symmetric, non-negative measure will be presented as a *dissimilarity* or a distance – not a similarity<sup>3</sup> – with range  $[0, 1]$ . As two clusterings become more similar, the distance between them approaches zero. This is in contrast with most indices for comparing clusterings, such as Rand (1971), where a higher value in the range  $[0, 1]$  indicates greater similarity.

Our spatially aware distance measure CDISTANCE is thus built to answer the question: *what is the disparity in the “overlap” of two clusterings in a given space?* CDISTANCE does not restrict its evaluation to the assignments of points to partitions alone, but crucially, also takes into account the locations of the points in each cluster, the shapes of the clusters, and the spatial relations among the clusters. We allow for the possibility that the two clusterings have different numbers of clusters, and also that they are over potentially distinct data sets. Below we present CDISTANCE together with an algorithm for computing its value on a pair of clusterings.

---

<sup>3</sup>As noted in Section 2.1.3 however, it can be easily converted to a similarity by subtracting from 1 since CDISTANCE is bounded between 0 and 1.

### 3.3.1 Defining CDISTANCE

CDISTANCE makes use of the quantities  $d_{KW}$  (Kantorovich-Wasserstein distance, also called optimal transportation distance) and  $d_{SIM}$  defined in Section 2.4.1. Conceptually, our approach is as follows. Let  $\mathcal{A}$  and  $\mathcal{B}$  be two clusterings of data sets  $D_A$  and  $D_B$ , consisting of  $n$  and  $m$  clusters respectively.  $D_A$  and  $D_B$  are subsets of a metric space  $\Omega$ . Recall that there is no requirement that  $D_A = D_B$ , namely, the set of points being clustered need not be equal; in fact, it may well be the case  $D_A \cap D_B = \emptyset$ .

1. We first construct a *new* metric space  $\mathcal{S}$  – distinct from the metric space  $\Omega$  in which the original data lie – which contains one distinct element for each cluster in  $\mathcal{A}$  and  $\mathcal{B}$ .
2. We define the distance between any two elements of this new space  $\mathcal{S}$  to be the optimal transportation distance  $d_{KW}$  between the corresponding clusters in  $\Omega$ . We note here that since  $d_{KW}$  is a proper metric (Rubner et al., 2000) for distributions of equal mass,  $\mathcal{S}$  is a metric space defined over the collection of all point sets (i.e. over the power set of  $\Omega$ ).
3. The *clusterings*  $\mathcal{A}$  and  $\mathcal{B}$  can now be used to construct corresponding weighted point sets  $\mathcal{A}'$  and  $\mathcal{B}'$  in  $\mathcal{S}$  (with the weights being determined by the relative cardinalities of the original clusters).
4. The degree of similarity between  $\mathcal{A}$  and  $\mathcal{B}$  is then defined as the degree of spatial overlap between their corresponding weighted point sets  $\mathcal{A}'$  and  $\mathcal{B}'$  in  $\mathcal{S}$ , as measured by  $d_{SIM}$ :

$$\text{CDISTANCE}(\mathcal{A}, \mathcal{B}) = d_{SIM}(\mathcal{A}', \mathcal{B}'; d_{KW}),$$

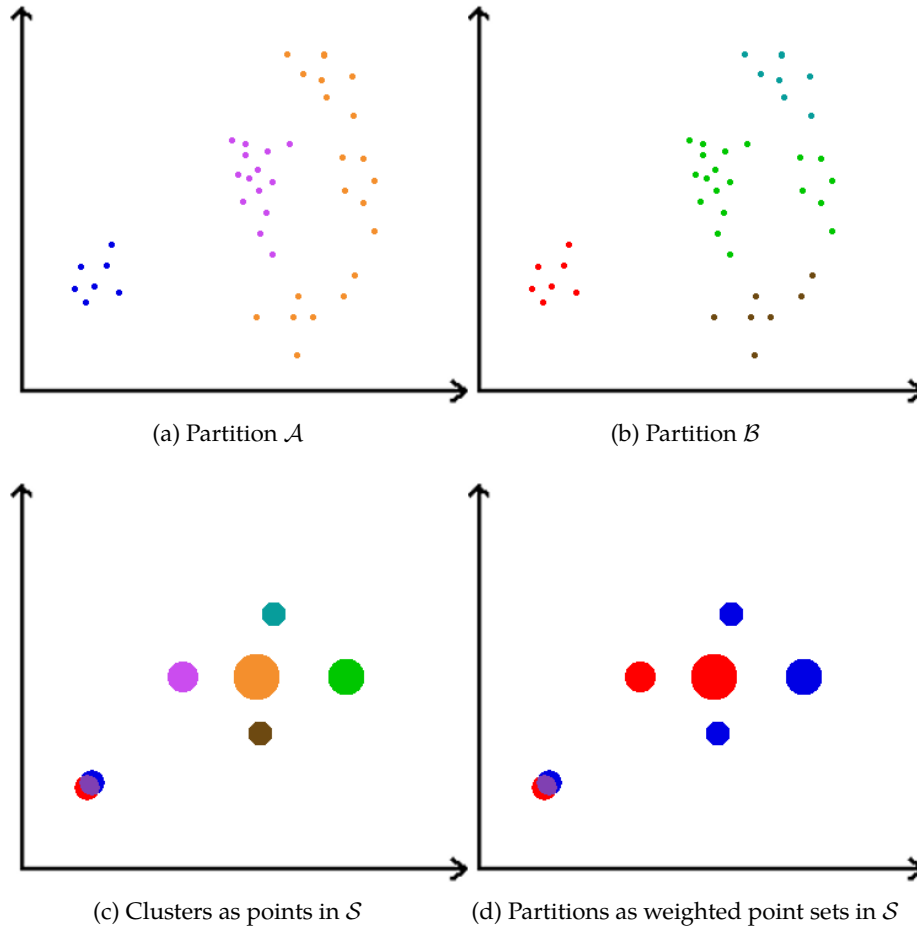


Figure 3.3: **Computing CDISTANCE**. This figure shows a visualization of how CDISTANCE is computed. Panels (a) and (b) show two partitions  $\mathcal{A}$  and  $\mathcal{B}$  of the same data set in  $\mathbb{R}^2$ . Panel (c) shows an approximation of how the clusters in  $\mathcal{A}$  and  $\mathcal{B}$  may be represented as points in a two-dimensional approximation of the new metric space  $\mathcal{S}$ . The metric in  $\mathcal{S}$  is the value of  $d_{KW}$  between their corresponding clusters in  $\mathbb{R}^2$ . The size and color of each point in panel (c) are adjusted proportionally for visual purposes to the size of the original cluster and its color in the original partition respectively. Panel (d) shows how the two clusterings may each be viewed as a weighted point set in the new metric space. The point set corresponding to partition  $\mathcal{A}$  is shown in red, and  $\mathcal{B}$  in blue. CDISTANCE  $(\mathcal{A}, \mathcal{B})$  is the value of  $d_{SIM}$  between these two weighted point sets.

where the weight  $\pi$  associated with each cluster  $A \in \mathcal{A}$  is equal to  $|A|/|D_A|$  and similarly the weight  $\rho$  of each cluster  $B \in \mathcal{B}$  is  $|B|/|D_B|$ , proportional to the number of points in the clusters.

Figure 3.3 shows a visual demonstration of these steps on an example data set. The procedure outlined above transforms a clustering into a weighted point set. Note the interesting fact that we do not know the coordinates of the points in  $\mathcal{S}$  representing each cluster. However, we do not need this information because the next step of computing  $d_{\text{sim}}$  only requires knowing *pairwise distances* between points in  $\mathcal{S}$  – which we have by way of  $d_{\text{KW}}$  – not their absolute locations. *Thus, we have reduced the problem of comparing clusterings to the (solved) problem of comparing similarity between two point sets.* This neatly sidesteps the computationally intractable problem of examining the exponential space of all possible permutations of matches between clusters in  $\mathcal{A}$  to clusters in  $\mathcal{B}$ ; these are explicitly enumerated in Bae et al. (2006).

An efficient algorithm to compute clustering distance is easily derived from the definition. Let  $\mathcal{A}$  and  $\mathcal{B}$  be as above. We use two steps:

#### Step 1.

This step computes the  $d_{\text{KW}}$  metric mentioned above. Each cluster  $A \in \mathcal{A}$  and  $B \in \mathcal{B}$  is a uniformly weighted point set; for each pair of clusters  $(A, B) \in \mathcal{A} \times \mathcal{B}$ , we compute the optimal transportation distance  $d_{\text{KW}}(A, B; d_\Omega)$  based on the distances according to  $d_\Omega$  between points in  $A$  and  $B$ .

The result of this step is a matrix  $(d_{\text{KW}}(A, B))$  which contains the optimal transportation distance evaluated on all points in  $\mathcal{S}$  of interest to us. We will use this matrix in Step 2.

#### Step 2.

In this step, we compute the CDISTANCE between  $\mathcal{A}$  and  $\mathcal{B}$ . We will first construct

weighted point sets in  $\mathcal{S}$  corresponding to  $\mathcal{A}$  and  $\mathcal{B}$ . For each cluster  $A_i \in \mathcal{A}$  let  $A'_i$  be its corresponding point in  $\mathcal{S}$ . Note that we are assured that  $A'_i$  exists – due to the metricity of  $d_{KW}$  – but we do not know its exact representation. We construct a weighted point set  $\mathcal{A}' = \{(A'_1, \pi_1), \dots, (A'_n, \pi_n)\}$  where  $\pi_i = |A_i|/|D_A|$  (recall that  $D_A$  is the underlying data set that was clustered to produce  $\mathcal{A}$ ). The weight on each cluster is proportional to the number of points in the cluster. Similarly, let  $B'_j$  be the corresponding point in  $\mathcal{S}$  for each cluster  $B_j \in \mathcal{B}$ ,  $\rho_j = |B_j|/|D_B|$  its associated weight, and the weighted point set  $\mathcal{B}' = \{(B'_1, \rho_1), \dots, (B'_m, \rho_m)\}$  corresponding to  $\mathcal{B}$ .

$\text{CDISTANCE}(\mathcal{A}, \mathcal{B})$  can now be computed as  $d_{\text{SIM}}(\mathcal{A}', \mathcal{B}'; d_{KW})$ , using the optimal transportation distances between clusters computed in Step 1.

### 3.3.2 Discussion of CDISTANCE Design Choices

Note that we use optimal transportation distance to measure the distance between individual clusters (Step 1), while we use similarity distance to measure the distance between the clusterings as a whole (Step 2). The reason for this difference is as follows. In Step 1, we are interested in the *absolute* distances between clusters: we want to know how much work is needed to move all the points in one cluster onto the other cluster. We are not interested here in the relative *improvement* of optimal transportation distance over naive transportation; rather, we would like to know how much work is needed to transform one cluster into the other and so we use  $d_{KW}$  as a measure of how “far” one cluster is from another.

In contrast, in Step 2 we want to know the *degree* to which the clusters in one clustering spatially overlap with those in another clustering using the distances derived in Step 1. Similarity distance is an appropriate measure of distance between two clusterings because it determines how well one clustering “fits” onto another, while respecting the weights of their constituent clusters and

distances between them. This is an exact and well-motivated optimization that distills both spatial and categorical information into a single measure.

Our choice to use uniform weights in Step 1 but proportional weights in Step 2 has a similar motivation. In a (hard) clustering, each point in a given cluster contributes as much to that cluster as any other point in the cluster contributes, so we weight points uniformly when comparing individual clusters. In contrast, Step 2 proportionally distributes the influence of each cluster in the overall computation of  $\text{CDISTANCE}$  according to its relative weight, as determined by the number of data points it contains. For example, if a single cluster  $A \in \mathcal{A}$  contains almost all the points of  $X$ , then the degree of spatial overlap of the clustering  $\mathcal{A}$  as a whole with any other clustering  $\mathcal{B}$  is dominated by the spatial overlap of the single cluster  $A$  with clusters of  $\mathcal{B}$ . By weighting the clusters in each clustering in proportion to their cardinality, we obtain this desired behavior.

### 3.3.3 Properties and Behavior

Since  $\text{CDISTANCE}$  is defined as the value of  $d_{\text{SIM}}$  between two quantities, many of its properties are identical to that of  $d_{\text{SIM}}$ . The difference of course, is that  $\text{CDISTANCE}$  is defined over collections of weighted point sets and  $d_{\text{SIM}}$  over weighted point sets. Similar to  $d_{\text{SIM}}$ ,  $\text{CDISTANCE}$  is a dissimilarity measure that lies in  $[0, 1]$ . When two clusterings are identical, it takes the value 0; this happens when the clusterings overlap perfectly, leading to two identical weighted point sets in the new metric space  $\mathcal{S}$ , which in turn causes  $d_{\text{SIM}}$  to be 0.

Figure 3.4 illustrates some values of  $\text{CDISTANCE}$  for sample clusterings. In each subfigure, we are comparing two clusterings, one of which has been translated for visualization purposes. For example, in Figure 3.4 (a), the two clusterings spatially overlap perfectly so their clustering distance is zero. Matching clusters are connected by lines to illustrate their correspondence. (These

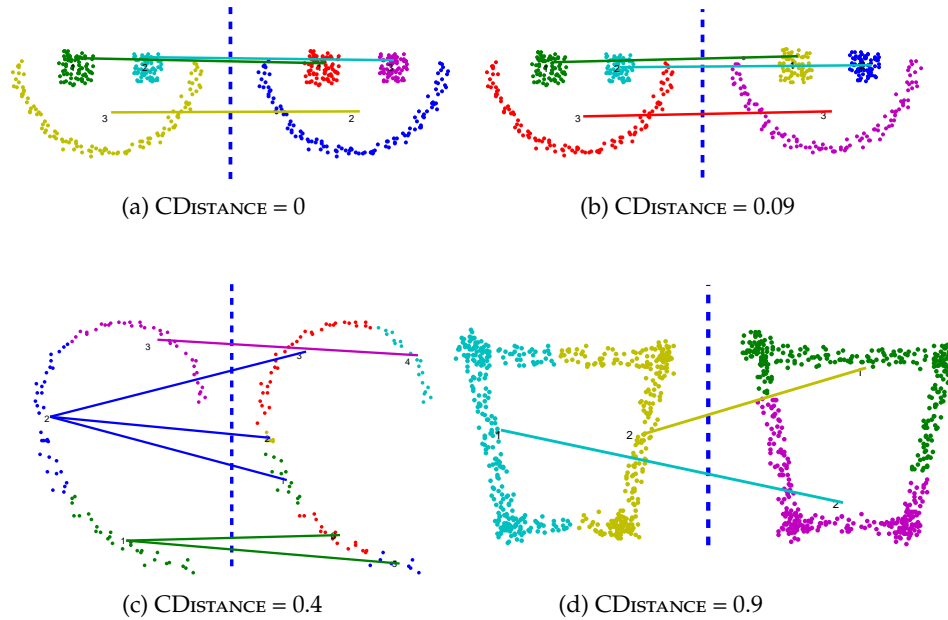


Figure 3.4: **Clustering Stability.** Each panel above shows a clustering of two data sets, either identical to one another or slightly perturbed, and the  $CDISTANCE$  value between them. While the clusterings are separated by a blue dotted line, the data sets overlap significantly and occupy similar regions of space; the separation is for visualization purposes only. (a) Identical clusterings and identical data sets;  $CDISTANCE$  is 0. (b) Similar clusterings, but over slightly perturbed data sets;  $CDISTANCE$  is 0.09. (c) Two different algorithms were used to cluster the same data set.  $CDISTANCE$  is 0.40, indicating a moderate mismatch of clusterings. (d) Two very different clusterings generated by spectral clustering over almost-identical data sets.  $CDISTANCE$  is 0.90, suggesting instability in the clustering algorithm's output paired with this data set.

lines are drawn solely for visualization purposes.) The most interesting panel is Figure 3.4 (d), which demonstrates that symmetries in a shape can produce wildly disparate clusterings; this is referred to as “instability.” Repeated applications of spectral clustering to the data shown in Figure 3.4 (d) produce very different clusterings, both visually and as measured by  $CDISTANCE$ . Multiplying clustering a data set and calculating  $CDISTANCE$  between the outputs allows us to gauge whether an algorithm/data set combination are mutually compatible.



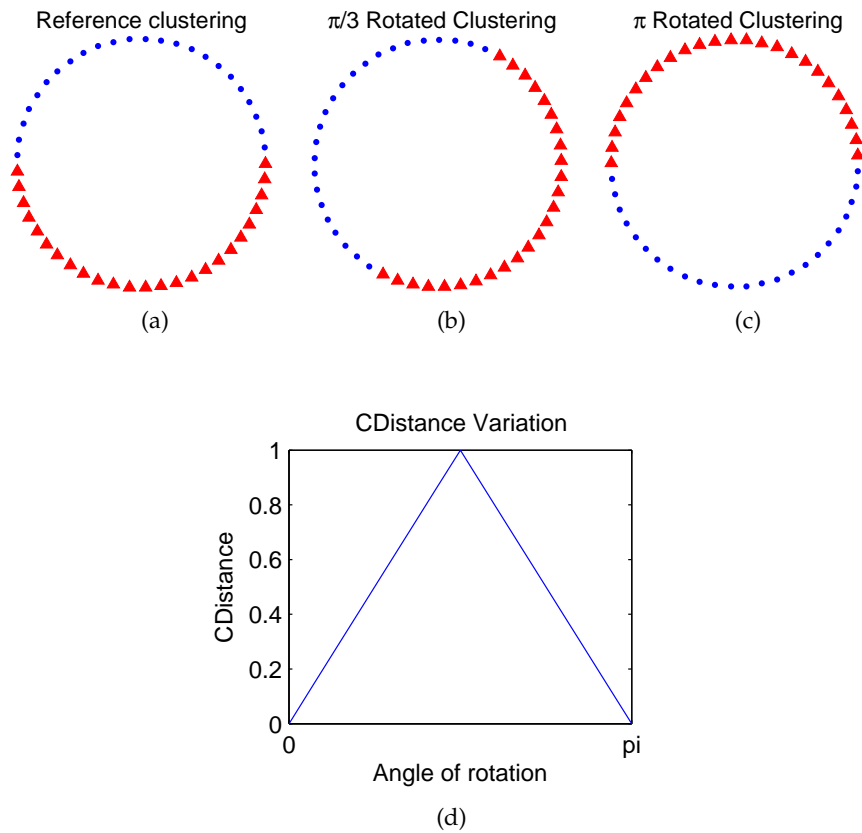


Figure 3.5: **CDISTANCE Stability.** Examining stability of  $CDISTANCE$  with respect to aggregations of small changes in partitioning. (a) Reference clustering. This data set is a subset of the unit circle with geodesic distance function. (b) An intermediate clustering.  $CDISTANCE$  between this clustering and the reference clustering is 0.60. (c) The completely rotated clustering;  $CDISTANCE$  between this clustering and the reference clustering is 0. (d) The graph of variation of  $CDISTANCE$  with angle of rotation is linear.

Figure 3.5 illustrates the smoothness of  $CDISTANCE$  as clusterings change in small increments. Figure 3.5 (d) reflects the distance between the clustering shown in Figure 3.5 (a) and intermediate clusterings as it is rotated incrementally to the clustering in Figure 3.5 (c). We note these two clusterings are indistinguishable because they cluster the data equivalently. Thus, rotations of both 0 and  $\pi$  radians have  $CDISTANCE$  zero.

For the final example, consider the clustering in Figure 3.6 (a), which contains the same data set as Example 4 from Table 3.3 (a). Suppose we incrementally increase the  $y$ -coordinates of the cluster consisting of blue triangles. At each step, we compute both ADCO (Bae et al., 2006) and CDISTANCE between the modified data set and the reference clustering in Figure 3.6 (a). The resulting values are plotted in Figure 3.6 (b). We see that ADCO suffers from swings and discontinuities due to the abrupt transition of data moving between discrete bins. As a result of this behavior, ADCO values are difficult to interpret intuitively.

### 3.4 Stability

A spatially aware comparison between clusterings finds application in an important technique for selecting a given clustering solution. As noted earlier, the unsupervised nature of clustering makes the task of directly evaluating

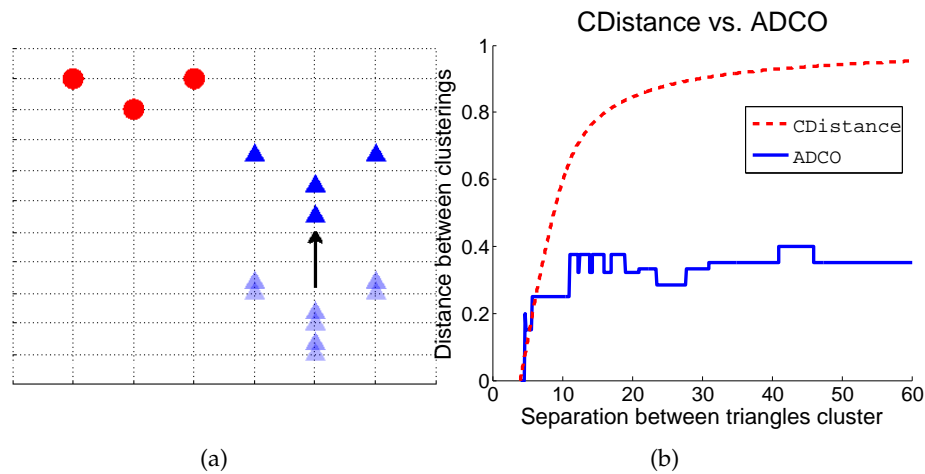


Figure 3.6: **ADCO Variation.** This figure shows the smooth variation of CDISTANCE as compared to ADCO with small changes in the data set. Panel (a) shows a reference clustering; Panel (b) shows a plot of CDISTANCE and ADCO values as a function of how far the “blue triangle” cluster is displaced upwards. This figure is adapted from Coen, Ansari, and Fillmore (2010).

the quality of its partitioned outputs very difficult. As advanced by a number of researchers in the past few years (Dudoit and Fridlyand, 2003; Lange et al., 2004; Ben-Hur et al., 2002; Shamir and Tishby, 2010), one desirable aspect of a good clustering solution is “stability,” i.e. the idea that a clustering should be robust to various perturbations in its input. The output partition should not be heavily dependent on specific minute features of the data set; if some aspects of a data set perturbed slightly, the overall clustering solution should not change significantly. It is here that a spatially aware comparison measure such as  $CDISTANCE$  finds immediate utility: estimating the changes in a partition caused to due perturbations in the data.

Ben-Hur et al. (2002) proposed that the stability of a clustering solution could be determined by repeatedly clustering subsamples of its data set. Finding consistently high similarity across clusterings indicates consistency in locating similar substructures in the data set. This can increase confidence in the applicability of an algorithm to a particular distribution of data. In other words, by comparing the resultant clusterings, one can obtain a goodness-of-fit between a data set and a clustering algorithm. The clustering comparisons in Ben-Hur et al. (2002) are all done via partitional methods. Shamir and Tishby (2010) extends this idea further and provides theoretical underpinnings and bounds for its applicability.

We can instead use  $CDISTANCE$  to perform this comparison. This is depicted in Figure 3.7, where we repeatedly cluster subsamples of a data set with both Self-Tuning Spectral Clustering (Zelnik-Manor and Perona, 2004) and Affinity Propagation (Dueck and Frey, 2007). Although these algorithms rely on mathematically distinct properties, we see their resultant clusterings on subsampled data agree to a surprising extent according to  $CDISTANCE$ .

Because  $CDISTANCE$  is able to compare clusterings of different cardinality, we can use it with algorithms that self-determine how many clusters to gener-

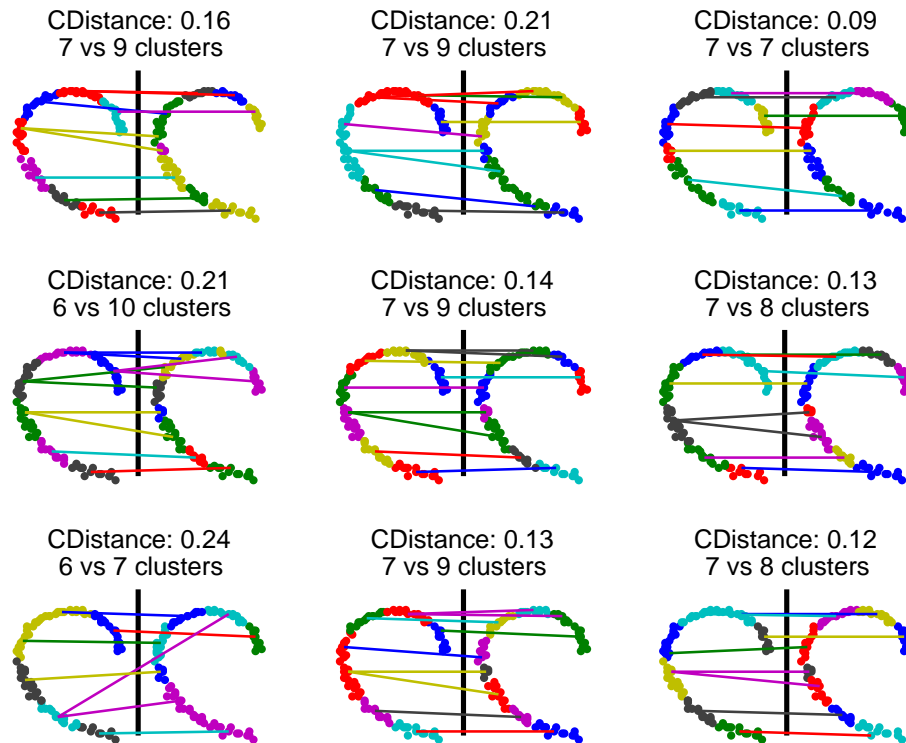


Figure 3.7: **Stability.** Comparing outputs of different clustering algorithms (self-tuning spectral clustering and affinity propagation) over randomly sampled subsets of a data set. The low values of  $CDISTANCE$  indicate relative similarity and stability between the clusterings. The numbers of clusters involved in each comparison are displayed below the  $CDISTANCE$  value. The intuition behind  $CDISTANCE$  is reflected in the lines connecting clusters (across clusterings) that are spatially similar to one another.

ate (such as self-tuning spectral clustering above). Thus, we can use a wider assortment of clustering algorithms in ensemble methods and for stability testing.

### 3.5 Conclusion

In this chapter we presented a new algorithm for comparing clusterings that computes a value we call  $CDISTANCE$ . This approach employed by  $CDISTANCE$  incorporates both spatial and categorical information into a single distance

function and varies smoothly with changes to the underlying data. It captures an intuitive notion of clustering overlap that makes it easy to apply and interpret. It is unique in enabling comparisons between clusterings that differ in their data sets, number of points, and number of clusters. This significantly broadens the range of applications for this measure in comparison to other approaches to comparing clusterings. The `CDISTANCE` algorithm is extensible to comparing soft clusterings (such as those generated by Expectation-Maximization techniques and other probabilistic methods) by replacing the uniform distribution assumed across points of a cluster in Step 1 with distributions describing the fractional clustering memberships.

#### 4 ENSEMBLE CLUSTERING

---

Recent advances in unsupervised learning seek to take advantage of “ensembles” of outputs of clustering algorithms (Nguyen and Caruana, 2007; Fern and Lin, 2008; Berikov, 2014; Franek and Jiang, 2014; Vega-Pons and Ruiz-Shulcloper, 2011). The contributions of many different clusterings in the ensemble are then distilled to yield a single consensus clustering. These ensemble methods provide improved results and can provide several important benefits over single algorithm clustering. As discussed in Topchy et al. (2004), advantages of applying ensemble techniques to clustering problems include improved robustness across data sets and domains, and increased stability of clustering solutions. Perhaps most importantly however, ensemble techniques provide the ability to arrive at clusterings that are not individually obtainable by any single practical clustering algorithm used while generating an “ensemble” (a collection of candidate clusterings). This phenomenon is possible because there are an exponential number of ways to partition a given set of points, and any clustering algorithm only searches a fraction of the space of possible partitions. Ensemble techniques provide ways to arrive at a clustering that lies outside the search spaces of individual algorithms, drawing solutions from a larger search space.

Unsurprisingly, ensemble methods rely on a notion of a distance or similarity – whether stated explicitly or implicitly – between clusterings. In order to create an integrated ensemble clustering from several individual clusterings, information about which clusterings are more similar or less similar or *which parts* of which clusterings are compatible with each other and can be combined is frequently necessary.

Despite the inherent spatial nature of clustering, previous approaches to ensemble clustering – like approaches to compare clusterings – have mainly



Figure 4.1: **Ensemble Clustering.** The typical flowchart of an ensemble clustering framework. Multiple clusterings of a single data set  $X$  are generated by varying parameters, perturbing the data, and subsampling. The collection of these clusterings, called an “ensemble” is then input to a consensus step that attempts to combine information from all these clusterings and generate a final consensus clustering from them.

used “partitional” or set-theoretic techniques to measure similarity between clusterings. They only account for the cluster identities of points, but not their spatial information. As we showed above, it is in many contexts a major weakness to ignore spatial information in clustering comparisons. Not taking into account the location of data points for which labels change across clusterings can lead to a misleading quantification of the difference between the clusterings.

We present below a state-of-the-art algorithm that compares and combines information from multiple clusterings by incorporating spatial information. Our algorithm **significantly outperforms** other existing ensemble clustering algorithms (as measured by agreement with ground truth on labeled data sets) that either do not take spatial information into account or that use spatial information less effectively, as shown in the experiments below. Specifically, we use  $\text{CDISTANCE}$  (introduced in previous sections) to compare the similarity of any two clusterings.  $\text{CDISTANCE}$  provides a measure of the extent to which the constituent clusters of each clustering overlap with the other, and by extension, how well the two clusterings agree with each other. In the current context, it is used to measure the diversity in a set of clusterings. Then, given a sufficiently diverse set of clusterings, we construct a consensus clustering using a variant of the optimal transportation distance  $d_{KW}$  (Section 2.4.1).

## 4.1 Ensemble Clustering Overview

An ensemble clustering is a consensus clustering of a data set  $X$  based on a collection of  $R$  clusterings. The goal is to combine aspects of each of these clusterings in a meaningful way so as to obtain a more robust and “superior” clustering that combines the contributions of each clustering. It is frequently the case that one set of algorithms can find a certain kind of structure or patterns within a data set and another set of algorithms can find a different structure. For example, k-means produces clusters with a search bias for locality, where locality is defined by radial distance to a center, whereas a spectral clustering algorithm prefers to segregate clusters of points that lie near each other in a low-dimensional manifold. An ensemble clustering framework tries to constructively combine inputs from different clusterings to generate a consensus clustering.

There are typically two steps involved in any ensemble framework; a diagram is shown in Figure 4.1. There is also sometimes an optional third step in between:

1. Generation of  $R$  clusterings of  $X$
2. Evaluation of diversity and other metrics in the generated clusterings
3. Generation of a consensus clustering from the clusterings in Step 1

Clusterings may be generated by a number of ways: by using different algorithms, by supplying different parameters to algorithms, by clustering subsets of the data, by clustering the data with different feature sets (Fern and Brodley, 2003), and by generating clusterings with a randomized Metropolis-Hastings process (Phillips et al., 2011).

The consensus clustering can also be obtained through a number of ways such as linkage clustering on pairwise co-association matrices (Strehl and



Ghosh, 2003), graph-based methods (Fern and Brodley, 2004) and maximum likelihood methods (Topchy et al., 2004). We provide more detail on each of these below.

## 4.2 Related Work

There are two primary ways in which ensemble clustering algorithms differ from one another. The first is the choice of similarity/distance measures used between clusters or clusterings, and the second is the consensus algorithm. There are a variety of approaches to distilling all the information contained in an ensemble of clusterings and coming up with a consensus clustering. A good survey of the variety of solutions in both problems can be found in Ghaemi et al. (2009). We focus in this section on prior work in the second category; work related to the first is detailed in Section 3.2.

### 4.2.1 Hypergraph-based methods

The set of  $R$  clusterings produced in Step 1 of the ensemble framework can be represented by a hypergraph,  $G$ . A hypergraph is a variant of the classic graph data structure where edges can connect two or more nodes and are called hyperedges. In this hypergraph, nodes correspond to individual points in data set  $X$ , and hyperedges are used to indicate that a set of points belong to the same cluster in one of the  $R$  clusterings. In other words, there is one hyperedge in  $G$  per cluster, per clustering. A consensus clustering can then be achieved by finding the minimum  $k$ -cut of  $G$  (explained in more detail below). Although the hypergraph min-cut problem is NP-hard, several techniques for efficiently approximating a solution have been proposed.

Strehl and Ghosh (2003) introduced a framework consisting of three different methods for performing the consensus step over a hypergraph representation

of the clusterings. In this framework, the consensus clustering is chosen to be one that maximizes its average mutual information with *all* clusterings in the ensemble.

#### 4.2.2 Graph-based methods

Several ensemble clustering frameworks employ graph-based algorithms for the combination step (Fred and Jain, 2002; Fern and Brodley, 2003). One popular idea is to construct a cluster or point graph taking into account the similarity between clusterings and perform a graph cut or partition that optimizes some objective function. Fern and Brodley (2004) describe three main graph-based methods: instance-based, cluster-based and hypergraph partitioning, also discussed in Strehl and Ghosh (2003). The cluster-based formulation in particular requires a measure of similarity between clusters, originally the Jaccard Index. The unsuitability of this index and other related indices are discussed in Section 3.2.

The approach ultimately adopted in Fern and Brodley (2004) uses graph nodes to represent both individual points (or instances) and clusters. A bipartite graph is then created with edges connecting each instance to each cluster it is a member of. The resulting Hybrid Bipartite Graph Formulation can then be partitioned with standard graph partitioning techniques. In forming a consensus clustering, this method simultaneously considers both the similarity of instances and the similarity of clusters.

#### 4.2.3 Others

Several other methods for performing the consensus step have been proposed. Dudoit and Fridlyand (2003) and others propose voting-based approaches where each point votes on which cluster it belongs to. In order for this to work, the correspondence problem must first be solved, where each cluster in each

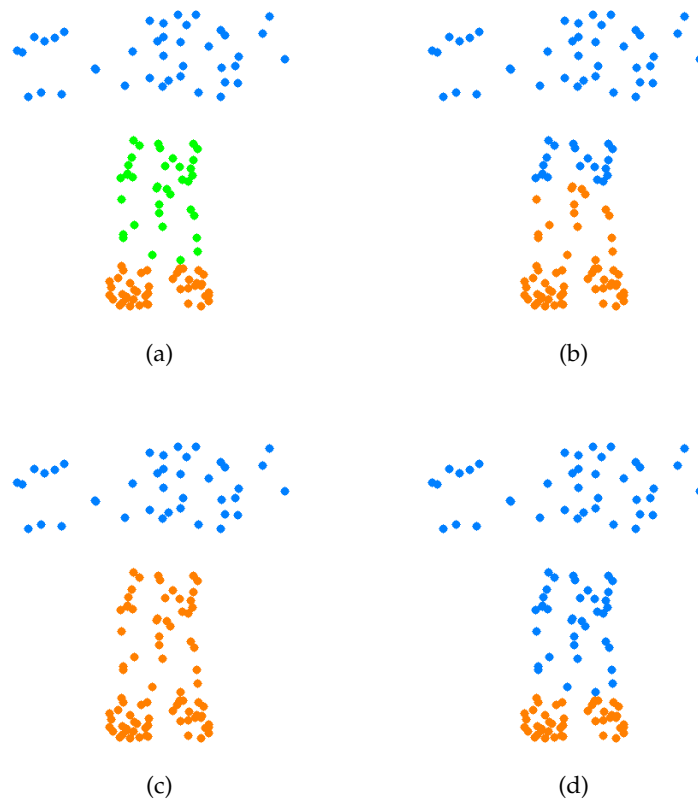


Figure 4.2: **Spatially-aware ensemble clustering.** Panels (a) and (b) show two different clusterings of the same data set using two different algorithms. Panel (d) shows the result of a spatially unaware cluster comparison measure (the Jaccard index used in Fern and Brodley (2004)) used to derive the consensus clustering, whereas panel (c) shows the result when a spatially aware cluster comparison measure ( $CDISTANCE$ ) is used.

clustering is assigned a corresponding cluster in every other clustering. Once this correspondence is determined, points within clusters can vote according to which consensus cluster they appear most frequently in. Other voting approaches construct a “co-association matrix” (Fred, 2001; Fred and Jain, 2002), incorporating information about pairs of points co-occurring in the same cluster. Agglomerative clustering algorithms are then applied to this matrix to generate a final consensus clustering.

Similar to the above, Raman et al. (2011) applies the approach of “clustering the clusterings” based on an informative distance measure between clusterings derived from the lift kernel (introduced in Section 2.6).

A number of other methods (Topchy et al., 2005; Vinh and Epps, 2009) are based on optimizing an objective function based on the mutual information between the labels of individual clusterings and the consensus clustering labels.

Figure 4.2 shows an example on a toy data set of the different consensus clusterings that can result from spatially aware and unaware methods.

### 4.3 Algorithm

We present below our ensemble clustering algorithm, followed by results of experiments using it on real world data sets and comparing it with other ensemble clustering algorithms. Our algorithm, which we call Spatial Ensemble Clustering (SEC), proceeds in 3 stages:

1. Ensemble Generation
2. Diversity Estimation
3. Consensus Finding

Before we proceed with a description of the algorithm we define first a variant of  $d_{KW}$  that we will use in the consensus finding step.

#### 4.3.1 Kantorovich-Wasserstein Distance (unnormalized)

In this variant of Kantorovich-Wasserstein Distance (first defined in Section 2.4.1) we remove the restriction that the set of associated weights for each point set sum to 1. We instead set each point in both point sets to have equal weight (set to 1). This has the effect of measuring overlap between two sets of points *while not penalizing for non-overlap*.

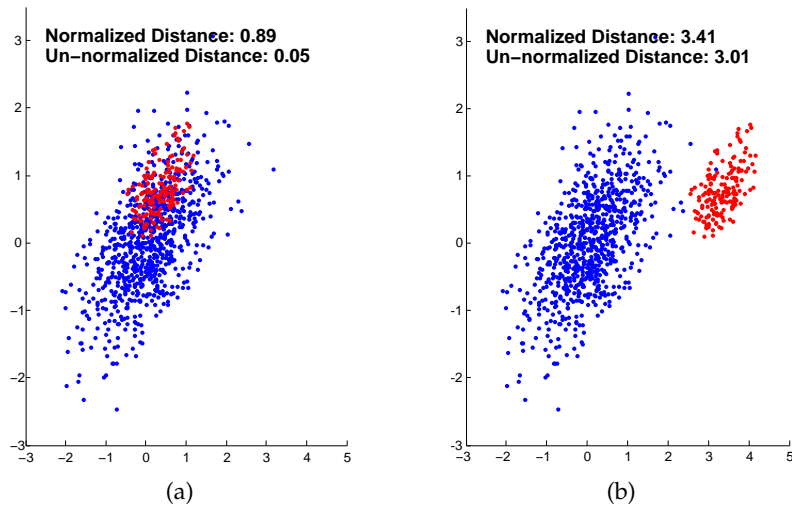


Figure 4.3: **Un-normalized**  $d_{KW}$ . This figure contrasts the difference between normalized and un-normalized Kantorovich-Wasserstein distance. When the clusters do not overlap, as in the figure on the right, normalization makes little difference. However, as one of the clusters approaches a subset of the other, un-normalized distance approaches 0, as can be seen in the figure on the left.

An example contrasting these two transportation distances on clusters of points is shown in Figure 4.3. We will use this variant of Kantorovich-Wasserstein distance, called  $d_{UKW}$ , in the consensus step of our ensemble clustering algorithm.

### 4.3.2 Stage 1: Ensemble Generation

We use the following methods to generate candidate clusterings for the ensemble:

- Running a variety of different clusterings algorithms, e.g. k-means (MacQueen et al., 1967), spectral clustering (Ng et al., 2002), affinity propagation (Dueck and Frey, 2007), k-medians (Jain and Dubes, 1988), and hierarchical linkage algorithms (Kaufman and Rousseeuw, 2009).

- Running the above algorithms with a variety of different parameters. In the examples above, among other things we can vary the number of clusters ( $k$ ), neighborhood size, and initializations, depending on the algorithm.
- Running the above algorithms and parameter combinations on random subsets of the original data set. This is a method of introducing diversity in the set of clusterings generated because the removal of some data points may cause the rest to cluster differently (Ben-Hur et al., 2002).
- Running the above algorithms and parameter combinations on the data set with small amounts of added Gaussian white noise. This achieves a similar goal as the above point and may lead to different clusterings of the data set.
- Running the above algorithms and parameter combinations on random subsets of features of the original data set. This leads to diversity in the clusterings by considering different views of the same data set.

The methods above allow us to include the output of several algorithms and parameter combinations without committing to any one such combination. Our goal will be to combine the information coming from these clusterings in Step 3.

There are many other methods of generating diversity in clusterings which may be used as well in this step. Examples include using a Metropolis-Hastings process to generate high-quality clusterings from existing clusterings (Phillips et al., 2011), or generating a high-quality clustering optimizing entropy and maximal difference from a given clustering using information theory (Dang and Bailey, 2010). These measures of cluster coherence or validity however introduce an additional inductive bias into the ensemble of clusterings that

can be avoided. For example, a cluster validity measure may prefer spherical clusters or clusters that are convex in their shape and assign them higher scores, whereas the data set may not be conducive to such scores of cluster quality.

### 4.3.3 Stage 2: Diversity Estimation

Some amount of diversity in clusters constituting the ensemble is crucial to the quality of the final generated clustering (Hadjitodorov et al., 2006). In Section 3.4 we noted that  $CDISTANCE$  is able to detect similarity between clusterings of subsets of a data set. A corollary of this is that  $CDISTANCE$  is also able to detect diversity in members of an ensemble. Previous methods have used the Adjusted Rand Index (Hubert and Arabie, 1985) or other such metrics to quantify diversity. As we demonstrated in Section 3.2, these measures are surprisingly insensitive to large differences in clusterings.  $CDISTANCE$  provides a smoother, spatially-sensitive measure of the dissimilarity of two clusterings leading to a more effective characterization of diversity within an ensemble.

Measurement of diversity guides the generation step by indicating whether the members of an ensemble are sufficiently diverse or differ from each other only minutely. In case there is insufficient diversity, a return to Step 1 is warranted in order to add more algorithm-parameter combinations to generate more clusterings.

In the experiments below we defined diversity as the average  $CDISTANCE$  among members of an ensemble. We set a threshold of 0.1 for the diversity in order to proceed with the final consensus step.

### 4.3.4 Stage 3: Consensus Finding

Our algorithm employs a graph partitioning method in the final step to generate the consensus clustering, based on the “cluster-based graph formulation” method of Fern and Brodley (2004), explained in more detail below. While the

authors chose not to proceed with this approach due to bad performance, we have found that it works remarkably well when the cluster similarity measure is sensitive to the spatial arrangement of the clusters.

Our algorithm requires an input value for  $k$ , the desired number of clusters in the final consensus clustering. This value may be preset according to the data set, or it may be determined by the domain, or one may use an algorithm such as Affinity Propagation (Dueck and Frey, 2007) to determine an “optimal” number of clusters according to some criterion.

We first describe the consensus step on hard clusterings, and extend it to soft clusterings next. Let  $\mathbb{S} = \{S_1, S_2, \dots, S_R\}$  be the set of  $R$  clusterings generated in the first step. Recall from Definition 3.1 that a hard clustering itself is a set of clusters representing groups of data points. Let  $\mathbb{C} = \cup_{i=1}^R S_i$ , and  $K$  be the cardinality of  $\mathbb{C}$  i.e. the total number of clusters in all the clusterings.

We generate a graph  $G = (V, W)$  where  $V$  is a set of  $K$  vertices, each vertex representing a cluster in  $\mathbb{C}$  and  $W$  is a set representing the edge weights between each pair of vertices. In our case,  $W$  is a pairwise similarity matrix between the clusters of  $\mathbb{C}$ , with the similarity measure  $w$  being derived from the unnormalized optimal transportation distance  $d_{UKW}$ . Since  $d_{UKW}$  is a distance rather than a similarity measure, we use the additive inverse of  $d_{UKW}$  to represent similarity. An example graph is shown in Figure 4.4.

Note that in this graph, clusters that contain points in similar regions of the original metric space will have high similarity (even if those clusters are from different clusterings), whereas clusters containing points from different regions of the metric space will have low similarity. We partition this cluster graph using the normalized graph cut (Ncut) criterion of Shi and Malik (2000)<sup>1</sup> and set the number of subgraphs to partition the original graph into to be  $k$ , the number of clusters we would like the consensus partition to contain.

---

<sup>1</sup>The implementation we used is from Cour et al. (2004).



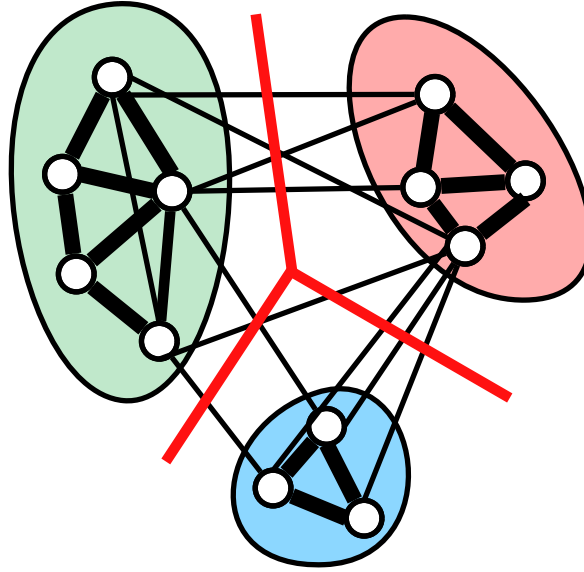


Figure 4.4: **Graph Cut.** This figure shows an example graph that is constructed from the clusters within an ensemble. Each node in the graph represents a cluster from one of the clusterings. The edges between nodes represent the value of the cluster similarity measure between any two clusters (not all edges are shown), and the red lines show a 3-way normalized cut of this graph. Since each point appears exactly once in each clustering, different clusters within a component can contain different instances of the same point. Each point thus votes as to the number of times it appears in each component, and the consensus clustering is constructed by placing each point in the component it occurs the most number of times in.

Ncut is an approximation algorithm that finds a  $k$ -way cut minimizing the sum of normalized graph cuts, i.e. minimizing

$$Ncut_k = \sum_{i=1}^k \frac{cut(A_i, V - A_i)}{assoc(A_i, V)} \quad (4.1)$$

where  $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$  is the sum of edges leaving a subgraph defined by vertices in  $A$ ,  $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$  is the sum of weights

from vertices in  $A$  to all vertices in the graph, and  $A_i$ 's are subsets of the vertex set  $V$ .

With the cluster graph now partitioned into  $k$  disjoint subgraphs, each subgraph leads to one cluster in the consensus clustering. This operation is shown on an example graph in Figure 4.4. Each node in the graph represents a cluster from one of the clusterings in an ensemble. The edges shown represent the value of the cluster similarity measure between any two clusters.

There are  $R$  instances of each point in the original data set scattered among clusters in the  $k$  subgraphs. For each point in the original data set we determine the subgraph its instances occur most frequently in, and assign it to the cluster in the consensus clustering corresponding to that subgraph. In this way, each point is assigned a cluster; the consensus clustering is generated from these assignments.

### 4.3.5 Extension to soft clusterings

If in the first step of the ensemble process we make use of clustering algorithms that output soft clusterings, we may use both subsequent steps with minor modifications.

We treat the output of a soft clustering algorithm as a matrix  $C_{n \times k}$  where  $n$  is the number of data points and  $k$  is the number of clusters. An element  $(i, j)$  of this matrix represents the probability  $p(A_j|x_i)$  of data point  $i$  belonging to cluster  $j$  of the soft clustering.

For the diversity estimation step, optimal distance and CDISTANCE both already require weighted point sets in their inputs. In the hard clustering case we set each point to have equal weight within the cluster, but in the soft clustering case we simply set the weight of all points to be equal to their probability of belonging to that cluster. Each point is thus replicated with different weights in all clusters, but the sum of weights across replications is 1.

In the graph construction part of the consensus step, we use a similar procedure as above to calculate the unnormalized optimal transportation distance  $d_{\text{UKW}}$ . Instead of using uniform weights, we use the probabilities assigned by the soft clustering of the point belonging to a cluster.

Finally, after performing the  $k$ -way graph cut with this weight matrix, we proceed with the voting step as before, but in this case the vote of each point contributes only as much as its weight. For each point  $x$  and each subgraph  $G_k$ , we sum the weights of all instances of  $x$  occurring in  $G_k$  and assign it to the probability of  $x$  belonging to cluster  $k$  in the final consensus clustering. This generates a final soft consensus clustering.

#### 4.3.6 Computational Complexity

The complexity of the generation step depends on the clustering algorithms employed. The complexity of the diversity estimation step is  $O(R^2k^2S)$  where  $R$  is the number of clusterings used in the ensemble and  $S$  is the running time of  $d_{\text{UKW}}$  for a cluster of points (see next paragraph). The final consensus step has two main parts: the pair-wise similarity matrix construction and the  $k$ -way graph cut. The similarity matrix construction will also take  $O(R^2k^2S)$  time since there are  $O(Rk)$  clusters in total from all clusterings. The graph cut step requires the solution of an expensive eigenvalue problem, but the algorithm we employ makes use of a special structure in the problem (Shi and Malik, 2000) to solve it in  $O(Rkt)$  time, where  $t$  is observed to be less than  $\sqrt{Rk}$ .

Finally,  $S$ , which is the time complexity of  $d_{\text{UKW}}$  above is  $O(p^3)$  in the worst case, and  $O(p^{2.6})$  in practice (see Section 2.4.4), where  $p$  is the number of points in the cluster ( $p$  is typically  $O(n/k)$  but can be  $n$  in the worst case). Optimal transportation distance runs quite fast in practice, especially when the density structure of points is taken advantage of (e.g. via hyperclustering described in Section 2.4.5). This technique lowers the input complexity of the data set with

Data Set	Dimensionality	# of instances	# of Classes
Iris	4	150	3
Wine	13	173	3
Ionosphere	34	351	2
Soybean	35	47	4
ISOLET	617	1559	26
MNIST (Test)	784	10,000	10
MNIST (Full)	784	60,000	10

Table 4.1: **Data Sets.** This table displays the characteristics of data sets used to evaluate ensemble clustering algorithms in this chapter. The data sets Iris, Wine, Ionosphere, Soybean, and ISOLET are from the UCI Machine Learning Repository Asuncion and Newman (2007) and form a diverse collection of data set with respect to high and low dimensionalities, small and large numbers of instances, and few and many classes. The MNIST LeCun et al. (1998) data sets are the train and test sets respectively of a large and popular digit recognition image database.

minimal impact on accuracy (all results shown in Table 4.2 are with hyperclustering).

We measured the training time taken in each experiment and include these results in Table 4.2. These times are very reasonable even for data sets as large as the MNIST (LeCun et al., 1998) testbed.

#### 4.4 Experiments

The goal of the experiments in this section is to evaluate the efficacy of our algorithm and compare the results to other methods such as LIFTKM (Raman et al., 2011) (dimensionality  $\rho = 2000$ ), the bipartite graph methods of Fern and Brodley (2004), and the hypergraph methods of Strehl and Ghosh (2003). We applied all these methods to standard labeled classification data sets (detailed in Table 4.1) that are used in the literature in evaluation of clustering algorithms and ensemble clustering techniques.

We generated the ensemble using a combination of the methods described in Section 4.3 and using the k-means algorithm (MacQueen et al., 1967), spectral

Data Set	Iris	Wine	Ionosphere	Soybean
<b>SEC error</b>	<b>10.67%</b>	<b>29.78%</b>	<b>28.77%</b>	<b>29.79%</b>
LiftKM error	11.33%	37.64%	33.90%	<b>29.79%</b>
KRF error	<b>10.67%</b>	53.37%	<b>28.77%</b>	<b>29.79%</b>
Fern-Brodley error	<b>10.67%</b>	42.13%	32.76%	<b>29.79%</b>
Mean error	15.68%	32.64%	27%	29.1% %
Min error	10.67%	29.78%	0%	0%
Diversity	0.11	0.19	0.06	0.24
SEC Time	11.6s	16.4s	10.5s	3.6s

Data Set	ISOLET	MNIST (Test)	MNIST (Full)
<b>SEC error</b>	<b>41.24%</b>	<b>40.03%</b>	<b>42.64%</b>
LiftKM error	46.76% ( $\rho = 4000$ )	63.94%	64.13%
KRF error	43.43%	-	-
Fern-Brodley error	44.26%	-	-
Mean error	48.91%	52.88%	51.92%
Min error	42.85%	50.40%	50.82%
Diversity	0.29	0.62	0.66
SEC Time	860s	483s	6733s

Table 4.2: **Results.** This table presents the results of applying different ensemble clustering algorithms on standard data sets. The first row contains the data set used, the second the error rate for our method (**SEC**), and the third the error rate of **LIFTKM**. The fourth row contains the least error rate of the knowledge reuse framework (KRF) of Strehl and Ghosh (2003) and the fifth the least error rate among the three methods proposed in Fern and Brodley (2004). The sixth and seventh rows contain the mean and minimum errors respectively of members of the ensemble relative to the “true” labeling of the data set. The eighth row displays the diversity measurement of the ensemble and the final row the time in seconds that SEC took to arrive at a consensus clustering on that data set.

clustering (Ng et al., 2002), and affinity propagation (Dueck and Frey, 2007). We generated hundreds of clusterings and set a threshold of 0.1 for the diversity (defined in Section 4.3) in order to proceed with the final consensus step. Each ensemble clustering algorithm was given the same ensemble from which to derive its consensus clustering, wherever applicable.

#### 4.4.1 Measuring Accuracy

A natural criterion to measure the usefulness of each method is the overall accuracy (with respect to provided ground truth) of the final consensus clustering and the improvement over individual members of the ensemble. We define accuracy by the following formula:

$$Accuracy = \max_p \frac{1}{n} \sum_{i=1}^k T(C_{p(i)}, L_i) \quad (4.2)$$

where  $L_i$  is the  $i^{\text{th}}$  class in the labeled data set,  $C_j$  is the  $j^{\text{th}}$  cluster in the consensus clustering,  $p$  varies over all permutations of labeling assignments between the clusters of the consensus clustering and the classes of the data set, and  $T(C_j, L_i)$  is the number of points that occur in both  $C_j$  and  $L_i$ .

We approximate the best correspondence  $p$  of cluster labels from the consensus clustering to the data set labels by solving the correspondence problem using the Hungarian algorithm (Munkres, 1957). The “accuracy” of each clustering with respect to the given labels is then computed using this correspondence.

It is useful to keep in mind here that on these data sets supervised methods are likely to obtain better accuracies. However, the utility of these experiments is to demonstrate that information from multiple partitions can be synthesized in a principled manner to create a more robust partition, still without any supervision. The utility of ensemble clusterings is with unlabeled data sets where supervised algorithms are inapplicable.

#### 4.4.2 Results

Table 4.2 shows the results of applying the above-mentioned ensemble methods to data sets from the UCI Machine Learning Repository Asuncion and Newman (2007), and the MNIST digit recognition database (60,000 data in 784 dimensions, categorized into 10 classes). These data sets were chosen to maximize the diversity in dimensionality, numbers of classes, and numbers of instances, to demonstrate the wide applicability of ensemble clustering.

In each column of the table we show results from the application to one data set of SEC and three other state-of-the-art ensemble clustering algorithms, each representative of a different approach to computing the consensus. In the columns we show the error of each algorithm rather than its accuracy to better demonstrate differences in performance. The least error in each case is shown in bold. Additionally, we also provide information such as the mean and minimum error of all clusterings in the ensemble. Finally, we show the running time of SEC on that data set.

As the bolded numbers show, SEC consistently finds a consensus clustering that has the least error rate among all the methods tested. In the case of the Iris data set, three of the four methods arrived at a consensus that has a mis-clustering error lower than the mean error over the ensemble, and equal to the minimum error. The data set Wine has a similar result for SEC. For the Ionosphere and Soybean data sets all methods perform similarly and arrive at clusterings that are comparable to the mean error in the ensemble. In the ISOLET and MNIST data sets, SEC reports an error that is significantly lower than the mean of the ensemble and, more importantly, also lower than the minimum error. In the case of MNIST, the difference is as much as 10.37%, corresponding to an error reduction of 20.58%. This is especially interesting as it indicates that with no further supervision we are able to use information

from disparate clusterings to reduce the mis-clustering error well below even the best performing member of the ensemble.

## 4.5 Conclusion

Following the theme of spatial sensitivity in comparing clusterings from the previous chapter, in this chapter we proposed an end-to-end ensemble clustering algorithm that is sensitive not only to the spatial layout of the data but of the clusters and clusterings as well. Our algorithm allows for domain-specific ground distance functions to be used on the data points. It outperforms other state-of-the-art algorithms on standard data sets as measured by accuracy on labeled data sets, and in some cases is even able to arrive at a partition that has a higher accuracy than the best-performing member of the ensemble.



## 5 NEUROIMAGING

---

*This chapter delves into neuroscience, an area that may be unfamiliar to many readers. The introductory parts of this chapter (Sections 5.2-5.5) are therefore devoted to providing background information about Alzheimer's disease and brain matter in order to provide context about the experimental data. Following these sections we demonstrate how methods used in this thesis can detect subtle changes in the brain that are of clinical relevance.*

Alzheimer's disease (AD) is a form of dementia that affects 18 to 24 million people worldwide (Prince et al., 2011). By primarily affecting memory and executive function it greatly reduces the quality of life of people afflicted with it. As the world's population ages rapidly in the coming few decades this number is projected to rise manifold to over 70 million by 2050. In this chapter we address the problem of detecting subtle changes in neural structure that are indicative of cognitive decline and correlate with risk factors for Alzheimer's disease. This is done by studying structural imaging data derived from middle-aged patients with and without cognitive impairment. We will use images from brain scans of a diverse set of subjects to study questions relating to structural differences between patients from different populations. These questions develop a progressive framework that allow classification to be performed on individuals, as opposed to methods that determine statistical differences at the group level. Through this we aim to advance the state-of-the-art in understanding how neural change is related to age, memory, and cognitive function. Figure 5.1 shows a visualization of the difference between views of a healthy brain and one with advanced Alzheimer's disease.

### 5.1 Framework

In the remainder of this chapter we show how longitudinal neuroimaging analysis can be conducted within a point set framework to solve a number of

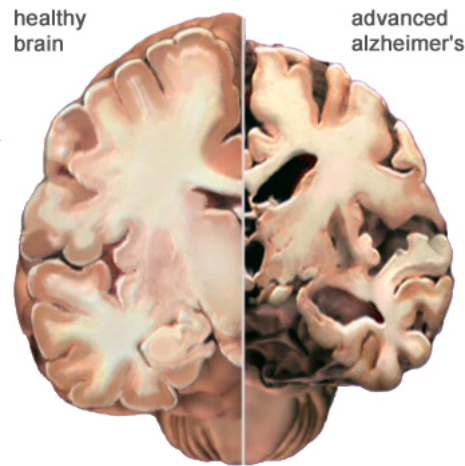


Figure 5.1: **Alzheimer's Brain** This figure shows a diagram of two halves of an axial cross-sectional view of the human brain. The left half depicts a healthy brain, and the right half a brain in an advanced state of Alzheimer's disease. As the figures shows, Alzheimer's disease is accompanied by significant loss in volume in both gray and white matter regions of the brain. This figure is taken from Alzheimer's Association, 2015b.

problems in this domain. Our approach is significantly different from current and previous work in this area. Previous approaches to the problem of detecting changes in brain matter, and relating structural features of brain scans to clinical observations have focused on separating populations based on gross changes, such as decreasing overall volume of matter (Magnin et al., 2009; Grydeland et al., 2013) or on voxel<sup>1</sup>-based comparisons (Klöppel et al., 2008; Dyrba et al., 2013; Haller et al., 2013; Gray et al., 2013; Bron et al., 2015; Sun et al., 2015). In contrast, we apply the spatially-sensitive kernels developed in Chapter 2 that allow us to characterize individuals, as opposed to populations, and detect minute differences among them. The problem of detecting structural differences in white matter over time or across different populations of patients thus lends itself neatly to the application of techniques that measure spatial overlap. We use this for both classification and regression, for example, to predict changes

<sup>1</sup>A voxel represents a location in a three-dimensional grid, similar to a pixel in two dimensions.

in a participant's cognitive test scores over time using neuroimaging data alone. This is a difficult problem, and in solving it, this work has been able to identify neural regions that are implicated in cognitive performance and change over time.

More generally, our approach introduces a simple paradigm for addressing *wide-data* longitudinal problems. It is not specific to neuroimaging analysis and shares a number of properties that are representative of this class of problems and which arise often in medical and related domains. These properties include:

1. The data sets are *wide* – they have many more features ( $p$ ) than they do samples ( $N$ ). For example in an MRI study, we may gather  $O(1e6)$  voxels for each of 100 patients. Similarly, a genome-wide association study may have 500,000 single nucleotide polymorphisms (SNPs) measured over a similar number of patients. Because  $p \gg N$ , linear models are often the tool of choice due to their speed and low variance. However, these models are also often extremely sparse, as described next.
2. Longitudinal studies track changes over time, with the goal of correlating significant features with some outcome or effect. Naturally occurring variations across features can mask these correlations. For example in medical studies based on neuroimaging, most neural variation is non-pathological and unrelated to the study outcome. The desired model is therefore often extremely sparse but identifying significant features may be difficult due to the next issue.
3. We often lack ground truth to validate results. Consider the problem of determining whether healthy participants tracked over time are expected to develop some condition, such as AD. Given the subjects are currently healthy, even if issues (1) and (2) could be ignored, we have few ways to validate any constructed models. Instead, results are often

presented as hypothesis tests that distinguish populations, e.g. those with a family history of the disease from control groups. Predictions about specific individuals are therefore elusive, outside of summary statistics for populations of which they are members.

4. It is increasingly common to track longitudinal changes over very short periods of time. In human neuroimaging, this interval has become as short as three months (Alzheimer's Disease Neuroimaging Initiative, 2003). One may ask if there is even a "signal" to find here. How do we know if there is anything meaningful to detect? This is exacerbated when the sampling time frame is much shorter than the onset time of observable phenomena we would like to predict.

### 5.1.1 Analysis

We focus our analysis on middle-aged participants from the Wisconsin Registry for Alzheimer's Prevention (Sager et al., 2005) who underwent both brain imaging and cognitive testing twice over a span of approximately two years. The main neuroimaging metric of interest is fractional anisotropy (FA) as indexed by magnetic resonance diffusion tensor imaging (MR-DTI or DTI). DTI is a very effective means of identifying and measuring the integrity especially of one kind of tissue in the brain known as white matter. For expositional purposes we focus on white matter in this chapter; the computational methods however are just as applicable to gray matter — the other type of matter in the brain. Based on prior cross-sectional work from our group, we expected that AD risk and cognitive function would be related to white matter microstructure. Specifically, we hypothesized that FA would decline over the two years, that we would find that participants with Alzheimer's risk factors such as Apolipoprotein (APOE)  $\epsilon 4$  genotype would show a greater change over time compared to non-carriers,

and that changes in cognitive function over time would be represented by alterations to white matter microstructure over time. Given that the population was cognitively healthy, we expected that the observed changes in both white matter microstructure would be subtle. This problem therefore is well-suited to benefit from the methods introduced in Chapter 2 that are able to detect minute changes in samples drawn from similar, yet different, distributions.

As is common in many classification problems, most previous work in machine learning applied to neuroimaging data abstracts voxel data into vector representations that fail to retain spatial information (Dyrba et al., 2013; Gray et al., 2013; Schnack et al., 2014; Klöppel et al., 2008). Given the inherently spatial nature of the voxel data, we hypothesize that incorporating voxel locations into our analysis can boost accuracy in a number of experiments. Rather than serialize the voxels of a brain or region into one vector and lose their locations, we represent them as a point set  $B = \{(v_1, w_1), (v_2, w_2), \dots, (v_N, w_N)\}$  (see Chapter 2) where each  $v_i \in \mathbb{R}^3$  is a voxel position,  $w_i \in \mathbb{R}$  its weight (corresponding to some observed value at that voxel), and  $N$  is the number of voxels in the brain.

From the general problem of characterizing neural change with respect to age and cognition we distill three more concrete and specific problems:

1. Predict the chronological order of two scans from the same subject. This will enable us to identify regions that change with age.
2. Predict the presence or absence of the APOE gene based on longitudinal changes in brain scans.
3. Predict the direction of change of cognitive performance based on longitudinal changes in brain scans.

These experiments demonstrate application of our framework to detecting minute, short-term changes in WM structure and relating them to changes in

cognitive test scores and genetic biomarkers. They present the first evidence demonstrating that very small changes in white matter structure over a two year period can predict change in cognitive function in healthy adults.

## 5.2 Alzheimer's Disease

Alzheimer's disease (AD) is a brain disorder that is a type of dementia, a set of conditions wherein brain cells die or do not function normally. It affects 18 to 24 million people worldwide (Prince et al., 2011). As the world's population ages rapidly in the coming few decades this number is projected to rise manifold to over 70 million by 2050. The problems we seek to solve in this chapter fall within a larger aim of understanding patterns of neural decay and cognitive decline in populations at risk of developing Alzheimer's disease.

Alzheimer's disease constitutes 50% to 80% of all dementia cases (Prince et al., 2011; Alzheimer's, 2015). It affects the brain by damaging and destroying brain cells (neurons), causing loss of memory, thinking, and other executive functions and thereby greatly reduces the quality of life of people afflicted with it. It is the sixth leading cause of death in the United States. It is a "progressive" disorder in that it gets worse with time, to the point where individuals with an advanced stage of the disease are frequently unable to conduct simple conversations. AD involves the development of protein buildups called beta amyloid plaques and neurofibrillary tangles in the brain that are toxic to nerve cells. The exact cause of AD is unknown; however multiple factors have been identified as contributing or leading to AD. In addition, the precise physiological changes in the brain that lead to AD in patients are also unknown. It is however established that changes in the brain leading to AD usually begin in the region responsible for dealing with short term memory and newly acquired information. Alzheimer's Disease ultimately leads to death and there is no

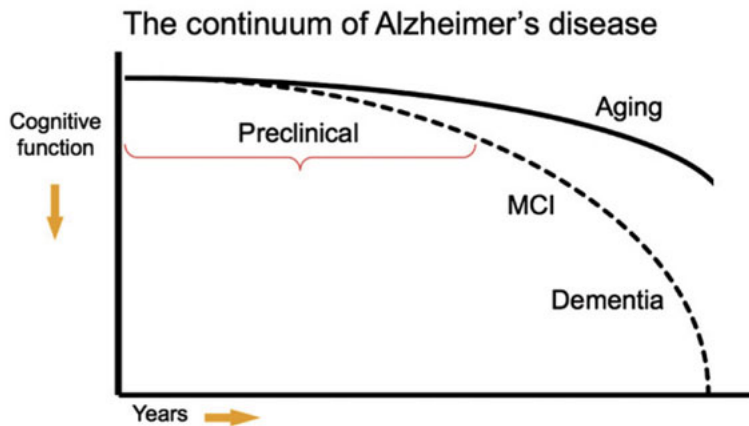


Figure 5.2: A visualization of the modern understanding of the three main stages in the continuum of Alzheimer's disease (AD) development and their relation to cognitive deficits. The first stage, preclinical AD, does not involve any outward symptoms of cognitive impairment or dementia but is characterized by physiological changes in the brain. This precedes mild cognitive impairment (MCI) which may be followed by dementia due to AD. This figure is taken from Sperling et al. (2011).

"cure" as of now, nor has any treatment been identified to stop or retard its progress (Alzheimer's, 2015).

### 5.2.1 Stages of AD

Modern understanding of AD views its progression as being along a continuum (Alzheimer's, 2015). At the beginning the individual functions as normally as before and at the end of the continuum they experience memory loss, memory change, and situational confusion. This understanding is reflected in the 2011 Alzheimer's Association recommendations for diagnostic criteria of AD Sperling et al. (2011), which demarcates three regions on the continuum, representing *preclinical AD*, *mild cognitive impairment (MCI)*, and *dementia due to AD*. Figure 5.2 shows these stages and the cognitive characteristics related to each one. In preclinical AD, the individual exhibits physiological changes in the brain and biomarkers such as in blood and cerebrospinal fluid (CSF) but

there are no outward symptoms. With MCI, the individual begins exhibiting symptoms of memory loss but is still able to function at a high level. In the final stage, the individual's ability to function normally is impaired. It is worth noting that not all individuals who develop MCI go on to develop full-fledged Alzheimer's Disease; in fact it is an area of active research to explore what characterizes these patients (Lindemer et al., 2015; Davatzikos et al., 2011).

### 5.2.2 Risk factors for AD

A number of attributes predispose one to developing Alzheimer's disease. These risk factors include age, family history, presence of a specific allele in the gene apolipoprotein E, risk factors for cardiovascular disease, and diet. We discuss some of them briefly below:

#### **Age**

The largest risk factor for AD is age. However, AD is not considered a part of the normal aging process. The more advanced an individual is in age, the higher they are at risk for developing AD. At the age of 65, the risk for developing AD is 12-13% and thereafter doubles every five years (Prince et al., 2011).

**Family History** Individuals with a history of AD in a closely related family member (parents or siblings) can be up to two to three times more likely to develop AD than subjects with no family history (Prince et al., 2011). This risk increases with number of close family members who developed AD.

**Genotype: APOE  $\epsilon$ 4 and TOMM40** The gene apolipoprotein E (APOE) codes for a protein that carries cholesterol in the bloodstream. Studies have shown that when this gene is present in an individual in the  $\epsilon$ 4 allele form, it places them at higher risk for developing AD. It has been estimated that presence of this allele accounts for 50% of the AD population (Ashford, 2004). This factor



is related to the family history factor in that an individual inherits one copy of this gene from each parent. If the inherited copies in both chromosomes of an individual are  $\epsilon 4$ , he/she is at up to 15 to 20 times higher risk for developing AD (Ashford, 2004) than an individual with only copies of the  $\epsilon 2$  or  $\epsilon 3$  allele. In the experiment detailed in Section 5.7 we examine whether the presence or absence of the  $\epsilon 4$  allele leads to a difference in the way WM changes over time.

Recent studies have identified another gene known as TOMM40 on the same chromosome in close proximity to the APOE gene whose length has been correlated to age of onset of AD in individuals with one APOE  $\epsilon 3$  allele (Roses et al., 2010).

### 5.3 White Matter

Solid matter in the brain is conventionally classified first into gray and white matter (Purves, 2012). Gray matter (GM), consisting of neuronal cell bodies, has been the subject of many decades of research, controls muscle control, perception, memory, speech and a host of other functions. White matter (WM), on the other hand, consists of the tissue that is responsible for the conduction of messages between different gray matter regions. Figure 5.3 shows an axial slice of the brain with marked WM and GM regions. WM constitutes about half of the matter in the brain and is composed of bundles of insulated nerve fibers called axons. The insulation is myelin, made of fatty tissue, and it is what gives white matter its distinctive pink-white color. Myelin continues growing and insulating axons long after the rest of the brain stops growing. As mentioned earlier, the most of the experiments described later in this chapter were performed on white matter regions of the brain, but the underlying techniques are equally applicable to gray matter regions as well.

Much previous research on Alzheimer's disease has focused on gray matter; white matter has historically been regarded as less relevant to cognition. In

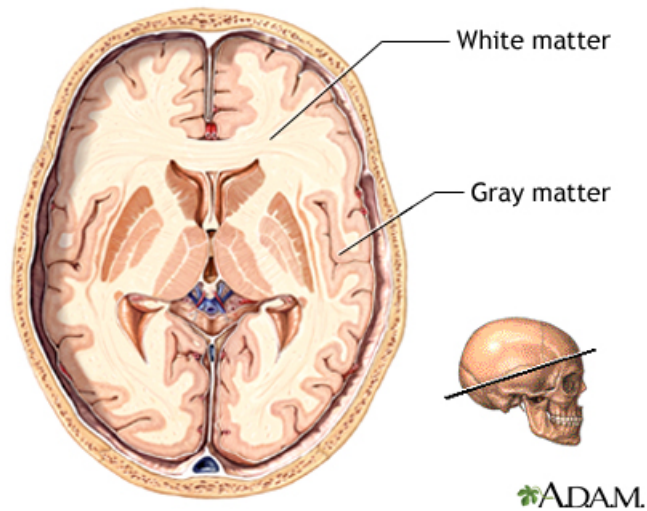


Figure 5.3: **Types of Brain Matter.** This figure shows a diagram of an axial cross-sectional view of the human brain. The types of regions corresponding to gray and white matter are labeled in the figure. This figure is taken from Ropper et al. (2009).

recent years, however, the role of white matter in the transfer of information has attracted vigorous interest (Ziegler et al., 2010). For example, the degree of myelination in white matter has been found to correlate with cognition, IQ, executive function, and learning (Fields, 2008). WM is also highly relevant in understanding deterioration in cognitive control and episodic memory, which were found to be accompanied by marked changes in specific white matter structure (Ziegler et al., 2010). Recent studies also show that risk factors for AD such as the apolipoprotein E  $\epsilon 4$  genotype and parental family history are associated with WM changes in the brain (Bartzokis et al., 2007; Bendlin et al., 2010b). Studies have shown that when this gene is present in the  $\epsilon 4$  allele form, individuals are at higher risk for developing AD (Ashford, 2004).

In the sections below we describe the mechanism that is used to develop accurate representations of white matter in the brain and the kind of data that are derived from this imaging process. The most popular imaging technology

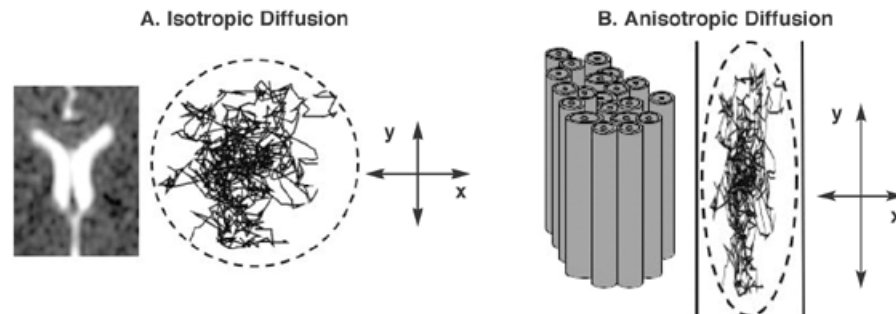


Figure 5.4: **Isotropic and anisotropic diffusion.** (A) Water molecules in the brain are constantly moving (i.e., in Brownian motion). When motion is unconstrained, as in the large fluid-filled spaces deep in the brain (i.e., the ventricles, as illustrated in the MR image on the left), diffusion is isotropic, which means that motion occurs equally and randomly in all directions. (B) When motion is constrained, as in white-matter tracts (illustrated on the right), diffusion is anisotropic, meaning that motion is oriented more in one direction than another (e.g., along the y axis rather than along the x axis). This picture and caption are taken from Rosenbloom et al. (2003).

used today to construct representations of white matter uses the same principles as used in nuclear magnetic resonance imaging (MRI). MRI uses a homogenous magnetic field to excite water molecules, which then spontaneously de-excite emitting radiation that is captured and measured. These measurements are then pieced together to infer soft tissue structure. The following sections describe how this technology is modified in the case of measuring diffusion of water molecules in white matter and the format of measurements produced.

### 5.3.1 Diffusion Tensor Imaging

Diffusion is a physical process by which particles move from one region to another by means of random motion and without a process of mass transport. It occurs due to a difference in concentration in the diffusing molecules in the two regions. In a three-dimensional context, diffusion can be shown to progress along the surface of an ellipsoid. In white matter, water molecules show higher diffusion along a nerve fiber than perpendicular to it; this is called

anisotropic diffusion. In gray matter and in fluid-filled ventricles, by contrast, diffusion occurs to a similar extent along all directions (i.e. it is isotropic). Figure 5.4 shows a depiction of the difference between isotropic and anisotropic diffusion of water in the brain. The difference in diffusion properties is the principal insight used in reconstruction of white matter structure by *diffusion tensor magnetic resonance imaging* (DT-MRI) or *diffusion tensor imaging* (DTI).

DTI uses bipolar magnetic field gradient pulses to measure diffusion effects arising out of the Brownian motion of water molecules in the brain. In DTI, pulses are applied with a time delay where the second pulse undoes the magnetic alignment effects of the first pulse. However, for water molecules that move between the two pulses, this second pulse will not completely undo the magnetization effect of the first pulse. It is this difference that causes a reduction in the signal received by the MRI machine and provides clues about the diffusion freedom available to water molecules in each voxel. Diffusion, being a highly directional process, needs to be measured along at least 6 (in practice, typically 40) directions in order to get a full three-dimensional profile of the diffusion taking place in an area. This profile will be referred to as a “DTI image” henceforth.

Le Bihan et al. (2001) explains that in their random diffusive movement, water molecules probe tissue structure while interacting with cell membranes and other entities in the brain. A voxel in the diffusion MRI image thus measures the displacement distribution of water molecules within that voxel, which in turn elucidates the tissue structure.

In addition, DTI is capable of measuring the *anisotropy* (inequality with respect to direction) of diffusion of the water molecules. This information is available due to the fact that it measures molecular displacements along a particular direction — the direction of the magnetic field gradient — in up to 40 different directions. This anisotropy information is utilized to construct

a *diffusion tensor*, which is an order 2 tensor describing molecular mobility along 3 directions, and correlations among mobilities along those directions. These diffusion tensors and anisotropy information they encode elucidate the structure and organization of bundles of myelinated axonal fibers (i.e. white matter) in the brain. In this way, DTI is helpful in assessing neuronal fiber tract integrity as a whole in the brain (Dyrba et al., 2012). Studies have shown that DTI is more sensitive than structural MRIs to detecting memory changes and Alzheimer’s Disease in patients (Carlesimo et al., 2010).

### 5.3.2 Diffusion Tensor

As mentioned above, a diffusion tensor encodes information on a per-voxel basis and is a  $3 \times 3$  matrix that fully characterizes the diffusion properties of that voxel (Hagmann et al., 2006). A voxel in our context is a three-dimensional volume element of a regular grid, or a coordinate system. The dimensions of one of the standard coordinate systems (MNI — from the Montreal Neurological Institute) are  $182 \times 218 \times 182$ . Of these voxels, approximately 500,000 voxels correspond to white matter and a comparable number of voxels to gray matter.

The complex pattern of displacements of molecules as modelled by a random diffusion process can be described with nine components — each one as a diffusion coefficient associated with a pair of axes  $xx$ ,  $yy$ ,  $zz$ ,  $xy$ ,  $yx$ ,  $xz$ ,  $zx$ ,  $yz$ , and  $zy$ . Since this process is symmetric, in reality there are only six independent components and the matrix is symmetric:

$$\bar{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}$$

This matrix can be thought of as representing an ellipsoid with the magnitude of the axes and its orientation representing the diffusion that it models.

The first eigenvector of this matrix corresponds to the principal direction of diffusion and the eigenvalues correspond to the magnitude of diffusion.

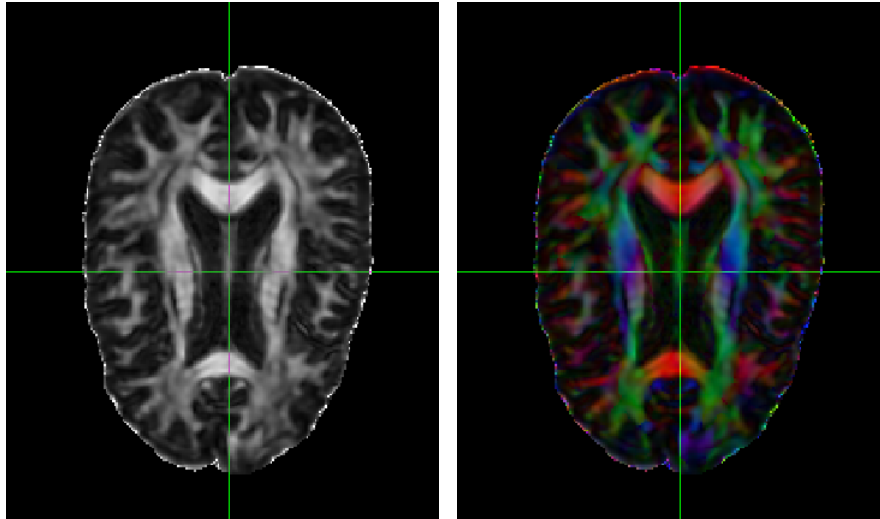


Figure 5.5: **Fractional Anisotropy.** This figure depicts the fractional anisotropy values per voxel for voxels in an axial cross-section of a subject's brain. (a) The intensity values for each voxel shown in this image corresponds to the FA value. Darker voxels mean a lower FA value and lighter voxels mean higher FA. (b) This figure also shows the voxel-wise FA values, but in this case they are overlaid with direction information encoded by color. Red signifies the left-right (sagittal) direction, green the front-back (coronal) direction, and blue the head-foot (axial) direction.

### 5.3.3 Summary Measures

In addition to the full tensors for each voxel, there are a number of scalar summary measures that extract and condense different types of information contained in the vector, such as information pertaining to isotropy and anisotropy, diffusivity, and tract organization (Basser and Pierpaoli, 1996). These summary measures have been found to be relevant and useful as representations and encodings of white matter structure for assessing and characterizing developmental processes in the brain. It is also worth noting that these summary

measures are invariant with respect to the orientation of the coordinate system used; in particular they do not vary due to different acquisition parameters or hardware. We focus on two summary measures: one to characterize the degree of directionality of diffusion and another to characterize the magnitude of the diffusion coefficient.

### **Fractional Anisotropy**

One such DTI summary measure is *fractional anisotropy* (FA), which is a scalar measure of how directional the diffusion of water molecules is in a voxel. In other words, FA describes the *degree* of a diffusion process (Mori, 2007) but does not contain any direction information. It measures tissue integrity and is sensitive to axon fiber density and myelination of axons. FA is a measured between 0 and 1, with 0 representing perfectly isotropic (equal in all directions) diffusion and 1 representing anisotropic diffusion that is characteristic of highly ordered white matter fiber bundles). FA characterizes the *degree* of a diffusion process Mori (2007) but does not contain any direction information. FA at a voxel is defined (Hagmann et al., 2006) as:

$$FA = \sqrt{\frac{3}{2}} \sqrt{\frac{(\lambda_1 - \hat{\lambda})^2 + (\lambda_2 - \hat{\lambda})^2 + (\lambda_3 - \hat{\lambda})^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the eigenvalues of the diffusion tensor and  $\hat{\lambda}$  is the mean of the eigenvalues. FA thus is a comparison of how different each eigenvalue is from the mean and is higher when one or two eigenvalues dominate the rest, i.e. when the diffusion is highly directional and anisotropic. FA can also be seen as a measure of the ellipsoid eccentricity if the tensor is viewed as representing an ellipsoid. Figure 5.5 shows an axial slice of FA values from a subject.

### **Mean Diffusivity**

Mean diffusivity (MD) in a voxel measures the local magnitude of diffusion (regardless of direction) through the mean squared displacement of molecules.

Higher organization of white matter corresponds to lower MD, while a breakdown in white matter integrity leads to higher MD values (Bennett et al., 2010). It is defined as the mean of the eigenvalues of the diffusion tensor:

$$\text{MD} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}$$

## 5.4 Literature Review

The past decade has seen vigorous interest in the application of machine learning techniques to neuroimaging data. We will focus in this section on providing a brief overview of work related to AD, MCI, and identifying structural differences between populations with AD, MCI, and risk factors for AD.

Up until recently, Alzheimer’s Disease and cognitive impairment were thought of as primarily gray matter phenomena. Research therefore has concentrated on studying the gray matter changes associated with AD progression. Recently however, there has been significant interest in studying the white matter aspects of AD. A large majority of neuroscience research on Diffusion Tensor Imaging (DTI) data and Magnetic Resonance Imaging (MRI) data in general has clustered around two approaches, neither of which exploits spatial information: voxel-wise comparisons of tensors (or summary measures) and summary statistics on regions of interest.

The first approach involves registration of all images to a common atlas, followed by statistical comparisons of corresponding voxels in all subjects; this is called the voxel-direct approach (Cuingnet et al., 2011). The majority of these techniques are for cross-sectional experiments, not longitudinal. The learning step in methods found in the literature following this approach typically consists of feature selection and classification via support vector machines (SVMs) using linear or radial basis function (RBF) kernels, random forests, or Naive Bayes. These methods are not specific to white matter; in fact the



experiments largely focus on weights derived from gray matter regions with white matter analysis largely an afterthought, if conducted at all. The factor that differentiates methods from one another is the feature selection step. Klöppel et al. (2008) presented the first application of SVMs to classify AD patients from healthy controls, and AD patients from another class of patients with frontotemporal lobar degeneration (FTLD). In this method, there is no feature selection involved, and the T1-weights from MRI scans were used to train a linear SVM directly. Voxels can then be ranked according to discriminative power by multiplying each image by its label (+1 or -1) and weight according to the trained SVM and adding all the images. The resulting image contains a weight for each voxel reflecting its importance in the classifier. Vemuri et al. (2008) follows a similar procedure, with downsampling and additional feature selection steps that discard negatively-weighted voxels. Dyrba et al. (2013) and Dyrba et al. (2012) proposed a variance reduction step via principal components analysis (PCA), followed by a feature selection step based on ranking the voxels by their information gain. This method was applied to both gray and white matter maps to classify healthy control subjects from those with probable AD. Dyrba et al. (2012) in addition discusses the use of multi-kernel methods for the same problem. Haller et al. (2013) performed a cross-sectional study to classify various sub-kinds of MCI patients using a voxel-direct method. In this study feature selection is done through the Relief-F algorithm Kononenko et al. (1997), and training via an SVM with RBF kernels. Falahati et al. (2014) and Cuingnet et al. (2011) provide surveys of related voxel-direct techniques along with comparisons between them. Bron et al. (2015) propose a voxel ranking method based on the "p-map," a map constructed from the p-value of every voxel (computed via permutation testing) with respect to the null distribution on the weight vector of a trained SVM (discussed above in context of the method

in Klöppel et al. (2008)). Sun et al. (2015) proposed a lasso method (combining the  $\ell_1$  and  $\ell_2$  norms) for feature selection and simultaneous classification.

The second broad class of methods involves aggregate measurements within a specified region of interest (ROI), such as means of FA values or WM intensities. These are called “atlas-based” methods. Magnin et al. (2009) classify AD patients from elderly controls by training an SVM with an RBF kernel on feature vectors constructed by estimating the relative gray matter content for each of 90 regions of interest in the brain. Grydeland et al. (2013) presents a longitudinal study using logistic regression on means of WM and GM changes aggregated in regions of interest to classify AD patients from age-matched controls. Desikan et al. (2009) follow a similar procedure, but for a cross-sectional study to separate individuals with MCI from healthy controls.

Other work has taken the approach of analysis on manifolds constructed from the diffusion tensors. Khurd et al. (2007) describes a method of voxel-based analysis on manifolds that are learned from diffusion tensors. Fletcher et al. (2007) uses a path integration based approach to derive connected pathways between regions and then derive summary volumetric measures along those pathways. Corouge et al. (2006) proposes a framework for analysis in which tracts, not voxels, are the units of analysis. This method uses tensor statistics to model and manipulate diffusion tensors, and defines new summary statistics on bundles of fibers to compare DTI images.

Adluru et al. (2009) uses a histogram-based method called “3-D shape context” to characterize fiber tracts as geometric curves and then performs machine learning analysis on them using the histogram profiles as features. A similar approach is taken by Chung et al. (2010) but the geometric model of a curve there is a cosine series representation.

## 5.5 Study Data

The data for our experiments come from studies being conducted by the Wisconsin Registry for Alzheimer's Prevention (Sager et al., 2005). WRAP is a longitudinally followed cohort comprising participants who either have a family history of late onset AD or no family history of AD. Its purpose is to gain a better understanding of the processes that occur during pre-clinical stages of Alzheimer's disease in those at risk. This understanding can contribute to the development of future diagnostic methods that identify AD at earlier stages. The majority of the WRAP participants are adult children of persons with AD who were evaluated at the Memory Assessment Clinic at the University of Wisconsin-Madison or satellite memory assessment clinics affiliated with the Wisconsin Alzheimer's Institute, and other participants who learned about the study from educational presentations, health fairs, newsletters, or word of mouth. The study includes participants with parental family history and genetic risk for AD, specifically, positive Apolipoprotein E  $\epsilon$ 4 (APOE4) status.

Each subject in WRAP undergoes regular brain imaging procedures and neuropsychological cognitive tests that measure memory, executive function, speed, and other related items. The cognitive tests are designed to measure those quantities that suffer with the onset of AD or MCI, such as episodic memory and delayed recall ability which measure the ability to learn over time and retain the learning. Additionally, patient data and medical data such as age, family history, results from genotyping tests, and measurements of vitals are also collected.

### 5.5.1 Participants

We enrolled 128 participants, ten of whom were excluded due to unexpected abnormalities found by the reviewing radiologist. The remaining 118 middle

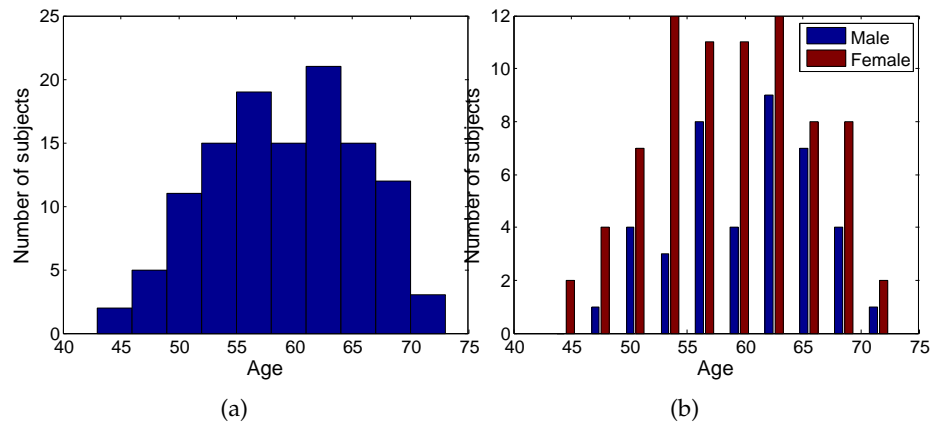


Figure 5.6: (a) A histogram of ages of all study participants. (b) A histogram of study participant ages broken down by gender.

to older-aged participants were 43 to 73 years of age at the time of the first scan (mean = 59.8, SD = 6.58); and 45 to 75 years of age at the time of the second scan (mean = 61.4, SD = 6.8). There were 41 men and 77 women. A histogram of subject ages, and subject ages by gender is shown in Figure 5.6. The inclusion criteria for this study consisted of:

- normal cognitive function determined by neuropsychological evaluation (MMSE<sup>2</sup> ≥ 25)
- no contraindications for magnetic resonance imaging (MRI) and a subsequent normal MRI scan
- no current diagnosis of major psychiatric disease or other major medical conditions (e.g., myocardial infarction, or recent history of cancer)
- no history of head trauma, stroke or transient ischemic attack.

Brain images were reviewed by a neuroradiologist to exclude infarcts and other abnormalities.

<sup>2</sup>Mini Mental State Examination (MMSE) is a quick way of quantifying cognitive function and screening for cognitive loss (Folstein et al., 1975)

Longitudinal imaging and cognitive testing data were available for these 118 subjects, who were healthy and all tested cognitively normal on neuropsychological assays. A significant percentage (78%) of subjects showed one or more risk factors for AD, such as parental family history, or presence of the apolipoprotein E allele. In addition to imaging data, each subject provided extensive demographic information.

### 5.5.2 Imaging

Participants were imaged on a General Electric 3.0 Tesla Discovery MR750 (Waukesha, WI) MRI system with an 8-channel head coil and parallel imaging (ASSET). DTI was acquired using a diffusion-weighted, spin-echo, single-shot, echo planar imaging pulse sequence in 40 encoding directions at  $b = 1300$ , with eight non-diffusion weighted ( $b = 0$ ) reference images. The cerebrum was covered using contiguous 2.5 mm thick axial slices, FOV = 24 cm, TR = 8000 ms, TE = 67.8, matrix = 96 x 96, resulting in isotropic 2.5 mm<sup>3</sup> voxels. High order shimming was performed prior to the DTI acquisition to optimize the homogeneity of the magnetic field across the brain and to minimize EPI distortions.

### 5.5.3 Neuropsychological Tests

All participants underwent comprehensive neuropsychological testing. Cognitive factor scores were derived from a factor analytic study of the WRAP neuropsychological battery and adapted from work published by Dowling et al. (2010). Based on prior studies (Kerchner et al., 2012; Madden et al., 2012; Lövdén et al., 2010; Bendlin et al., 2010a; Birdsill et al., 2013) showing a strong relationship between indicators of white matter health and processing speed, the factor score of interest chosen for our experiment was the *Speed and Flexibility* factor, a composite measure based on the interference trial from the Stroop

Test (Trenerry et al., 1989), and Trail Making Test A and B (Reitan and Wolfson, 2009).

#### 5.5.4 Preprocessing

We utilized a standard series of preprocessing steps to construct tensor and FA maps for each patient. To make the images comparable, we applied the registration steps of Tract-Based Spatial Statistics (TBSS) (Smith et al., 2006). TBSS performs nonlinear registration to a template image followed by transformation to MNI152 standard space. Each scan underwent identical preprocessing. Smoothing<sup>3</sup> was not applied. Using the white matter atlas from the Johns' Hopkins University research group (Oishi et al., 2008), we extracted mean FA and MD values from the voxels corresponding to the corpus callosum, superior longitudinal fasciculus, fornix, and cingulum bundle. These regions were chosen based on their vulnerability to Alzheimer's disease (Di Paola et al., 2010; Benitez et al., 2014; Canu et al., 2013). The sizes of these regions range from hundreds to over 20,000 voxels. Figure 5.7c shows a three-dimensional view of the location and shape of one of the regions we analyze: the splenium of the corpus callosum (over 12,000 voxels), a region known to show significant changes both in healthy aging and AD.

#### 5.6 Before vs. After

Our approach begins with a "simple" classification problem. For longitudinal data, one instance of ground truth is the chronological order in which the data sets were collected. Thus, a natural question is: can we determine this order for a given individual (see Figure 5.8 for an example)? In other words, given two scans, our task is to identify which was taken earlier. The aim of

---

<sup>3</sup>A blurring technique sometimes applied to neuroimaging data as part of the preprocessing step to improve the signal-to-noise ratio and ameliorate the potential impact of registration errors.

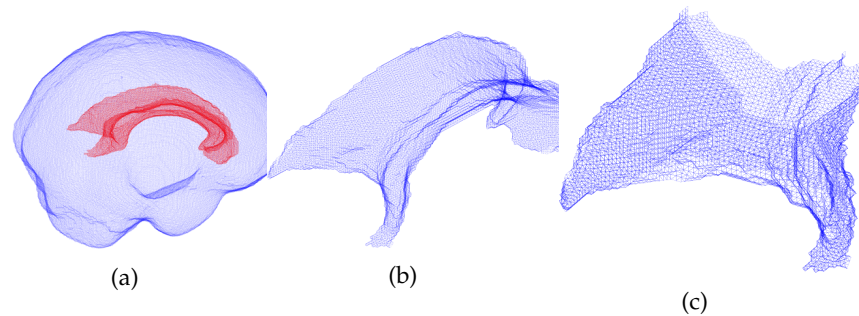


Figure 5.7: **Corpus Callosum** (a) The blue outer mesh is a 3-D view of a representation of the surface of the human brain. The red inner mesh outlines the corpus callosum. (b) A view of the corpus callosum in isolation. The corpus callosum is a thick band of nerve fibers that connects the left and right hemispheres of the brain. (c) A view of the splenium of the corpus callosum, which contains over 12,000 voxels. The splenium of the corpus callosum carries fibers that connect the bilateral temporal, parietal and occipital lobes.

this experiment is to understand the neurobiology of changes in the brain with the passing of time. We hypothesize that features that enable the solution of this problem are implicated in aging as well as cognitive decline. A related experiment was performed by Mwangi et al. (2013) using DTI data to predict the ages of subjects.

The problem of differentiating between earlier and later scans is challenging for several reasons:

1. The time period between scans is extremely short (1.5-2 years) and the subtle changes in the scans are believed to be largely unrelated to cognition.
2. All subjects are healthy, middle-aged, and do not exhibit any pathology.
3. Domain experts in neuroscience and radiology are unable to solve this problem for healthy patients better than chance.

To successfully solve this problem we must identify and rank the most temporally significant (longitudinally) and consistent (cross-sectionally) voxels in

our data. We hypothesize that these voxels correlate with other temporally sensitive data, such as cognitive test scores.

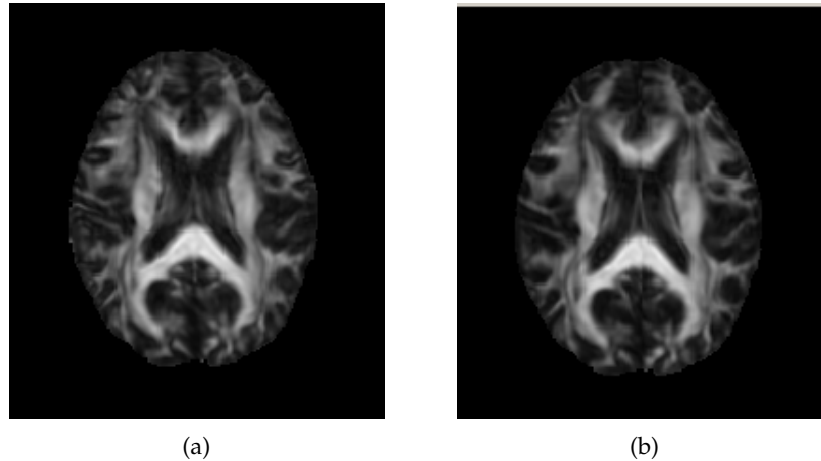


Figure 5.8: **Diffusion Tensor Imaging (DTI)**. Two axial slices from DTI scans of the same participant taken approximately two years apart. In our first task, we treated the order of the scans as unknown and proceeded to use data from 118 subjects to predict the order. The images shown here are slices from the full three-dimensional scan, which is a stack of these axial slices. The analysis is performed on the full scans.

The identification of cross-sectional consistency in FA change is of high value to neuroscientists seeking to understand the short-term evolution of white matter microstructure in subjects at high risk for AD. The goal of this experiment, therefore, is simple from a clinical viewpoint, but not quite straightforward computationally: We want to develop a classifier to distinguish between earlier and later scans of each subject by exploiting cross-sectionally consistent changes across voxels.

### 5.6.1 Identifying subsets of informative voxels

Given the large number of available voxels in our neuroimaging data, we combined longitudinal and cross-sectional data to identify those that had comparatively *large*, *consistent*, and *similar* values in all difference images corresponding



to a class. Our hypothesis is that the voxels that change similarly in all subjects (cross-sectionally) across time (longitudinally) are the ones most sensitive to temporal ordering. Towards this, we define two values, “ $Q$ ” and  $\text{CONS}$ , for each voxel. Recall that we represent any region (or collection of regions) of interest in the brain as a point set  $R = \{(v_1, w_1), (v_2, w_2), \dots, (v_N, w_N)\}$  where each  $v_i \in \mathbb{R}^3$  is a voxel position,  $w_i \in \mathbb{R}$  some value of interest at that voxel, and  $N$  is the number of voxels in the region(s).

$$Q(v_i) = \frac{\text{mean}(\text{FA}_i^1 - \text{FA}_i^2)}{\text{var}(\text{FA}_i^1 - \text{FA}_i^2)} \quad (5.1)$$

where  $\text{FA}_i^1$  is the FA value at voxel  $i$  at time 1,  $\text{FA}_i^2$  the value at time 2, and mean and variance are computed cross-sectionally over the subject population.

$\text{CONSISTENCY}$  or  $\text{CONS}$  for a voxel is defined as follows:

$$\text{Pos}_i = \frac{1}{\# \text{subjects}} \sum_{\text{subjects}} [\text{FA}_i^1 - \text{FA}_i^2 > 0] \quad (5.2)$$

$$\text{CONS}_i = \max(\text{Pos}_i, 1 - \text{Pos}_i) \quad (5.3)$$

$\text{CONSISTENCY}$  in a voxel measures the percentage of subjects who show the same sign change in that voxel from time 1 to time 2.

For a point set  $R = \{(v_1, w_1), \dots, (v_N, w_N)\}$  (such as those corresponding to a WM region in a brain scan), we define another point set  $\Delta R$  corresponding to the same region in a new image constructed from the *difference* of the two brain scans taken from the same subject and time 1 and time 2.  $\Delta R = \{(v_1, \Delta w_1), \dots, (v_N, \Delta w_N)\}$  where  $\Delta w_i$  is the change in FA at voxel  $i$  from time 1 to time 2. We set thresholds on  $Q$  and  $\text{CONS}$  to identify subsets of “informative” voxels  $\Delta \hat{R}_Q(\tau)$  and  $\Delta \hat{R}_{\text{CONS}}(\tau)$  for a region  $R$  and its corresponding  $\Delta R$  as follows:

$$\Delta\widehat{R}_Q(\tau) = \{(v_i, \Delta w_i) \mid (v_i, \Delta w_i) \in \Delta R \text{ and } Q(v_i) > \tau\} \quad (5.4)$$

$$\Delta\widehat{R}_{\text{CONS}}(\tau) = \{(v_i, \Delta w_i) \mid (v_i, \Delta w_i) \in \Delta R \text{ and } \text{CONS}(v_i) > \tau\} \quad (5.5)$$

By setting an appropriately high threshold for  $Q$  in a voxel we are able to identify a subset of a few hundred voxels (e.g. in the case of the genu of the corpus callosum, 382 voxels) from among over 12,000 that we now focus on.

For an analysis of CONS and its utility in the experiments below, see Figure 5.10. At each CONS level (x-axis), the y-axis shows the cross-validated accuracy of the before-after prediction using 50 randomly selected voxels at that CONS value. The random voxel subset was varied with every fold while cross-validating.

### 5.6.2 Experimental Setup

For each of the 118 subjects, we construct two “difference” images. The first subtracts the latter image from the earlier one (the “positive difference image”), and the second by reverses the order of subtraction (the “negative difference image”). This is done so that when given two new images from a single subject with no ordering information, we perform the subtraction in an arbitrary manner and compute which set of difference images this new difference image is more “similar” to, using the spatially aware kernels in Chapter 2.

### 5.6.3 Baseline

Since there are an equal number of positive and negative difference images, the baseline accuracy for this experiment is 50%. We applied two classification methods for comparisons with our method.

**Region-wide means.** A standard approach for characterizing images is to compare mean FA values over a whole WM region across one time point (see e.g. Magnin et al. (2009)). The classification rule “the image with the higher mean is the earlier image” achieves an accuracy rate of 53.8% on the splenium of the corpus callosum — little better than random chance. The reason for this is that not all voxels show a decrease in FA value over time; in fact some voxels show an increase. Change in one direction offsets change in the other direction, leading to a low accuracy. This insight leads us to the next baseline method.

**Sign-weighted voxel means.** The sign of  $Q$  indicates whether the voxel saw an overall increase or decrease in its value over all subjects. As in the earlier method, we compute the mean FA value within a region, but this time weighted by the sign of  $Q$  for that voxel. Applying the same classification rule yields an accuracy of 63.6% for the same region. This accuracy is obtained using all 12,729 voxels; experiments with subsets chosen according to  $Q$  or Cons achieved accuracies little better than chance.

#### 5.6.4 Classification & Accuracy

We trained a support vector machine (SVM) (Shawe-Taylor and Cristianini, 2000) with the lift kernel (Section 2.6) to classify “positive” and “negative” difference images. For comparison purposes to a widely used non-spatially sensitive kernel, we also provide results for the radial basis function (RBF) kernel. Figure 5.9 shows a visualization of the Gram matrices corresponding to these different kernels. Accuracy for each region was determined with 10-fold cross validation; experiments for all three kernels were run for each fold. Cons and voxel selection were re-calculated per fold in order to prevent any information leakage from the test set during training. The 10-fold cross-validation accuracies in predicting “before” scans from “after” scans (i.e. “positive” difference images

Region	Label
Corpus Callosum (whole)	101
Corpus Callosum (splenium)	5
Corpus Callosum (genu)	3
Cingulum bundle (R & L)	35
Superior longitudinal fasciculus (R & L)	41
Uncinate fasciculus (R & L)	45
Fornix (column, body, and cres)	6

Table 5.1: **Brain regions.** This table lists the regions in the brain we conducted analysis on. Each region is assigned a label which it is referred to by in the results in Table 5.2.

Region	Mode	$ \Delta\widehat{R}_{\text{CONS}}(\tau) $	RBF	Lift
101	FA	1512.7 voxels	79.7%	<b>91.5%</b>
	MD	4803.5 voxels	75.4%	<b>87.3%</b>
5	FA	425.7 voxels	84.7%	<b>93.2%</b>
	MD	2282.8 voxels	69.5%	<b>81.4%</b>
3	FA	170.6 voxels	80.9%	<b>88.1%</b>
	MD	948.2 voxels	74.6%	<b>86.4%</b>
35	FA	1466.2 voxels	84.7%	<b>91.5%</b>
	MD	1090.2 voxels	64.4%	<b>85.6%</b>
41	FA	2032.5 voxels	83.1%	<b>92.4%</b>
	MD	505.6 voxels	63.6%	<b>81.4%</b>
45	FA	124.8 voxels	73.8%	<b>86.4%</b>
	MD	127.3 voxels	64.4%	<b>79.7%</b>
6	FA	19.3 voxels	66.1%	<b>76.3%</b>
	MD	203.2 voxels	71.2%	<b>84.7%</b>

Table 5.2: **Before-After Results.** Classification results using a linear SVM with different kernels for predicting the before image from the later image using seven different WM regions (listed in Table 5.1). The kernels used are radial basis function (RBF) and lift from Chapter 2.  $\tau$  was fixed for all experiments at 0.7 for FA and 0.65 for MD, and the number of voxels reported is the mean cardinality of the set  $|\Delta\widehat{R}_{\text{CONS}}|$  across the different folds in each experiment.

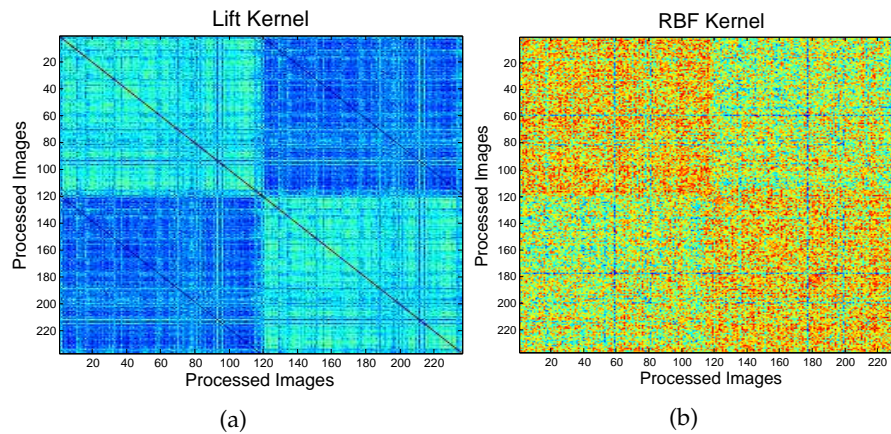


Figure 5.9: **Gram Matrix.** The panels in this image show Gram matrices for the first experiment represented as images. A Gram matrix in this context is composed of kernel values between all instances in the data set. Our data set contains two scans from each of 118 subjects, leading to 236 positive and negative difference images. The Gram matrix therefore contains similarity values for every pair of difference images in this data set. The first 118 entries along the  $x$  and  $y$  axes correspond to the positive difference images, and thereafter the negative difference images. Panels (a) and (b) correspond respectively to Gram matrices constructed using the lift and RBF kernels. As the colors in the image in panel (a) show, the positive images are similar to one another, as are the negative images to one another, but the positive and negative images are not similar to each other (reflected by the darker color). These differences are exploited by a linear SVM classifier to yield the accuracy results in Table 5.2.

from “negative” difference images) is shown for different WM regions in Table 5.2. As the table shows, approximately 425 voxels are sufficient to achieve a classification accuracy of 93%. Figure 5.10 shows a further analysis of the variation of classification accuracy with random subsets of 50 voxels chosen at different levels of Cons.

### 5.6.5 Regions of Consistent Cross-Sectional Change

The hypothesis of this experiment was that there exist voxels that undergo consistent and similar changes across subjects, and identification of these voxels would help in characterizing cross-sectional FA change. The experimental

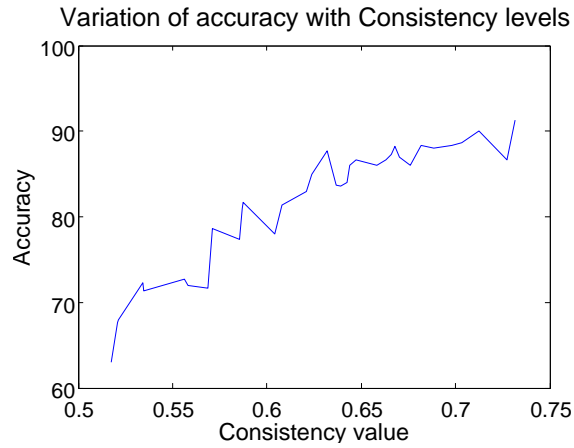


Figure 5.10: **Classification Accuracy vs. CONS.** This figure illustrates the utility of the CONS statistic. The plot shows the variation of scan order prediction accuracy (Section 5.6) using 50 voxels selected at different thresholds of CONS. As the graph shows, prediction accuracy increases when using voxels with higher values of CONS.

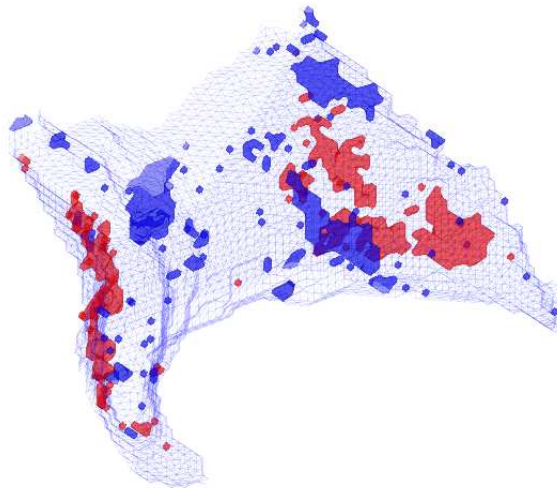


Figure 5.11: **Corpus Callosum.** This figure illustrates the portions of the splenium of the corpus callosum that contain voxels with high CONS ( $\geq 0.65$ ) value. Red voxels indicate a consistent increase in FA value across subjects, while blue represents a consistent decrease.

results above show that this hypothesis holds. In Figures 5.11 and 5.12 we pinpoint those voxels and visualize them in the context of the WM regions they belong to. Voxels can be distinguished based on whether they show an upward trend in FA value or a downward trend. Figure 5.11 shows that voxels tend to be spatially proximal to other voxels of the same type. We note that this naturally-occurring “clustering” of nearby voxels with similar trends is readily apparent **even when no smoothing is applied to the data**. Study of these regions and the trends within them will be useful in understanding patterns of age-related change in FA. Of particular interest are the correlations between FA changes, demyelination, and cognitive impairment, as discussed in Section 5.9.

## 5.7 APOE Status Classification

We now apply the framework developed above to a different problem: is there a difference in the way that WM changes in subjects with different APOE genotypes? Prior studies have established that subjects with the  $\epsilon 4$  allele are at higher risk for developing AD Ashford (2004). We attempt to answer this question by predicting the APOE  $\epsilon 4$  status (i.e., the presence or absence of this allele) based on the changes in FA values. This experiment is similar to the previous one. Rather than have two sets of positive and negative difference images, we take just one (positive difference images) and group them by the APOE  $\epsilon 4$  status of the subjects they correspond to. We transform these images into point sets and apply a slightly different voxel selection scheme than before: within each group we identify the voxels that exhibit increases and decreases most consistently, and take the union across both groups:

$$\Delta \widehat{R}_{\text{CONS}}(\tau) = \Delta_{\text{ApoE -ve}} \widehat{R}_{\text{CONS}}(\tau) \cup \Delta_{\text{ApoE +ve}} \widehat{R}_{\text{CONS}}(\tau)$$

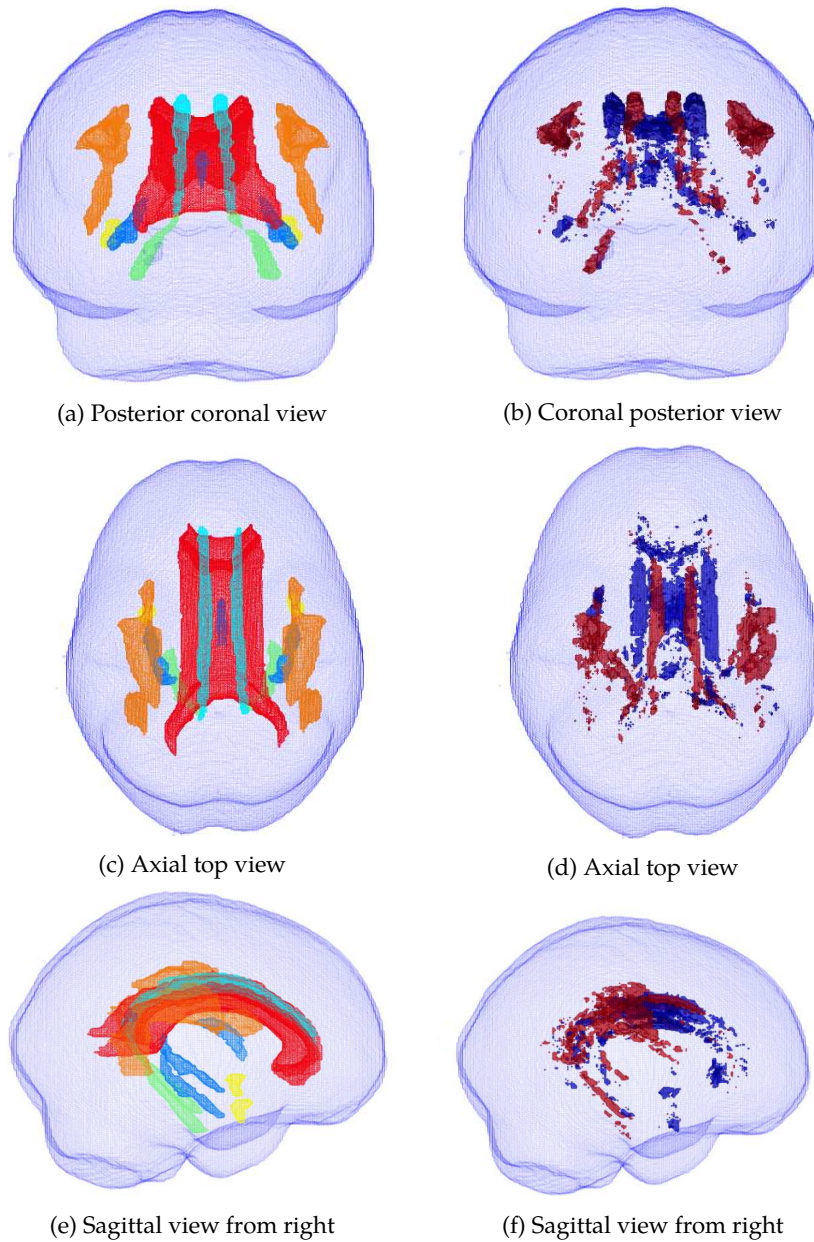


Figure 5.12: **FA trends in the brain.** In the left column are views from three directions of the corpus callosum (red), fornix (blue), cingulum (cyan), cingulum projecting to hippocampus (green), uncinate (yellow), and the superior longitudinal fasciculus (orange). The figures on the right show voxels in these regions exhibiting large and consistent FA increase (red) and FA decrease (blue).



We used the lift kernel in conjunction with an SVM to differentiate between these two classes of point sets. The baseline accuracy for this experiment is 66.95%, since 79 out of 118 subjects are APOE  $\epsilon 4$  negative. The best cross-validated accuracy of 76% was obtained using the whole body of the corpus callosum, with  $\tau = 0.63$ . The non-spatial RBF kernel was not able to achieve more than baseline accuracy. The regions of the corpus callosum we found most predictive of  $\epsilon 4$  status are shown in Figure 5.13 and are in fibers connecting to the premotor and supplementary motor areas as well as the temporal lobe.

## 5.8 Predicting Cognitive Changes

We would like to model changes in subjects' neuropsychological test scores using FA differences observed over time. Even employing the  $Q$  and  $CONS$  scores defined above to prune the space of voxels, it remains the case that  $p > N$ . Fitting multivariate linear models in this case cannot be done without constraints. Common approaches that limit model exploration including step-

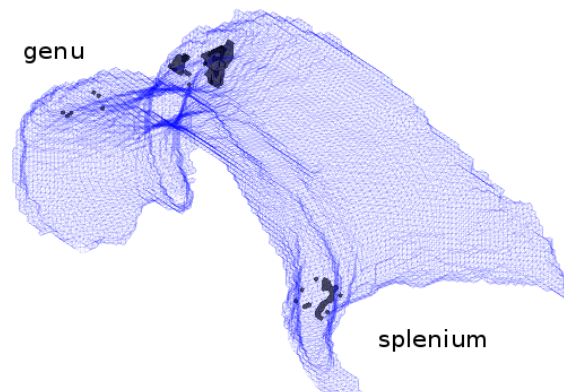


Figure 5.13: **Corpus Callosum: APOE-predictive regions.** This figure shows a view of the corpus callosum from behind. The genu (front) and splenium (back) are labeled. The shaded regions correspond to voxels that are highly predictive of APOE status. In subjects that have at least one  $\epsilon 4$  allele, these regions showed an FA increase. In subjects without any  $\epsilon 4$  allele, these regions showed an FA decrease.

wise, best-subset, lasso, and ridge regression. The latter two are often combined via elastic net regularization. There are many ways to validate these models including: using adjusted  $R^2$  values, cross-validation, hold out sets, and checking the distributions of the residuals. However, with a limited number of samples  $N$ , evaluating the assessments themselves is difficult. The data are difficult to work with, as none of the differences between earlier and later test scores is statistically significant according to paired t-tests adjusted for inequality of variances. Scatterplots of earlier vs. later test scores fit lines of slope 1 with relatively high adjusted  $R^2$  (see Figure 5.14 for a plot of these scores). In these cases, even null models perform well.

While most of the study's cognitive tests had negative adjusted  $R^2$  values when fit to linear models using the high  $Q$  voxels from Section 5.6, the *Speed and Flexibility* score (Section 5.5.3) yielded an adjusted  $R^2$  of almost 0.4. ANOVA analysis revealed wide levels of variability within the model, suggesting that while  $Q$  is useful for "screening" informative voxels, it may not be sufficient for model feature selection.

In cases where  $p \gg N$ , many regression methods become implausibly slow, due to their computational complexity as a function of  $p$ . Even stepwise techniques become impractical and yield suboptimal models. The problem of model selection for large  $p$  is further compounded by the need for extensive bootstrapping and cross-validation to guard against overfitting given the relative paucity of samples  $N$ , especially in determining the  $\lambda$  shrinkage parameter for regularization-based methods.

To better manage the need for constrained variable selection with wide data, we used the coordinate descent approach for lasso and ridge described in Friedman et al. (2010). To make the results easier to interpret, we modified our approach to perform logistic regression on the *signs* of the test score changes, viewed as binomial distributions. Doing so normalizes the error penalty and



Figure 5.14: *Speed and Flexibility*. (a) A plot of values of the *Speed and Flexibility* factor for subjects in our study. The score at Time 2 is plotted against score at Time 1; each point in the scatter plot represents one subject. The red line (slope = 0.76) shows the best linear fit for this data. As the plot shows, not all subjects perform worse at Time 2; in fact, a significant number show improvement. Panel (b) shows a histogram of the *differences* in this score for the subjects in our study. While there is no unit for this factor score, the scale of the scores can be seen in panel (a). These figures show that roughly similar numbers of people had lower and higher scores respectively at the second time point.

allows us to pose a well-defined problem: can changes in neuroimaging data predict whether a subject's score for some neuropsychological test has increased or decreased? One might suppose that cognitive abilities uniformly deteriorate monotonically with age. However, evidence does not bear this out, as discussed in Section 5.9.

Initially, the chance of finding a successful solution to this problem seemed implausible. Our output variable is the sign of a small difference that appears to fluctuate around zero at random. However, lasso logistic regression via coordinate descent run 100 times with 10-fold cross validation achieved a classification accuracy of 70% with shrinkage parameter  $\lambda = .011$ , which corresponds to the  $\lambda$  within one standard error of the minimum. Results for this and other

Method	Parameters	Accuracy
Lasso logistic regression Friedman et al. (2010)	$\lambda = .011$	<b>70%</b>
SVM, Lifted kernel	$2D = 500, C = 1$	58%
SVM, Gaussian kernel	$\sigma = 1, C = 1$	57%
Baseline Random Guessing		54%

Table 5.3: Classification results for predicting *Speed and Flexibility* from voxels

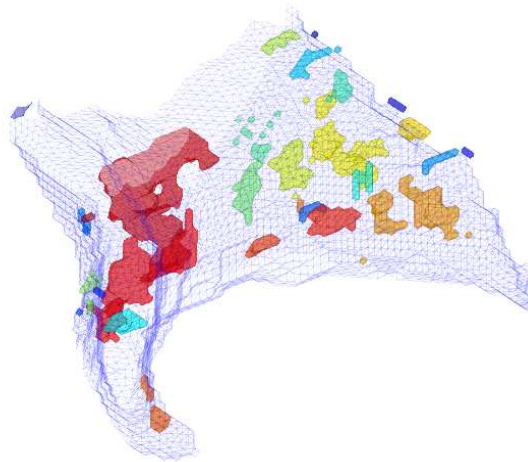


Figure 5.15: **Corpus Callosum**. A view of voxels in the splenium of the corpus callosum clustered by  $Q$  values. Colors correspond to different clusters.

methods are shown in Table 5.3. No significant improvement was seen for other parameters on competing approaches.

These results are quite surprising. Although achieving 70% accuracy seems a modest achievement, consider that this prediction is made using voxel-based neuroimaging data selected because they were able to accurately answer our initial "Which image came first?" question. Within their own representation, the outcome data do not appear separable. Viewing them from the neuroimaging perspective however, we are able to classify them.

### 5.8.1 Clustering

In general, we prefer as few explanatory variables in a model as possible. Wide linear models always raise the specter of overfitting and are notoriously difficult

Method	Parameters	Accuracy
Ridge logistic regression(Friedman et al., 2010)	$\lambda = .013$	<b>75%</b>
SVM, Lifted kernel	$2D = 500, C = 1$	55.7%
SVM, Gaussian kernel	$\sigma = 1, C = 1$	58.5%
Baseline Random Guessing		54%

Table 5.4: Classification results for predicting *Speed and Flexibility* from 30 clusters of voxels

to interpret, particularly when constructed with lasso. For example, one cannot determine the significance of variables by the magnitude of their coefficients. Following the spatial point set approach in Chapter 2, we cluster the voxels based on spatial proximity and their  $Q$  values. Simple linkage-based clustering connects voxels with their neighbors if their  $Q$  values are within  $\rho$  percent of each other. We typically take  $\rho = 15$  and specify the maximum number of desired clusters as 30. Emerging from the clustering was the observation that spatially adjacent voxels are likely to have similar  $Q$  values. The regions corresponding to clustered voxels are shown in Figure 5.15.

Because the clusters are internally consistent with respect to  $Q$  values, we used their mean FA values in a ridge logistic regression analysis to predict the sign of the change in the *Speed and Flexibility* score. Because  $p = 30$  here, which is the number of regions, we are no longer dealing with wide data, alleviating many of the concerns that they raise. While one might imagine the clustering process is lossy, the clusters are better predictors than the voxels used in the previous model. Ridge logistic regression via coordinate descent run 100 times with 10-fold cross validation achieved a classification accuracy of 75% for shrinking parameter  $\lambda = 0.13$ , as chosen above. Results for this and other methods are shown in Table 5.4. No significant improvement was seen for other parameters on competing approaches.

## 5.9 Discussion

In this chapter we presented a new approach for longitudinal analysis of neuroimaging data. From a computational perspective, our approach relies on the spatial nature of the data both for defining similarity and for clustering voxels based on their perceived quality or  $Q$  value. We demonstrated this kernel can be used to reliably classify longitudinal neuroimages based on small changes in their white matter structure. This task cannot be solved by human experts. We then used the voxels that enabled this classification to predict changes in the significant cognitive factor of *Speed and Flexibility*, a cognitive function known to be tightly associated with white matter health. While a relationship between speed based cognitive tests and white matter microstructure has been *qualitatively* examined in cross-sectional studies, this is the first work to determine that change in FA over two years can predict change in cognitive function in healthy adults.

From a neuroscience perspective, this work found that over time, certain portions of the splenium show a decrease in FA from the first time point to the second time point approximately 2 years later. Given what is known about aging in general, this was expected. More unexpected were the portions of white matter tracts that showed an increase in FA from time 1 to time 2 (the red regions of Figure 5.11). The splenium of the corpus callosum carries fibers that connect the bilateral temporal, parietal and occipital lobes. While occipital brain regions do not show high levels of change with age, the temporal and to a lesser extent parietal cortices do change with age. Studies on white matter in the frontal cortex of rhesus macaques indicate that age is associated with loss of nerve fibers, but that this degenerative process may be accompanied by continued myelination (Bowley et al., 2010). It is possible that changes occurring over time include both loss of fibers and regenerative myelination. However, in the

absence of post-mortem pathological findings, this interpretation is speculative. Another possibility is that our finding reflects the continued myelination that occurs in aging. Visuo-motor skill training in young adults has been shown to increase FA over time (Scholz et al., 2009), and Lövdén et al. (2010) have shown that experience-dependent changes in FA occur even in older age.

Thus, it is possible that we are capturing patterns of white matter change that reflect continued plasticity in the brain. This is underscored by the tight relationship found with the *Speed and Flexibility* factor score. Because speed of neural conduction relies on intact myelin, it is not surprising that cognitive speed of processing is linked with white matter health.

### 5.9.1 Clinical Implications

These results have the potential to inform several promising directions of research in Alzheimer's disease. The white matter structural patterns that we found in patients that are at high risk of developing MCI and AD can be used as part of an early detection test that looks for distinct imaging markers in new subjects' brains. Early detection can help in developing more effective treatment protocols for subjects. Results from this research can also help clinicians and diagnosticians in making effective treatment decisions early on for patients whose WM structure is found to correspond to the patterns detected by our research.

Our computational techniques are capable of identifying structural patterns that can inform inclusion and exclusion criteria for subjects in future trials. A full battery of imaging and tests can cost up to thousands of dollars, so ascertaining the relevance of a subject to further studies can be cost-effective. For example, in a research project focused on studying the mechanism of transition from MCI to AD, an exclusionary criteria could be to exclude those

subjects who show WM microstructure patterns consistent with *not* going on to develop AD.

The diagnostic criteria today for AD are solely based on cognition and memory tests. As such, they are only useful very late in the development of AD, after it has already manifested clearly. Structurally, the only existing test for AD is a post-mortem study of the brain. Our results can lead to a diagnostic criteria that is applicable earlier on in the process when changes in the WM structure of the brain that are known to eventually lead to AD are visible. Finally, the results of this research can help in method development for neuroscience studies.



## 6 CHARACTERISTIC NUMBERS: REVEALING NEW PROPERTIES OF CLASSICAL PROBABILITY DISTRIBUTIONS

---

In Chapter 2, we discussed a ratio of two transportation distances that measures spatial overlap between two point sets. In this chapter we apply a similar idea and show how it leads to a new and surprising property of classical probability distributions that we call “characteristic numbers.” Our result enables the characterization of an entire family of distributions by a single number that, in many cases, is independent of the distribution parameters as well as the dimensionality of the distribution. Characteristic numbers are shown for three common distributions in Table 6.1. This characterization extends neatly to discrete samples from those distributions as well; the quantity computed for a sample is, in the limit, the same as the characteristic number for the source distribution. This is a surprising result, especially noteworthy for its independence of distribution parameters for families such as the uniform, normal, exponential distributions. In these cases the characteristic number depends only on the family of the distribution and the choice of a ground

Distribution	Ground distance $d_\Omega$	
	$\ell_1$	$\ell_2^2$
<b>Normal</b>	$\sqrt{2}$	<b>2</b>
<b>Uniform</b>	$3/2$	<b>2</b>
<b>Exponential</b>	$2 \log(2)$	$\pi^2/6$

Table 6.1: **Characteristic numbers.** Analytically calculated characteristic numbers for three common distributions using two ground distance functions. These numbers are independent of any parameters of the distribution.

distance function.<sup>1</sup> As a consequence of this property, characteristic numbers enable a powerful new goodness-of-fit test discussed in the next chapter.

Characteristic numbers are grounded in the overall spatial properties of a distribution, rather than a summary statistic describing one aspect of it, e.g., its mean, moments, or tail. Recall that the Kantorovich-Wasserstein distance  $d_{KW}$  (see Section 2.4.1) is a metric from optimization theory providing the minimum expected distance between two probability distributions. We define below a variant of  $d_{KW}$  that is used to create a *measure between a distribution and itself*. A key strength of this approach is its sensitivity to the pairwise relationships between *all* points in a sample. In the following sections we define this quantity, show how to compute it for distributions and point samples, analyze it empirically for five common families of distributions, and derive a goodness-of-fit test based on it.

## 6.1 Characteristic Numbers for Distributions

Characteristic numbers are based upon a variant to the solution to the Transportation Problem by Kantorovich (2006), through what is commonly known as the Kantorovich-Wasserstein metric ( $d_{KW}$ ) (Deza and Deza, 2009). Recall from Section 2.4.1 that  $d_{KW}$  computes the minimum amount of work to cooperatively transport a distributed source mass to a sink distribution. It can equivalently be seen as the minimum expected  $\ell_2$ -distance between two probability distributions (Levina and Bickel, 2001).  $d_{NT}$  is defined the amount of work in the same scenario with no cooperation, where each source distributes its mass to all sinks independently.

In this section, we define a variant of  $d_{KW}$ . We alternate between discrete and continuous versions of these measures; the discrete formulations are used for

---

<sup>1</sup>see Chapter 2 for a discussion of ground distance functions

determining the values for finite samples, whereas the continuous definitions allow us to prove properties of entire distributions.

### 6.1.1 Anti-Transportation Distance

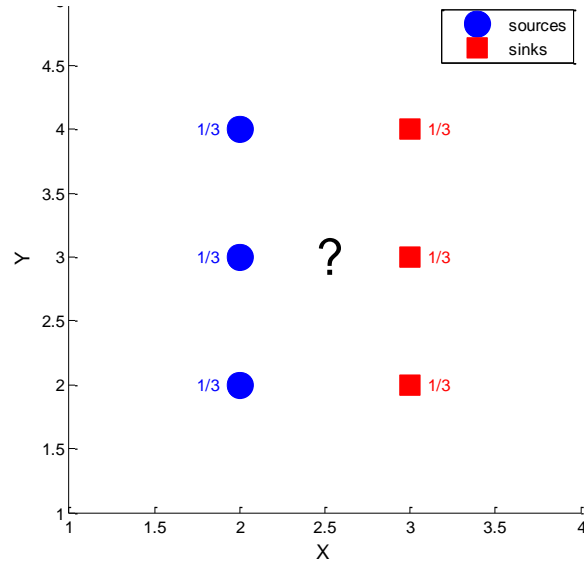
In contrast with  $d_{KW}$ , we define the *anti-transportation distance* ( $d_{AT}$ ) as the *least* efficient solution to the Transportation Problem, as shown in Figure 6.1 (c). Let  $\mu$  and  $\nu$  be two probability distributions on a metric space  $\Omega$  with associated distance metric  $d_\Omega$ .  $d_{AT}$  is defined as:

$$d_{AT}(\mu, \nu; d_\Omega) = \sup_J \{E(d_\Omega(x, y)) : \mathcal{L}(x) = \mu, \mathcal{L}(y) = \nu\} \quad (6.1)$$

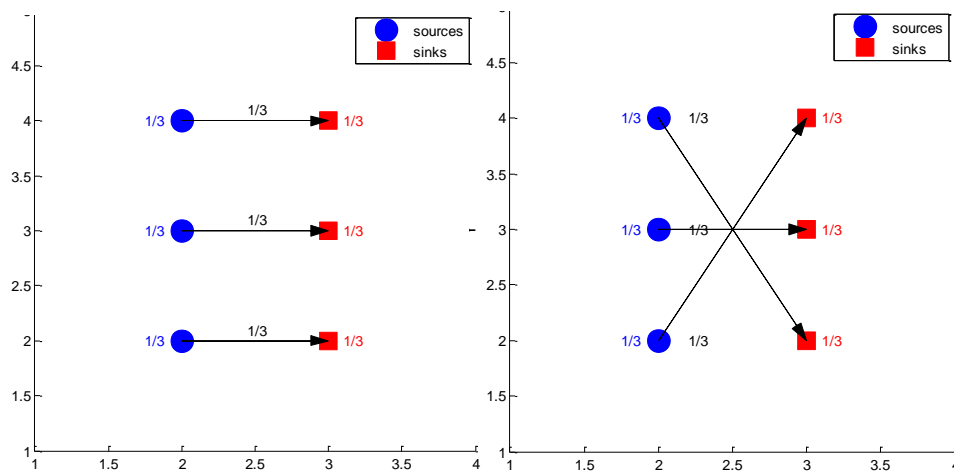
where the marginals  $\mathcal{L}$  are  $\mu$  and  $\nu$  respectively, and the supremum – in contrast to the infimum in  $d_{KW}$  – is taken over all joint distributions  $J$  on  $\mu$  and  $\nu$  (which are in  $\Omega \times \Omega$ ). Here,  $d_\Omega$  is the distance metric for  $\Omega$  and  $d_\Omega(x, y)$  represents the cost to move a unit amount of mass from  $x$  to  $y$ .

For the definition of  $d_{AT}$  in the discrete case, let  $A = \{(a_1, p_1), \dots, (a_m, p_m)\}$  and  $B = \{(b_1, q_1), \dots, (b_n, q_n)\}$  be two weighted point sets. The discrete version of  $d_{AT}$  follows the formulation in Section 2.4.1 and transforms into a maximization problem as follows. Treating  $A$  and  $B$  as random variables taking values  $\{a_i\}$  and  $\{b_j\}$  with probabilities  $\{p_i\}$  and  $\{q_j\}$  respectively,  $d_{AT}$  is obtained by maximizing the expected distance between  $A$  and  $B$  over all joint distributions  $F = (f_{ij})$  of  $A$  and  $B$ :

$$E_F \|A - B\| = \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_\Omega(a_i, b_j)$$



(a) The source (blue) and sink (red) masses



(b) The optimal solution, minimizing the amount of work required to transport the masses.

(c) The **most inefficient** solution; each source finds the sink the maximizes the total work performed.

Figure 6.1: **A view of the Transportation Problem.** The goal is to transport the mass in the sources to the sinks. Different solutions involve differing amounts of *work* where work is  $\sum mass \times distance$  Shown here are the most optimal and suboptimal ways of transportation. This figure is adapted from Coen (2010).

where  $F$  is subject to:

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n \quad (6.2)$$

$$\sum_{j=1}^n f_{ij} = p_i, 1 \leq i \leq m \quad (6.3)$$

$$\sum_{i=1}^m f_{ij} = q_j, 1 \leq j \leq n \quad (6.4)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \sum_{i=1}^m p_i = \sum_{j=1}^n q_j = 1 \quad (6.5)$$

For both the discrete and continuous cases, we may compute  $d_{AT}$  by observing that it is the “opposite” of the transportation problem. Thus, we could modify the linear program above to maximize rather than minimize the flow. Similarly, we could modify the infimum to be a supremum in the continuous definition of  $d_{KW}$ .

A more elegant approach is simply to note that a supremum over a set can be computed via the infimum over a new set consisting of additive inverses of the original quantities. The supremum of the original set is then the additive inverse of the infimum over the modified set. In other words, we can define

$$d_{AT}(\mu, \nu; d_{\Omega}) = -d_{KW}(\mu, \nu; -d_{\Omega}).$$

Minimization is performed over the negative distances and then negated to obtain a positive value.

### 6.1.2 Anti-Similarity distance

The optimization measure we will use to derive characteristic numbers is called **anti-similarity distance**, defined equivalently to similarity distance ( $d_{SIM}$ ) but using  $d_{AT}$  in the numerator rather than  $d_{KW}$ . Namely, the anti-similarity distance between  $A$  and  $B$  is defined as:

$$d_{AS}(A, B; d_{\Omega}) = \frac{d_{AT}(A, B; d_{\Omega})}{d_{NT}(A, B; d_{\Omega})} \quad (6.6)$$

Likewise it can be defined between continuous probability distributions by replacing  $A$  and  $B$  above with distributions  $\mu$  and  $\nu$  and using the continuous counterparts of  $d_{AT}$  and  $d_{NT}$ . Note that when specifying  $d_{AS}$ , we often leave out  $d_{\Omega}$  when it is clear from context. In the remainder of this chapter for the purpose of computing  $d_{AS}$  we will assume that each point in  $A$  and  $B$  is weighted identically. The anti-similarity distance,  $d_{AS}$ , is a measure of the *inefficiency gained* when moving distributions in the worst way possible as opposed to naively. Its most important property is derived when computing the anti-similarity distance of a sample or distribution to *itself*, namely, the value of  $d_{AS}(\mu, \mu; d_{\Omega})$ . We refer to this as the **self-anti-similarity distance** or the *self- $d_{AS}$*  of  $\mu$ . Whereas  $d_{sim}(\mu, \mu) = 0$  by definition, the value of  $d_{AS}(\mu, \mu)$  is highly dependent both on  $\mu$  and the distance function  $d_{\Omega}$ . It provides an intrinsic measure of how expensive it is to *move something onto itself* in the worst possible way. It is this number that we call the **characteristic number** of a distribution with respect to some distance function  $d_{\Omega}$ .

## 6.2 Computing $d_{AT}$

To compute  $d_{AT}$  efficiently we first formally define the notion of *worst* in the anti-transportation problem; namely, how do you *move* a distribution onto itself in the most expensive way possible as defined in Section 2.4.1? We derive an analytic solution to this problem for a general probability distribution using the following two theorems. The results below require  $d_{\Omega}$  to be *quasi-antitone*:

**Definition 6.1.** *Quasi-antitone:*

A function  $d(x, y)$  on  $\mathbb{R} \times \mathbb{R}$  is called quasi-antitone if for all  $x' \geq x$  and  $y' \geq y$

$$d(x', y') + d(x, y) \leq d(x, y') + d(x', y) \quad (6.7)$$

Examples of quasi-antitone functions include  $|x - y|^p$  for  $p \geq 1$ ,  $\max(x, y)$ , and  $f(x - y)$  for convex and continuous  $f$ .

The first theorem is taken from Cambanis et al. (1976) and stated here without proof:

**Theorem 6.2** (Upper bound for expected distance between distributions). *Let  $\mu$  and  $\nu$  be two probability distributions with (cumulative) distribution functions  $F$  and  $G$  respectively. Let  $H$  be a joint distribution function between them, and  $d_\Omega(x, y)$  a quasi-antitone, symmetric, and right continuous function. Then the supremum of the expected value of  $E_H d_\Omega(\mu, \nu)$  over all joint distribution functions, if it exists, is*

$$\sup_H E_H d_\Omega(\mu, \nu) = \int_0^1 d_\Omega(F^{-1}(u), G^{-1}(1 - u)) du$$

**Theorem 6.3** (Self-anti-transportation distance for a distribution). *Let  $\mu$  be a probability distribution defined on  $\mathbb{R}$  and  $f_\mu$  and  $F$  its density function and distribution function respectively. Let  $d_\Omega$  be a symmetric, quasi-antitone, and continuous function. If  $F$  is differentiable and the quantity below exists, the self-anti-transportation distance of  $\mu$  may be computed as*

$$d_{AT}(\mu, \mu; d_\Omega) = \int_{\mathbb{R}} f_\mu(x) d_\Omega(x, M(x)) dx$$

where  $M(x) = F^{-1}(1 - F(x))$ .

*Proof.* The result follows immediately from the definition of  $d_{AT}$ , and the substitution  $u = F(x)$  in the result from Theorem 6.2.  $\square$

Note that we no longer require  $d_\Omega$  to be a metric. This formulation captures the notion that the worst transportation distance between  $\mu$  and itself can be achieved when each  $x \in \mathbb{R}$  transports all of its mass to its *match*  $M(x)$ , which is its counterpart in the other half of the distribution. If the extremes of the domain of  $\mu$  are  $a$  and  $b$  (possibly infinity), then the probability mass contained between  $x$  and  $a$  is equal to the mass contained between  $M(x)$  and  $b$ . The condition that  $d_\Omega$  is quasi-antitone simply ensures that there is no better global match for a probability mass  $f_\mu(x)dx$  than  $f_\mu(M(x))dx$ . In the specific case of a continuous symmetric probability distribution  $\mu$ , we have  $M(x) = 2m - x$ . That is,  $M(x)$  is the point reflected about and equidistant from the mean  $m$ .

### 6.2.1 Computational Complexity

We now discuss the computational complexity of the *discrete* versions of  $d_{AT}$  and  $d_{AS}$ . The complexity of computing self-anti-similarity distance for discrete point samples is identical to that of SIM. This is because, as mentioned earlier,  $d_{AT}$  can be computed by inverting<sup>2</sup> the ground distance function, computing  $d_{KW}$ , and inverting the result. In the discussion below, complexity is calculated assuming an input point set  $P = \{a_1, a_2, \dots, a_n\}$  of size  $n$  that is unweighted. The complexity of the  $d_{NT}$  step in the general case is  $O(n^2)$ , as before. The final theoretical complexity of  $d_{AS}$  in the general case is therefore  $O(n^3)$ , dominated by the complexity of  $d_{KW}$ .

In the univariate case however, there is significant room for improvement by building upon the idea of a “match” as discussed above and by using algebraic tricks to simplify the computation of  $d_{NT}$ . Once a point set is sorted, each point can be matched with its match, and this assignment will lead to the worst transportation distance possible, without the need to perform any optimization. This is proven below. Recall that the computation of anti-transportation distance

---

<sup>2</sup>i.e. taking the additive inverse



between a point set and itself involves finding an assignment of points to one another such that the total summed distance between them is maximized.

**Theorem 6.4** (Self-anti-transportation distance for a univariate sample). *Let  $P$  be a point set as defined above and  $d$  a quasi-antitone distance function. Assume without loss of generality that  $P$  is sorted in ascending order. The anti-transportation distance between  $P$  and itself is*

$$\frac{1}{n} \sum_{i=1}^n d(a_i, a_{n-i+1}) \quad (6.8)$$

*In other words, the anti-transportation distance is achieved when each point  $a_i$  is matched to its “opposite point”  $a_{n-i+1}$ .*

*Proof.* Suppose this were not the case; suppose that another assignment  $C$  led to the highest transportation distance  $W$ . There must then exist at least one pair of points in  $C$  whose members are not “opposite points” of each other. For notational convenience we will refer to the coordinate of the point  $a_i$  using  $a_i$  itself. Let  $a_j$  be the first point in  $P$  such that all point masses  $a_k$  where  $k < j$  are transported to their opposite point  $a_{n-k+1}$  (and vice versa) but  $a_j$  is transported elsewhere. Suppose  $a_j$  is instead transported to  $a_u$  and  $a_{p-j+1}$  is transported to  $a_v$ . We note that  $u$  must be less than  $n - j + 1$  and  $v$  must be greater than  $j$ . Construct a new transportation assignment  $C'$  such that  $a_j$  and  $a_{n-j+1}$  exchange masses, as do  $a_u$  and  $a_v$  and let  $W'$  be the transportation distance corresponding to  $C'$ . Then we have

$$W' = W - d(a_j, a_u) - d(a_{n-j+1}, a_v) + d(a_j, a_{n-j+1}) + d(a_u, a_v). \quad (6.9)$$

From the quasi-antitone property of  $d$  we have

$$d(a_j, a_{n-j+1}) + d(a_u, a_v) \geq d(a_j, a_u) + d(a_{n-j+1}, a_v). \quad (6.10)$$

From the above and Equation 6.9 we obtain  $W' \geq W$ . It cannot be the case that  $W' > W$  because  $W$  is the highest transportation distance possible. Therefore the assignment  $C'$  leads to a distance no less than the highest distance. Applying a similar argument to all other points not matched with their respective opposite points, we conclude that the assignment where each point is matched to its opposite point leads to the highest transportation distance.  $\square$

Using the result from Theorem 6.4 **the computational complexity of  $d_{AS}$  for univariate samples reduces to  $O(n^2)$**  (dominated by the complexity of  $d_{NT}$ ).

A second optimization involves the computation of  $d_{NT}$  for both the  $\ell_1$  and  $\ell_2^2$  ground distances. For  $\ell_1$  the complexity of  $d_{NT}$  can be reduced to  $O(n)$  for one dimensional point sets from  $O(n^2)$  in the general case as follows. As above, we will assume  $A$  is sorted in ascending order. Applying the definition of  $d_{NT}$  from Equation 2.10 to the point set  $A$  and itself, we have:

$$\begin{aligned}
 d_{NT}(A, A) &= \sum_{i=1}^n \sum_{j=1}^n \frac{1}{n^2} |a_i - a_j| \text{ (since each weight } p_i = \frac{1}{n} \text{)} \\
 &= \frac{2}{n^2} \sum_{j < i} (a_i - a_j) \\
 &= \frac{2}{n^2} \sum_{j < i} \sum_{k=j}^{i-1} (a_{k+1} - a_k) \\
 &= \frac{2}{n^2} \sum_{k=1}^{n-1} k(n-k-1)(a_{k+1} - a_k) \tag{6.11}
 \end{aligned}$$

This sum can be computed in  $O(n)$  time.

In the case of the  $\ell_2^2$  ground distance we similarly apply the definition of  $d_{\text{NT}}$  to the point set  $A$  and itself to obtain:

$$\begin{aligned}
 d_{\text{NT}}(A, A) &= \sum_{i=1}^n \sum_{j=1}^n \frac{1}{n^2} (a_i - a_j)^2 \text{ (since each weight } p_i = \frac{1}{n}\text{)} \\
 &= \frac{1}{n^2} \left[ 2n \sum_{i=1}^n a_i^2 - 2 \sum_{i=1}^n \sum_{j=1}^n a_i a_j \right] \\
 &= \frac{2}{n^2} \left[ n \sum_{i=1}^n a_i^2 - \left( \sum_{i=1}^n a_i \right)^2 \right] \tag{6.12}
 \end{aligned}$$

Each term in square brackets in Equation 6.12 can be computed in  $O(n)$  time.

For both  $\ell_1$  and  $\ell_2^2$  therefore, the computational complexity of  $d_{\text{AS}}$  in one dimension is  $O(n \log(n))$ , dominated by the complexity of the sorting step in the computation of  $d_{\text{AT}}$ .

### 6.3 Analytic Results for Self- $d_{\text{AS}}$

We now examine values of  $d_{\text{AS}}(\mu, \mu)$  for several classical probability distributions and distance measures using Theorem 6.3. These values are presented above in Table 6.1. Characteristic numbers are computed for continuous distributions using the continuous form of  $d_{\text{AS}}$ ; for samples from these distributions we will use its discrete form. Our hypothesis – tested in the following section – is that  $d_{\text{AS}}$  for increasingly larger samples will approach the characteristic number for its source distribution. In the limit, of course, the density profiles will be identical, leading to the same values for  $d_{\text{AS}}$ .

Note that the notion of a *characteristic number* is modular in that it depends upon a given distance function, of which we will study two in detail,  $\ell_1$  and the non-metric  $\ell_2^2$ . Derivations of characteristic numbers for univariate uniform, exponential, and normal distributions using  $\ell_1$  ground distance follow below, with those for  $\ell_2^2$  in Appendix A.

Characteristic numbers can also be computed for multivariate distributions using the same definitions. Note that in the one-dimensional case the  $\ell_1$  and  $\ell_2$  ground distances are identical; the distinction is relevant in the case of multi-dimensional distributions. Importantly, it can be shown that for  $\ell_1$  and certain other choices of ground distance functions these **characteristic numbers are independent of the dimensionality of the data and the distribution parameters**.

### 6.3.1 Uniform Distributions

We now show the calculation of  $d_{AS}$  for a uniform distribution using the  $\ell_1$  ground distance function. In doing so we illustrate the general framework used to obtain characteristic numbers.

Let  $P$  be a uniform distribution between  $a$  and  $b$  ( $a, b \in \mathbb{R}$ ) with probability density function  $f(x) = \frac{1}{b-a}$  for  $x$  between  $a$  and  $b$  and 0 otherwise. Let  $m = \frac{b+a}{2}$ . To calculate  $d_{AS}$  we must first compute  $d_{AT}$  and  $d_{NT}$ .

Anti-transportation distance for  $P$  can be computed as follows:

$$\begin{aligned} d_{AT}(P, P) &= \int_a^b 2|x - m|f(x)dx \\ &= \int_a^b \frac{2|x - m|}{b - a} dx \end{aligned} \quad (6.13)$$

Removing the absolute value using symmetry around  $m$  we have

$$\begin{aligned} d_{AT}(P, P) &= \frac{2}{b - a} 2 \int_a^m (m - x)dx \\ &= \frac{2}{b - a} [2m(m - a) - (m - a)(m + a)] \\ &= \frac{2(m - a)^2}{b - a} \\ &= \frac{b - a}{2} \end{aligned} \quad (6.14)$$

The naive distance can be computed as follows:

$$d_{NT}(P, P) = \int_a^b \int_a^b \frac{|x-y|}{(b-a)^2} dy dx \quad (6.15)$$

Splitting one interval of integration to remove the absolute value sign we have

$$\begin{aligned} d_{NT}(P, P) &= \frac{1}{(b-a)^2} \int_a^b \left( \int_a^x (x-y) dy + \int_x^b (y-x) dy \right) dx \\ &= \frac{1}{2(b-a)^2} \int_a^b ((x-a)^2 + (b-x)^2) dx \end{aligned} \quad (6.16)$$

(performing the inner integrations)

$$\begin{aligned} &= \frac{1}{2(b-a)^2} \int_a^b (2x^2 - 2(a+b)x + (a^2 + b^2)) dx \\ &= \frac{1}{6(b-a)^2} (2(b-a)(b^2 + ab + a^2) - 3(a+b)^2(b-a) \\ &\quad + (a^2 + b^2)(b-a)) \\ &= \frac{1}{6(b-a)} (2(b^2 + ab + a^2) - 3(a+b)^2 + (a^2 + b^2)) \\ &= \frac{1}{6(b-a)} (2(a^2 + b^2) - 4ab) \\ &= \frac{b-a}{3} \end{aligned} \quad (6.17)$$

From Equations 6.14 and 6.17 we have

$$d_{AS}(P, P) = \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{(b-a)/2}{(b-a)/3} = \frac{3}{2}$$

**The self-anti-similarity distance of any univariate uniform distribution using the  $\ell_1$ -norm is therefore  $\frac{3}{2}$ .**

### 6.3.2 Normal Distributions

Let  $P$  be a normal distribution with mean  $\mu$  and variance  $\sigma^2$  and probability density function  $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  for  $x \in \mathbb{R}$ . The anti-transportation

distance for this normal distribution using  $\ell_1$  ground distance can be computed as:

$$d_{\text{AT}}(P, P) = \int_{-\infty}^{\infty} 2|x - \mu|f(x)dx$$

By symmetry of the integrand about  $\mu$  we have

$$\begin{aligned} d_{\text{AT}}(P, P) &= 2 \int_{\mu}^{\infty} 2(x - \mu) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= 2\sigma \int_0^{\infty} 2y \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \text{ substituting } y = \frac{x - \mu}{\sigma} \\ &= \frac{4\sigma}{\sqrt{2\pi}} \int_{-1}^0 dz \text{ substituting } z = -e^{-\frac{y^2}{2}} \\ &= \frac{2\sqrt{2}\sigma}{\sqrt{\pi}} \end{aligned} \tag{6.18}$$

Naive distance for this normal distribution:

$$\begin{aligned} d_{\text{NT}}(P, P) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |x - y|f(x)f(y)dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{|x - y|}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2 + (y-\mu)^2}{2\sigma^2}} dx dy \\ &= \sigma \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{|x' - y'|}{2\pi} e^{-\frac{x'^2 + y'^2}{2}} dx' dy' \\ &\quad \left( \text{substituting } x' = \frac{x - \mu}{\sigma}, y' = \frac{y - \mu}{\sigma} \right) \end{aligned}$$

Splitting the interval of integration in order to eliminate the absolute value sign we have

$$\begin{aligned} d_{\text{NT}}(P, P) &= \frac{\sigma}{2\pi} \int_{-\infty}^{\infty} \left( \int_{-\infty}^{x'} (x' - y') e^{-\frac{x'^2 + y'^2}{2}} dy' \right. \\ &\quad \left. + \int_{x'}^{\infty} (y' - x') e^{-\frac{x'^2 + y'^2}{2}} dy' \right) dx' \\ &= \frac{\sigma}{2\pi} \int_{-\infty}^{\infty} x' e^{-\frac{x'^2}{2}} \left( \int_{-\infty}^{x'} e^{-\frac{y'^2}{2}} dy' - \int_{x'}^{\infty} e^{-\frac{y'^2}{2}} dy' \right) dx' \end{aligned}$$

$$-\frac{\sigma}{2\pi} \int_{-\infty}^{\infty} e^{-\frac{x'^2}{2}} \left( \int_{-\infty}^{x'} y' e^{-\frac{y'^2}{2}} dy' - \int_{x'}^{\infty} y' e^{-\frac{y'^2}{2}} dy' \right) dx' \quad (6.19)$$

(re-arranging terms)

We note that for an even function  $f$

$$\int_{-\infty}^a f(x) dx = \int_{-\infty}^0 f(x) dx + \int_0^a f(x) dx = \int_0^{\infty} f(x) dx + \int_0^a f(x) dx$$

and

$$\int_a^{\infty} f(x) dx = \int_a^0 f(x) dx + \int_0^{\infty} f(x) dx = -\int_0^a f(x) dx + \int_0^{\infty} f(x) dx$$

so that the difference of the two left hand sides becomes

$$\int_{-\infty}^a f(x) dx - \int_a^{\infty} f(x) dx = 2 \int_0^a f(x) dx \quad (6.20)$$

On the other hand, for an odd function  $f$

$$\begin{aligned} \int_{-\infty}^a f(x) dx &= \int_{-\infty}^0 f(x) dx + \int_0^a f(x) dx = -\int_0^{\infty} f(x) dx - \int_a^0 f(x) dx \\ &= -\int_a^{\infty} f(x) dx \end{aligned} \quad (6.21)$$

Using the two results from Equations 6.20 and 6.21 in Equation 6.19 above we obtain

$$\begin{aligned} \frac{\pi}{\sigma} d_{\text{NT}}(P, P) &= \frac{1}{2} \int_{-\infty}^{\infty} x' e^{-\frac{x'^2}{2}} 2 \left( \int_0^{x'} e^{-\frac{y'^2}{2}} dy' \right) dx' \\ &\quad + \frac{1}{2} \int_{-\infty}^{\infty} e^{-\frac{x'^2}{2}} 2 \left( \int_{x'}^{\infty} y' e^{-\frac{y'^2}{2}} dy' \right) dx' \\ &= \int_{-\infty}^{\infty} \int_0^{x'} x' e^{-\frac{x'^2+y'^2}{2}} dy' dx' + \int_{-\infty}^{\infty} e^{-\frac{x'^2}{2}} \left[ -e^{-y'^2/2} \right]_{x'}^{\infty} dx' \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x' e^{-\frac{x'^2+y'^2}{2}} [0 < y' < x'] dy' dx' + \int_{-\infty}^{\infty} e^{-x'^2} dx' \end{aligned}$$

Exchanging integration order by Fubini's Theorem we have

$$\begin{aligned}
\frac{\pi}{\sigma} d_{NT}(P, P) &= \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \int_{-\infty}^{\infty} x' e^{-\frac{x'^2}{2}} [0 < y' < x'] dx' dy' + \sqrt{\pi} \\
&= \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \left[ -e^{-\frac{x'^2}{2}} \right]_y^{\infty} dy' + \sqrt{\pi} \\
&= \int_{-\infty}^{\infty} e^{-y^2} dy' + \sqrt{\pi} \\
&= \sqrt{\pi} + \sqrt{\pi} \\
&= 2\sqrt{\pi} \tag{6.22}
\end{aligned}$$

$$d_{NT}(P, P) = \frac{2\sigma}{\sqrt{\pi}} \tag{6.23}$$

From Equations 6.18 and 6.23 we have

$$d_{AS}(P, P) = \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{\frac{2\sqrt{2}\sigma}{\sqrt{\pi}}}{\frac{2\sigma}{\sqrt{\pi}}} = \sqrt{2}$$

**The self-anti-similarity distance of any univariate normal distribution using the  $\ell_1$ -norm, regardless of its mean and variance, is therefore  $\sqrt{2}$ .**

### 6.3.3 Exponential Distributions

Let  $P$  be an exponential distribution with rate parameter  $\lambda$  and probability density function  $f(x) = \lambda e^{-\lambda x}$  for  $x \geq 0$ . Since  $P$  is not symmetric, we need its cumulative distribution function  $F(x) = 1 - e^{-\lambda x}$  as well. The anti-transportation distance for this exponential distribution using  $\ell_1$  ground distance can be computed as:

$$\begin{aligned}
d_{AT}(P, P) &= \int_0^{\infty} |x - F^{-1}(1 - F(x))| f(x) dx \\
&= \int_0^{\infty} \lambda |x - F^{-1}(e^{-\lambda x})| e^{-\lambda x} dx
\end{aligned}$$



$$\begin{aligned}
&= \int_0^\infty \lambda|x + \frac{\log(1 - e^{-\lambda x})}{\lambda}|e^{-\lambda x} dx \\
&= \int_0^\infty |x\lambda + \log(1 - e^{-\lambda x})|e^{-\lambda x} dx \\
&= \frac{1}{\lambda} \int_0^\infty |y + \log(1 - e^{-y})|e^{-y} dy \quad (\text{substituting } y = \lambda x) \\
&= -\frac{1}{\lambda} \int_0^{\log(2)} e^{-y}(y + \log(1 - e^{-y}))dy \\
&\quad + \frac{1}{\lambda} \int_{\log(2)}^\infty e^{-y}(y + \log(1 - e^{-y}))dy \tag{6.24}
\end{aligned}$$

since  $h(y) = y + \log(1 - e^{-y})$  has a unique zero at  $\log(2)$ .

Consider  $I(a, b) = \int_a^b e^{-x}(x + \log(1 - e^{-x}))dx$ . Integrating by parts we have

$$\begin{aligned}
I(a, b) &= [-e^{-x}(x + \log(1 - e^{-x}))]_a^b + \int_a^b e^{-x}(1 + \frac{e^{-x}}{1 - e^{-x}})dx \\
&= [-e^{-x}(x + \log(1 - e^{-x}))]_a^b + \int_a^b (\frac{e^{-x}}{1 - e^{-x}})dx \\
&= [-e^{-x}(x + \log(1 - e^{-x}))]_a^b + \int_{e^{-a}}^{e^{-b}} (\frac{-1}{1 - z})dz \\
&= [-e^{-x}(x + \log(1 - e^{-x}))]_a^b + [\log(1 - z)]_{e^{-a}}^{e^{-b}} \\
&= [-e^{-x}(x + \log(1 - e^{-x}))]_a^b + [\log(1 - e^{-x})]_a^b \\
&= [-xe^{-x} + (1 - e^{-x})\log(1 - e^{-x})]_a^b \tag{6.25}
\end{aligned}$$

Let  $g(x) = (1 - e^{-x})\log(1 - e^{-x}) - xe^{-x}$ . Then by Equations 6.24 and 6.25

$$\begin{aligned}
\lambda d_{\text{AT}}(P, P) &= -I(0, \log(2)) + I(\log(2), \infty) \\
&= -\left(g(\log(2)) - \lim_{x \rightarrow 0^+} g(x)\right) + \left(\lim_{x \rightarrow \infty} g(x) - g(\log(2))\right) \\
&= -2g(\log(2)) + \lim_{x \rightarrow 0^+} g(x) + \lim_{x \rightarrow \infty} g(x) \tag{6.26}
\end{aligned}$$

These terms can be computed as

$$\begin{aligned} g(\log(2)) &= \frac{1}{2} \log\left(\frac{1}{2}\right) - \frac{1}{2} \log(2) \\ &= -\log(2) \end{aligned} \quad (6.27)$$

$$\begin{aligned} \lim_{x \rightarrow 0^+} g(x) &= \lim_{x \rightarrow 0^+} (1 - e^{-x}) \log(1 - e^{-x}) \\ &= \lim_{u \rightarrow 0^+} u \log(u) \\ &= \lim_{u \rightarrow 0^+} \frac{\log(u)}{1/u} \\ &= 0 \text{ (applying L'Hôpital's rule)} \end{aligned} \quad (6.28)$$

$$\begin{aligned} \lim_{x \rightarrow \infty} g(x) &= -\lim_{x \rightarrow \infty} x e^{-x} \\ &= -\lim_{x \rightarrow \infty} \frac{x}{e^x} \\ &= 0 \text{ (applying L'Hôpital's rule)} \end{aligned} \quad (6.29)$$

From Equations 6.26-6.29 we obtain

$$d_{\text{AT}}(P, P) = \frac{2 \log(2)}{\lambda}. \quad (6.30)$$

Naive distance for this exponential distribution:

$$\begin{aligned} d_{\text{NT}}(P, P) &= \int_0^\infty \int_0^\infty |x - y| f(x) f(y) dy dx \\ &= \lambda^2 \int_0^\infty \int_0^\infty |x - y| e^{-\lambda(x+y)} dy dx \\ &= \frac{1}{\lambda} \int_0^\infty \int_0^\infty |x - y| e^{-(x+y)} dy dx \end{aligned}$$

Splitting the interval of integration:

$$\begin{aligned} d_{\text{NT}}(P, P) &= \frac{1}{\lambda} \int_0^\infty \left[ \int_0^x (x - y) e^{-(x+y)} dy + \int_x^\infty (y - x) e^{-(x+y)} dy \right] dx \\ &= \frac{1}{\lambda} \int_0^\infty \left[ x e^{-x} \int_0^x e^{-y} dy - x e^{-x} \int_x^\infty e^{-y} dy - e^{-x} \int_0^x y e^{-y} dy \right] dx \end{aligned}$$

$$\begin{aligned}
& + e^{-x} \int_x^\infty y e^{-y} dy \Big] dx \\
= & \frac{1}{\lambda} \int_0^\infty [x e^{-x} [1 - e^{-x}] - x e^{-x} [e^{-x}] \\
& - e^{-x} [(1 - e^{-x}(1+x)) - e^{-x}(1+x)]] dx \\
= & \frac{1}{\lambda} \int_0^\infty [-2x e^{-2x} + x e^{-x} - e^{-x} + 2e^{-2x}(1+x)] dx \\
= & \frac{1}{\lambda} \int_0^\infty [2e^{-2x} + e^{-x}(x-1)] dx \\
= & \frac{1}{\lambda} [-e^{-2x}]_0^\infty + \frac{1}{\lambda} \int_0^\infty e^{-x}(x-1) dx \\
= & \frac{1}{\lambda} [1] + \frac{1}{\lambda} [(1-x)e^{-x}]_0^\infty - \frac{1}{\lambda} \int_0^\infty e^{-x} dx \text{ (integrating by parts)} \\
= & \frac{1}{\lambda} + \frac{1}{\lambda} [1] - \frac{1}{\lambda} [-e^{-x}]_0^\infty \\
= & \frac{1}{\lambda} + \frac{1}{\lambda} [1] - \frac{1}{\lambda} [1] \\
= & \frac{1}{\lambda} \tag{6.31}
\end{aligned}$$

From Equations 6.30 and 6.31 we have

$$d_{AS}(P, P) = \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{2 \log(2)}{\frac{1}{\lambda}} = 2 \log(2) \approx 1.3863.$$

**The self-anti-similarity distance of any univariate exponential distribution using the  $\ell_1$ -norm, regardless of its rate parameter, is therefore  $2 \log(2)$ .**

### 6.3.4 Summary of Analytic Results

Characteristic numbers for uniform, normal, and exponential distributions, analytically derived in Sections 6.3.1-6.3.3 are summarized in Table 6.2. Derivations for characteristic numbers using  $\ell_2$  ground distance are in Appendix A.

Distribution	Ground distance $d_\Omega$	
	$\ell_1$	$\ell_2^2$
<b>Normal</b>	$\sqrt{2}$	<b>2</b>
<b>Uniform</b>	<b>3/2</b>	<b>2</b>
<b>Exponential</b>	$2 \log(2)$	$\pi^2/6$

Table 6.2: **Characteristic numbers.** Analytically calculated characteristic numbers for three common distributions using two ground distance functions. These numbers are independent of any parameters of the distribution.

## 6.4 Empirical Results

In the previous section we calculated asymptotic values of self- $d_{AS}$  for distributions. Here we compute self- $d_{AS}$  empirically for point samples drawn repeatedly from various instances of those source distributions and two others. We ran over 5.8 million Monte Carlo simulations on a massively parallel grid computing system called the Open Science Grid (Pordes et al., 2008), which provided 75,000 CPU hours over the course of a weekend. The resulting self- $d_{AS}$  values are averaged over 500 runs for each parameter setting and displayed for five distributions in Figure 6.2 plotted against distribution parameters.

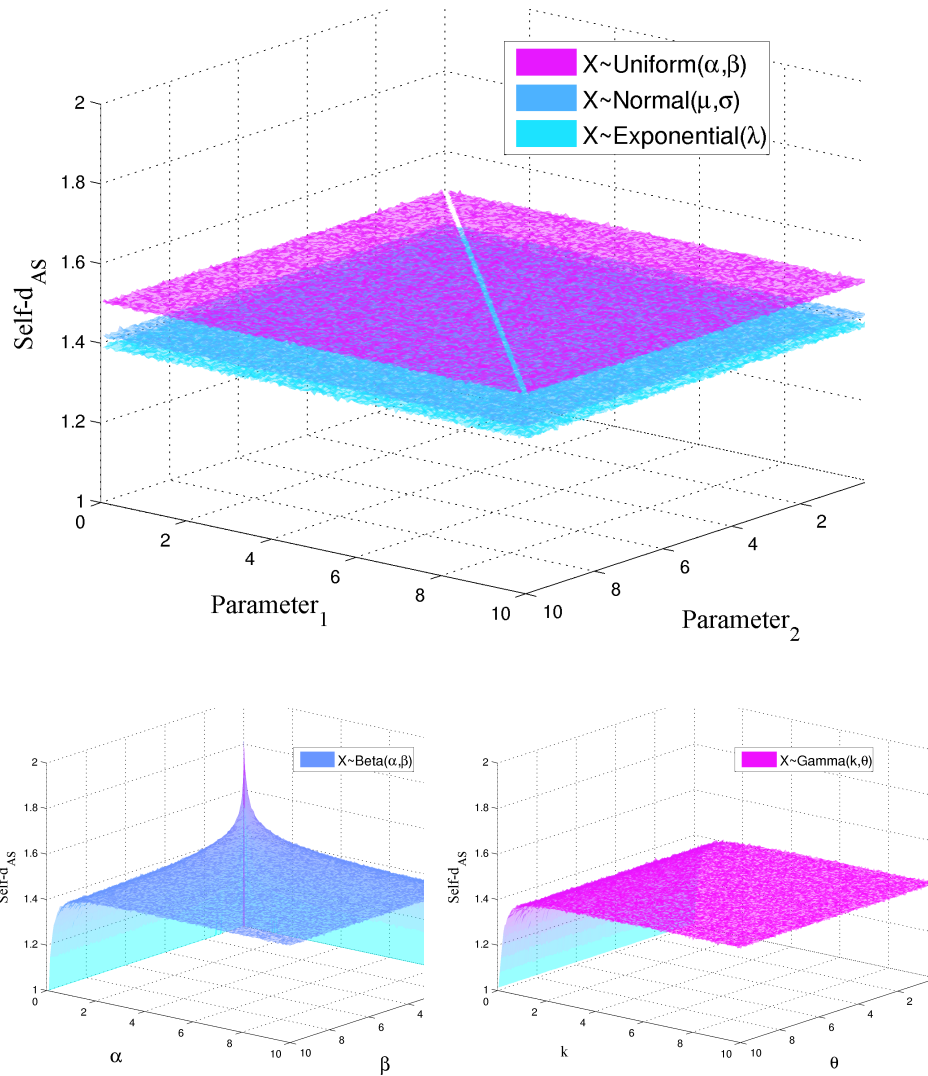


Figure 6.2: These plots show the variation of  $d_{AS}$  for 5 different distributions according to different values of their parameters. (a) As Table 6.2 indicates,  $d_{AS}$  for normal, exponential, and uniform distributions is constant and does not change with the parameters. (b) For the beta distribution  $d_{AS}$  is constant for parameter values  $> 1$ , and asymptotes to 1 and 2 for values  $< 1$ . (c) The value of  $d_{AS}$  for a gamma distribution is constant for any given value of the scale parameter, but varies with the shape parameter between 1 and  $\sqrt{2}$ .

## 6.5 Conclusion

In this chapter we introduced a new measure  $d_{AS}$  between probability distributions. When it is applied to a distribution and itself, it results — in many cases — in a static value for an entire family of distributions. This measure furthermore has a straightforward counterpart that can be applied to discrete samples; it is empirically verified that  $d_{AS}$  in point sets asymptotically approaches the characteristic numbers for their source distributions. This is a new theoretic property of a number of classical probability distributions that we call “characteristic numbers.” An entire family of distributions, without regard to parameters or dimensionality, surprisingly has a single characteristic number. We derive analytic results for the characteristic numbers of several ubiquitous distributions and demonstrate that it can be calculated in  $O(n \log(n))$  time for one-dimensional point samples. In the following chapter we present a new approach to goodness-of-fit testing using  $d_{AS}$ .

## 7 GOODNESS-OF-FIT TESTING

---

*From the earliest days of statistics, statisticians have begun their analysis by proposing a distribution for their observations and then, perhaps with less enthusiasm, have checked on whether this distribution is true.*

— D'AGOSTINO AND STEPHENS (1986, PG V.)

Goodness-of-fit tests (D'Agostino and Stephens, 1986; Rayner and Best, 2009; Thas, 2009; Thode, 2002) address the question: *is a given data sample consistent with having been drawn from a specified distribution?* For example, one may question whether a data set “appears” normal. Certainly, we cannot ask if they were *truly* sampled from a normal distribution. It is possible to be unlucky and have data generated from a uniform distribution appear normal or vice versa – particularly with small sample sizes. However, we can ask about the plausibility that the data conform to a particular distribution, for example, based on some common statistic describing them.

Self-anti-similarity distance, introduced in Section 6.1 enables a novel approach to goodness-of-fit testing. Since it is a *measure between a distribution and itself*, it allows us to create one-sample goodness-of-fit tests without reference to a hypothetical comparison distribution, e.g. a standard Gaussian. A key strength of this approach, specifically its sensitivity to the pairwise relationships between all points in a sample, provides the statistical power (Cohen, 1988) of our technique. This framework provides the most statistically powerful goodness-of-fit techniques of which we are aware.

### 7.1 Background

Goodness-of-fit tests play a fundamental role in statistics, particularly given the intensive use of computational statistical methods in recent years. Among their

most important applications is in assessing the validity of models involving statistical distributions, a step which is often neglected during the modeling process. This concern is raised by numerous statisticians, e.g. "One can only speculate on how many wrong decisions are made due to the use of an incorrect model." (Rayner and Best, 2009, p vii.). In the same vein, knowing the distribution of a sample can shed light on the process that generated it; if a suggested model for the process is correct, the sample data follow a specific distribution, which can be tested. Similarly, the parameters describing the distribution are sometimes recovered during goodness-of-fit testing and can be connected with important parameters describing the underlying model. Extensive discussion of the relation between goodness-of-fit testing and model validity may be found in Huber-Carol (2002).

Knowledge of the data distribution allows for application of standard statistical and estimation procedures. For example, if the data follow a normal distribution, inferences concerning the means and variances can be made using t-tests, analysis of variances, and F-tests. Similarly, if the residuals after fitting a regression model are normal, tests may be made on the model parameters. Estimation procedures such as the calculation of confidence intervals, tolerance intervals, and prediction intervals, often depend strongly on the underlying distribution. Also, when a distribution can be assumed, extreme tail percentiles, which are needed for highly unlikely events, can be computed. Additional issues relevant to statistical inference are outlined in D'Agostino and Stephens (1986).

There is also a close relationship between goodness-of-fit tests, particularly those employing smoothness approaches, and non-parametric density estimation (orthogonal series expansions) (Thas, 2009). While this is not uncommon in statistical applications, the connection with goodness-of-fit testing has not been well examined, where the parameters in the orthogonal series expansions



are replaced by estimates derived from distributional fit tests. This type of approach would allow improved density estimation.

While normality testing is by far the most common type of distributional test, specific tests exist for essentially every common family of distributions. For example, Knuth (1969) advocates empirical evaluation of pseudo-random number generators to ensure their output is compatible with that of a uniform distribution over  $[0, 1]$ . Perhaps the very first goodness-of-fit test was the  $\chi^2$  test for multinomial distributions, famously applied to demonstrate the Mendelian theory of genetics (Pearson, 1900; Fisher, 1918). Variants of the Pearson  $\chi^2$  test formed the predominant basis for goodness-of-fit testing until the mid 20<sup>th</sup> century.

## 7.2 Viewing as Hypothesis Tests

Hypothesis testing is a statistical method for testing whether observed data are consistent with an assumed scientific hypothesis. Formally, there are two hypotheses concerning an observation of  $X$ : the “null” hypothesis, and an alternative hypothesis that is its logical negation. The null hypothesis is rejected if the observed data are highly improbable according to the assumptions made by it. It is not possible to prove that the null hypothesis is true; we can only establish that the data have some level of consistency with it. Let  $X$  be a random variable and  $T$  a statistic computed on  $X$ . The alternative hypothesis corresponds to a set of values in the range of  $T(X)$  called the critical region; if the value of  $T(X)$  falls in this region, the null hypothesis is rejected. Values of  $T(X)$  less than a specified threshold correspond to the critical region. This threshold depends on how “significant” the test is required to be. Test significance is an upper bound on the probability of the test statistic (corresponding to the observed data) falling into the critical region purely due to chance, under the

null hypothesis. The “power” of a test, on the other hand, is the probability of correctly rejecting the null hypothesis. Power can be seen as a measure of how effectively a test statistic bifurcates the range of a statistic in order to differentiate between the two hypotheses. For a more detailed treatment of hypothesis tests see Wasserman (2013).

In the context of goodness-of-fit testing, a hypothesis test consists of a test statistic and thresholds corresponding to different significance levels and sample sizes (e.g. see D’Agostino and Stephens (1986); Stephens (1974)). The null hypothesis is that a sample was drawn from the distribution of interest, say a normal distribution. Given a sample of observations, if the computed test statistic lies within the critical region as defined by the thresholds (corresponding to sample size and significance level), the null hypothesis is rejected. Otherwise, it fails to be rejected. The sample size is important because some tests were designed for small samples, and others for large. Frequently, tests designed for small sample sizes do not work well for large sizes, and vice versa. This is particularly the case for hypothesis tests that attempt to find a connection between the point sample and a very specific property of the assumed source distribution, such as its moments, tail, and other phenomena.

### 7.2.1 Goodness-of-fit as a Hypothesis Test

An unusual property of goodness-of-fit tests is that they are typically framed as hypothesis tests in the null hypothesis  $H_0$  is desired to be true. For example, with a Kolmogorov-Smirnoff test (Stephens, 1974), we are interested in determining if a given sample is compatible with a standard normal distribution ( $\mu = 0, \sigma = 1$ ). The alternative hypothesis  $H_1$  is that it is not compatible, giving little or no information as to the actual source distribution of the data. Thus, goodness-of-fit testing is somewhat unique in that we want the null hypothesis

to be accepted.<sup>1</sup> In contrast, for much of hypothesis testing, the intent is to reject the null hypothesis (Lehmann and Romano, 2005; Wasserman, 2013), thereby proving some population effect, e.g., a group treated with drug had extended lifespan over a control. Even in cases where we seek to demonstrate a treatment had no effect, the test statistic is usually clear and we rely upon the Neyman-Pearson Lemma (Neyman and Pearson, 1933) to maximize the power of the test.

However, in goodness-of-fit testing, the range of possible statistics is immense and each has varying statistical power. Thus, numerous techniques with increasing degrees of statistical power have evolved to increase the likelihood that “failing to reject the null hypothesis” is the same as “accepting the null hypothesis.” However, none of these has singlehandedly proved to be superior to the others. Tests are often hand-picked based on visual characteristics of an empirical density function, e.g., its degree of kurtosis. We outline the range of approaches to goodness-of-fit testing in the next section.

### 7.3 Related Work

The body of literature exploring goodness-of-fit techniques is extensive. They share a common framework, however, of corresponding to hypothesis tests on unique statistics calculated on given samples. The works of D’Agostino and Stephens (1986), Rayner and Best (2009), and Thode (2002) provide a thorough discussion of these tests and the different approaches taken to construct them. Stephens (1974) also provides a short but informative overview. We compare statistical power of relevant named tests below with our approach in Section 6.4. In the discussion here,  $X$  refers to an entire sample and  $x_i$  refers to a member of that sample.

---

<sup>1</sup>Note, “accepting the null hypothesis” is often elliptically referred to as “failing to reject the null hypothesis.” Namely, not being able to reject something generally does not prove it. In spite of this, both terms are often used equivalently in statistics and we do so here.

### 7.3.1 Graphical Tests

The oldest and most straightforward approach to identifying distributions is to look at them (Cleveland and McGill, 1985; J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey, 1983). Tools such as quantile based Q-Q plots (Wilk and Gnanadesikan, 1968) have become popular for this purpose. However, visual approaches are subject to the vagaries of perceptual processing and inductive biases. They have been subject to little scientific study until fairly recently (Cleveland and McGill, 1987), although they are widely employed by statisticians. Their greatest utility may well be in dismissing candidate distributions. Visual observation of a sample is generally considered an analytic prerequisite, e.g., “There is no excuse for failing to plot and look.” Tukey (1977, p. 43).

### 7.3.2 $\chi^2$ Based Tests

Classical  $\chi^2$  tests introduced by Pearson (1900) are generally considered the first approaches to formalizing goodness-of-fit testing. He introduced a discrete, multinomial framework motivated by a mounting body of evidence that biological populations were not normally distributed. The test statistic,  $X^2 = \sum_{i=1}^n \frac{(x_i - E_i)^2}{E_i}$  measures the difference between observed ( $x_i$ ) and theoretically expected ( $E_i$ ) values for a series of discrete events. Comparing the value of the statistic to the  $\chi^2$  distribution allows derivation of a  $p$ -value for the test. While the original formulation was on discrete (or binned) values, much early work in distributional testing was focused on generalizing the  $\chi^2$  test for continuous data and other distributions, notably the *smooth* tests of Neyman (1937). A modern smoothness test we compare with is the 4<sup>th</sup> order Hermite polynomial statistic ( $\hat{S}_4$ ) described in Rayner and Best (1986).

### 7.3.3 Tests Based on EDF Statistics

A very popular class of goodness-of-fit techniques are based on empirical distribution functions (EDFs). The simplest and most well-known of these is the Kolmogorov-Smirnov test (Kolmogorov, 1933; Smirnov, 1948). Given  $n$  samples, one defines  $F_n(x) = \frac{\text{number of samples} \leq x}{n}$  and  $F(x)$  is the probability of an observation less than or equal to  $x$ . The test statistic  $D$  is then defined as  $D = \sup_x |F_n(x) - F(x)|$ . A wide variety of statistics are based upon the discrepancy between observed and predicted values according to an EDF. These include Cramer-von Mises, Anderson-Darling, and Watson tests (Stephens, 1974), which all modify the  $D$  statistic in some way. Modifications to Kolmogorov-Smirnov tests were motivated by its low statistical power, as observed in Section 6.4.

### 7.3.4 Regression and Correlation Tests

Whereas EDF-based tests make use of order statistics, regression tests are a formalization of the graphical tests discussed above. Namely, they fit a straight line to a sample and the tests are based on statistics associated with this line. If the test statistic is derived from the correlation coefficient between the line and the sample, the test is called a correlation test. The best known of these tests is perhaps the Shapiro-Wilk test for normality (Shapiro and Wilk, 1965), which is a regression test based on the residuals. As the full test statistic is somewhat involved, e.g., (D'Agostino and Stephens, 1986, pg. 206–208), a simpler variation is the d'Agostino test, whose statistic is  $D_A = \sum_i^n x_i(i - \frac{1}{2}(n + 1))/Sn^{3/2}$ , where  $S = (\sum x_i - \bar{X})^{1/2}$ .

### 7.3.5 Transformation Methods

Among the most interesting approaches to distributional testing is changing the underlying distribution of the initial sample to have a distribution for

which we already have a good test. For example, a straightforward case is transforming a lognormal distribution into a normal one; we simply take the log of each term. We can then use a normality test to determine if the original data are compatible with a lognormal distribution. We note the transformations themselves may be approximate, introducing their own error into the goodness-of-fit computation. The Watson statistic for testing uniformity,  $U^2 = (1/12N) + \sum_{i=1}^n ((2i-1)/2N - x_i)^2 - N(\bar{X} - 0.5)^2$ , has been found to be quite useful in transformational tests given its reasonable power against numerous classes of alternatives (Quesenberry and Miller, 1977).

### 7.3.6 Other Approaches

Two other avenues of measuring goodness of fit deserve mention. The first is moment techniques that compute descriptive statistics of data sets such as skew, kurtosis, and variance. The difficulty with these approaches is that they are exceedingly complex and few exact results can be calculated for non-normal distributions.

A second and highly important class of alternatives correspond to Bayesian approaches to goodness-of-fit testing, which are often called *posterior predictive checks* or *conditional predictive checks* (Gelman et al., 2003; Mukhopadhyay et al., 2005; Rubin, 1984). These are natural in Bayesian modeling and analysis, where we may have reasonable priors on the distributions of sampled data. The Bayesian community is notable for typically including distributional checks in modeling. An increasing number of the above approaches have Bayesian variants (Gelman et al., 2003).

## 7.4 Building a Hypothesis Test for Normality

Normality testing is by far the most common goodness-of-fit test (see Section 7.3). This is in large part due to the assumption made in linear regression techniques that the residuals be normally distributed. We use the analytic results for  $\ell_1$  and  $\ell_2^2$  to construct a test statistic  $AS$  (from anti-similarity) for samples from normal distributions. The values of  $d_{AS}$  taken by samples quickly approach the asymptotic values of the distributions shown in Table 6.2. Thus, our statistic is determined by the difference between a sample's  $d_{AS}$  and the characteristic number for the parent distribution. We therefore define our anti-similarity-based normality test statistic ( $AS$ ) for a point set  $P$  as:

$$AS = |d_{AS}(P, P; \ell_1) - \sqrt{2}| + |d_{AS}(P, P; \ell_2^2) - 2| \quad (7.1)$$

This statistic is highly robust because it incorporates values from *all* pairwise distances between points; it does not rely only on outliers or on non-robust statistics such as mean, kurtosis, or skewness.

### 7.4.1 The $AS$ Normality Test

$AS$  is used to construct a hypothesis test as follows. We estimate the distribution of  $AS$  under the null hypothesis empirically using samples drawn from normal distributions. We selected two significance levels (0.05 and 0.1) and estimated the thresholds for  $AS$  corresponding to each level for different sample sizes. Given our statistic is lower bounded by zero, we use a one-tailed (upper tail) test. Monte Carlo simulations provided the necessary upper bound for obtaining 95% confidence intervals. If the test statistic lies above this bound, we reject the null hypothesis that it came from a normal distribution. Upper bounds for two significance levels are presented for various sized samples drawn from normal distributions in Table 7.1. Similar tables were constructed for other distributions

as well. These tables were used to construct goodness-of-fit hypothesis tests based on  $AS$ . Our approach provides a powerful normality test, which is particularly noticeable on small samples. Table 7.2 provides power comparisons of the  $AS$  test for several different sample sizes at the 5% significance level with the most common tests for normality. These include the Cramer-von Mises, Watson, Kolmogorov-Smirnov, Anderson-Darling, Neyman-type smooth, Lilliefors, and Shapiro-Wilk ( $W$ ) tests.

#### 7.4.2 Multivariate Normality Testing with $d_{AS}$

Self-anti-similarity distance can be used to construct goodness-of-fit tests for multivariate data as well, following the procedure above. Multivariate goodness-of-fit testing has received little attention in the literature, as opposed to univariate cases (D'Agostino and Stephens, 1986). As for the univariate case above, we conducted experiments to derive confidence intervals using Monte Carlo simulations for multivariate normal distributions. We measured the power of a test using the  $\ell_1$  norm to reject samples from multivariate uniform distributions. Figure 7.1 shows the variation of power with respect to point set size and dimensionality. We note that as dimensionality increases, fewer points are required to achieve high power. This is a result of each point contributing increasing information about its parent distribution as dimensionality increases.

Sample size(n)	Threshold	
	$\alpha = 0.05$	$\alpha = 0.1$
10	.3209	0.1439
20	.2071	0.0876
30	.1436	0.0671
50	.1122	0.0488
100	.0681	0.0308

Table 7.1: **AS Statistic.** This table provides upper bounds at two significance levels for values of  $AS$  for normality testing of different point set sizes. If  $AS$  lies above the corresponding bound for that point set size, we reject the hypothesis that the sample originated from a normal distribution.



Distribution	n	AS	$W^2$	$U^2$	$D$	$A^2$	$\hat{S}_4$	$D'$	$W$
<i>Uniform</i> (-1, 1)	10	<b>8.2</b>	6.9	8.0	6.1	7.4	2.2	6.1	4.7
	20	14.3	14.8	<b>16.8</b>	8.9	16.2	0.1	9.0	9.9
	30	21.9	21.5	25.5	15.1	<b>27.9</b>	0.0	15.7	21.6
	50	46.7	43.5	49.4	26.2	<b>57.4</b>	7.4	26.2	57.1
	100	80.0	85.3	88.4	62.0	95.2	25.4	61.7	<b>98.2</b>
<i>Exponential</i> (1)	10	<b>45.0</b>	38.0	37.4	31.7	40.8	34.5	31.7	35.6
	20	<b>82.5</b>	72.5	70.6	58.5	78.0	62.8	59.1	75.7
	30	<b>95.6</b>	88.5	84.2	77.1	92.1	82.9	77.4	93.1
	50	<b>99.9</b>	99.2	98.4	95.7	99.7	97.6	95.8	99.8
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
<i>Lognormal</i>	10	<b>60.8</b>	55.7	54.4	46.6	58.0	48.8	46.4	53.5
	20	<b>93.6</b>	87.8	86.8	78.0	90.3	84.6	78.5	89.5
	30	<b>99.5</b>	98.5	97.3	94.6	98.9	95.3	94.7	98.6
	50	<b>100.0</b>	99.9	99.8	99.7	<b>100.0</b>	<b>100.0</b>	99.7	<b>100.0</b>
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
<i>t</i> (1)	10	56.7	60.1	<b>60.9</b>	57.8	60.1	57.5	57.8	57.7
	20	83.8	<b>88.3</b>	88.2	85.1	<b>88.3</b>	86.5	85.1	87.1
	30	94.9	95.8	<b>96.0</b>	94.3	95.6	95.1	94.3	95.0
	50	99.5	99.3	99.4	98.8	99.4	<b>100.0</b>	98.8	99.3
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
$\chi^2_8$	10	<b>13.3</b>	12.0	11.7	10.9	<b>13.3</b>	<b>13.3</b>	10.9	11.3
	20	<b>26.9</b>	22.7	20.3	18.6	25.5	25.3	18.8	23.4
	30	<b>44.8</b>	31.5	26.2	24.6	35.9	37.2	24.9	34.8
	50	<b>66.0</b>	51.4	43.3	40.2	57.5	54.0	40.5	59.2
	100	91.2	84.0	75.9	71.1	89.6	89.0	71.1	<b>92.4</b>
<i>Beta</i> (2, 1)	10	<b>13.3</b>	12.5	12.8	11.0	13.1	6.7	10.8	8.8
	20	<b>26.2</b>	20.7	20.2	16.1	22.9	8.1	16.1	15.8
	30	<b>51.7</b>	36.4	34.9	26.2	42.4	7.8	26.5	35.3
	50	<b>75.7</b>	60.7	58.4	43.5	71.7	20.0	43.6	72.9
	100	96.2	92.5	90.1	79.9	98.2	84.4	79.8	<b>99.1</b>
<i>Gamma</i> (1, 2)	10	<b>46.9</b>	37.0	36.8	29.0	39.3	34.0	28.8	33.7
	20	<b>81.2</b>	71.7	68.5	56.6	76.5	62.8	56.8	73.0
	30	<b>95.6</b>	90.3	87.5	78.0	93.5	81.6	78.8	92.6
	50	<b>99.8</b>	99.1	98.4	96.1	99.7	96.5	96.1	<b>99.8</b>
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
<i>Cauchy</i> (0, 1)	10	55.2	62.4	<b>62.6</b>	58.4	62.2	57.9	58.3	58.8
	20	83.6	86.9	86.9	82.9	<b>87.8</b>	86.8	82.9	86.4
	30	95.1	96.1	96.3	93.9	<b>96.5</b>	95.6	94.0	96.2
	50	99.6	<b>99.7</b>	<b>99.7</b>	99.6	<b>99.7</b>	99.6	99.6	99.6
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>

Table 7.2: This table contains the statistical powers Cohen (1988) of our test statistic ( $AS$ ) with the Cramer-von Mises ( $W^2$ ), Watson ( $U^2$ ), Kolmogorov-Smirnov ( $D$ ), Anderson-Darling ( $A^2$ ), Neyman-type smooth ( $\hat{S}_4$ ), Lilliefors ( $D'$ ), and Shapiro-Wilk ( $W$ ) tests. These tests were performed for each given sample size on 1000 instances drawn from each specified distribution, with  $\alpha = 0.05$ . The bolded red values indicate the statistically most powerful test for each sample from a given distribution. All tests have approximately 5% statistical power on normal samples.

Variation of power with dimensionality and number of points

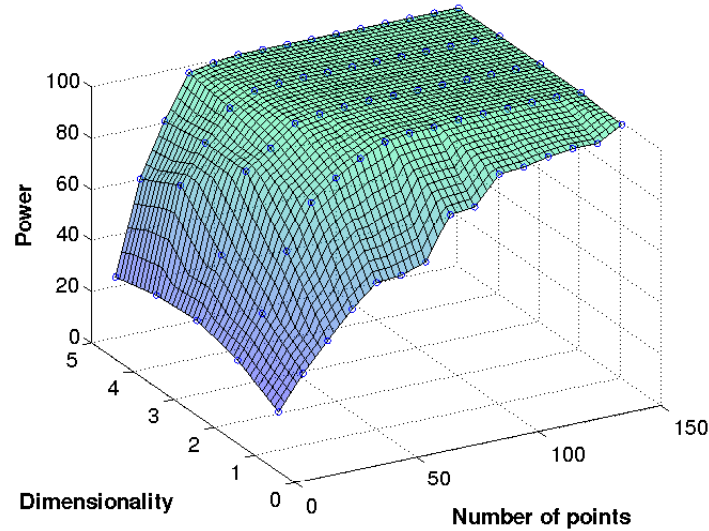


Figure 7.1: Examining the power of characteristic numbers in  $\ell_1$  in a multivariate normality test to reject multivariate uniform samples as a function of the number of dimensions and points. Holding the number of points constant, we note the statistical power increases with dimensionality, as each point is contributing more information about its source distribution.

## 7.5 Conclusion

In this chapter we introduced a new framework for deriving goodness-of-fit tests based on the concept of characteristic numbers developed in Chapter 6. We used the characteristic numbers for the normal distribution to present a practical goodness-of-fit test for normality involving our new statistic  $AS$  that measures deviations from expected asymptotic values. The test statistic incorporates a robust measure from optimization theory that is sensitive to the pairwise relationships between all sample points. Our framework provides the most statistically powerful goodness-of-fit techniques of which we are aware.

## 8 EVALUATIONS

---

This chapter contains a summary of experiments conducted in the course of this research. Results from previous chapters, together with results from three other experiments not previously mentioned, are gathered together here in one place to provide an easy-to-peruse collection of results from experiments and learning tasks using the point set representation. The section-topic correspondence below is as follows:

- 8.1: Classification of Samples from Probability Distributions (Chapter 2)
- 8.2: Ensemble Clustering (Chapter 4)
- 8.3: Neuroimaging (Chapter 5)
- 8.4: Goodness of fit (Chapter 7)
- 8.5: Document Classification
- 8.6: Object Classification in Images
- 8.7: Protein Structure Similarity Detection

### 8.1 Classification of Samples from Probability Distributions

One of the common assumptions in machine learning is that data from different classes originate from different underlying distributions. When data are represented in the form of point sets, the classification problem becomes one of being able to differentiate between *samples* of points from differing distributions. Below we show results from experiments wherein we sample sets of points from two different multivariate probability distributions and evaluate the utility of various spatial and non-spatial point set comparison methods in classifying these point sets.

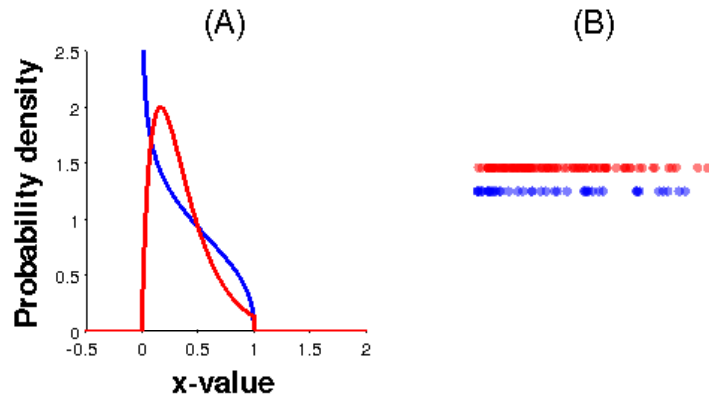


Figure 8.1: (A) Probability density function plots for Experiment 1 in Section 8.1. The plot in blue is a beta distribution with parameters 0.8 and 1.4, and the one in red is a truncated gamma distribution, with shape parameter 1.83 and rate parameter 0.19. (B) Two examples of point sets that are sampled from distributions shown in (A).

### 8.1.1 Experiment 1

Consider the pair of 1-dimensional distributions shown in Figure 8.1 where the densities of both distributions are plotted along the  $y$ -axis; one in blue and the other in red. The distribution shown in blue is a beta distribution (parameters  $\alpha = 0.8$  and  $\beta = 1.4$ ) and the one in red is a truncated gamma distribution (shape  $k = 1.83$  and rate  $\theta = 0.19$ ). The parameters for these distributions were chosen so as to make both the means and variances of these distributions identical to one another respectively.

We sampled 100 point sets from each distribution, each containing a varying numbers of points between 30 and 60. We then trained a support vector machine (Shawe-Taylor and Cristianini, 2000) to separate between point sets originating from these two distributions, using  $S_{IM}$ , density overlap, lift kernel, Bhattacharyya kernel (Kondor and Jebara, 2003), and pyramid match (Grauman

	Accuracy	Time
Density overlap ( $\sigma = 0.05$ )	<b>93.5%</b>	2.71s
SIM	87%	33.4s
Pyramid match	85.8%	4.84s
Lift kernel ( $\rho = 400$ )	74%	45.25s
Bhattacharyya kernel	83%	120.9s

Table 8.1: Accuracy results for five point set similarity measures on classifying synthetic data in 1 dimension sampled from the distributions shown in Figure 8.1. 100 point sets of varying cardinality between 30 and 60 were sampled from each distribution and used to train a support vector machine classifier. This classifier was then tested on another 100 samples; the classification accuracies are shown in the table above.

and Darrell, 2007) as similarity functions. Classification accuracy results on a holdout set of another 100 point sets from each distribution are tabulated in Table 8.1.

### 8.1.2 Experiment 2

We perform another experiment similar to the one above but with the two-dimensional distributions shown in Figure 8.2. The density functions in one dimension of both distributions are plotted along the  $x$ -axis, and along the other dimension on the  $y$ -axis. One distribution (in blue) follows a beta distribution with parameters (1.3, 1.3) in the  $x$ -coordinate and a normal distribution with mean 0.5 and variance 0.04 in the  $y$ -coordinate. The other distribution (in red) follows a uniform distribution with parameters (0, 1) in the  $x$ -coordinate and a beta distribution with parameters  $\alpha = 2.4$  and  $\beta = 2.4$  in the  $y$ -coordinate.

Similar to Experiment 1 above, we sampled 100 point sets from each distribution, each containing a varying numbers of points between 30 and 60. We then trained a support vector machine to separate between point sets originating from these two distributions, using SIM, density overlap, lift kernel, Bhattacharyya kernel (Kondor and Jebara, 2003), and pyramid match (Grauman

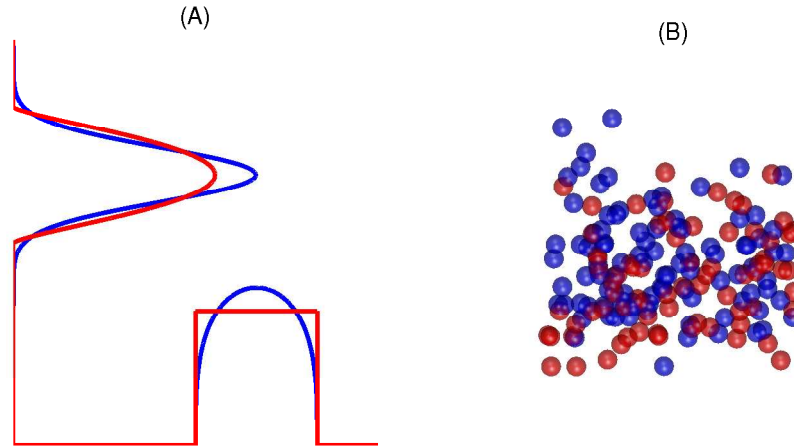


Figure 8.2: (A) Probability density function plots for Experiment 2 in Section 8.1. The plots in blue represent the density in each dimension of one distribution; it follows a beta distribution with parameters 1.3 and 1.3 in the  $x$ -dimension and a normal distribution with mean 0.5 and variance 0.04 in the  $y$ -dimension. The plots in red represent the per-dimension densities of the second distribution; along the  $x$ -dimension is a uniform distribution on  $[0, 1]$  and along the  $y$ -dimension a beta distribution with parameters 2.4 and 2.4. (B) Two examples of point sets that are sampled from distributions shown in (A).

	Accuracy	Time
Density overlap ( $\sigma = 0.05$ )	<b>77.5%</b>	2.96s
$S_{IM}$	76%	107.52s
Pyramid match	59.3%	5.3s
Bhattacharyya kernel	63%	116.79s
Lift kernel ( $\rho = 1000$ )	54.5%	79.48s

Table 8.2: Accuracy results for five point set similarity measures on classifying synthetic data in 2 dimensions sampled from the distributions shown in Figure 8.2. 100 point sets of varying cardinality between 30 and 60 were sampled from each distribution and used to train a support vector machine classifier. This classifier was then tested on another 100 samples; the classification accuracies are shown in the table above.

Data Set	Dimensionality	# of instances	# of Classes
Iris	4	150	3
Wine	13	173	3
Ionosphere	34	351	2
Soybean	35	47	4
ISOLET	617	1559	26
MNIST (Test)	784	10,000	10
MNIST (Full)	784	60,000	10

Table 8.3: This table displays the characteristics of data sets used to evaluate ensemble clustering algorithms in this chapter. The data sets Iris, Wine, Ionosphere, Soybean, and ISOLET are from the UCI Machine Learning Repository Asuncion and Newman (2007) and form a diverse collection of data set with respect to high and low dimensionalities, small and large numbers of instances, and few and many classes. The MNIST LeCun et al. (1998) data sets are the train and test sets respectively of a large and popular digit recognition image database.

and Darrell, 2007) as similarity functions. Classification accuracy results on a holdout set of another 100 point sets from each distribution are tabulated in Table 8.2.

## 8.2 Ensemble Clustering

In this section we evaluate our ensemble clustering algorithm SEC and compare results with other methods such as LIFTKM (Raman et al., 2011) (dimensionality  $\rho = 2000$ ), the bipartite graph methods of Fern and Brodley (2004), and the hypergraph methods of Strehl and Ghosh (2003). We applied all these methods to standard labeled classification data sets (detailed in Table 8.3) that are used in the literature in evaluation of clustering algorithms and ensemble clustering techniques.

We generated the ensemble using a combination of the methods described in Section 4.3 and using the k-means algorithm (MacQueen et al., 1967), spectral clustering (Ng et al., 2002), and affinity propagation (Dueck and Frey, 2007). We generated hundreds of clusterings and set a threshold of 0.1 for the diversity

Data Set	Iris	Wine	Ionosphere	Soybean
<b>SEC error</b>	<b>10.67%</b>	<b>29.78%</b>	<b>28.77%</b>	<b>29.79%</b>
LiftKM error	11.33%	37.64%	33.90%	<b>29.79%</b>
KRF error	<b>10.67%</b>	53.37%	<b>28.77%</b>	<b>29.79%</b>
Fern-Brodley error	<b>10.67%</b>	42.13%	32.76%	<b>29.79%</b>
Mean error	15.68%	32.64%	27%	29.1% %
Min error	10.67%	29.78%	0%	0%
Diversity	0.11	0.19	0.06	0.24
SEC Time	11.6s	16.4s	10.5s	3.6s

Data Set	ISOLET	MNIST (Test)	MNIST (Full)
<b>SEC error</b>	<b>41.24%</b>	<b>40.03%</b>	<b>42.64%</b>
LiftKM error	46.76% ( $\rho = 4000$ )	63.94%	64.13%
KRF error	43.43%	-	-
Fern-Brodley error	44.26%	-	-
Mean error	48.91%	52.88%	51.92%
Min error	42.85%	50.40%	50.82%
Diversity	0.29	0.62	0.66
SEC Time	860s	483s	6733s

Table 8.4: This table presents the results of applying different ensemble clustering algorithms on standard data sets. The first row contains the data set used, the second the error rate for our method (SEC), and the third the error rate of LiftKM. The fourth row contains the least error rate of the knowledge reuse framework (KRF) of Strehl and Ghosh (2003) and the fifth the least error rate among the three methods proposed in Fern and Brodley (2004). The sixth and seventh rows contain the mean and minimum errors respectively of members of the ensemble relative to the “true” labeling of the data set. The eighth row displays the diversity measurement of the ensemble and the final row the time in seconds that SEC took to arrive at a consensus clustering on that data set.



(defined in Section 4.3) in order to proceed with the final consensus step. Each ensemble clustering algorithm was given the same ensemble from which to derive its consensus clustering, wherever applicable.

### 8.2.1 Measuring Accuracy

A natural criterion to measure the usefulness of each method is the overall accuracy of the final consensus clustering and the improvement over individual members of the ensemble. We define accuracy by the following formula:

$$Accuracy = \max_p \frac{1}{n} \sum_{i=1}^k T(C_{p(i)}, L_i) \quad (8.1)$$

where  $L_i$  is the  $i^{\text{th}}$  class in the labeled data set,  $C_j$  is the  $j^{\text{th}}$  cluster in the consensus clustering,  $p$  varies over all permutations of labeling assignments between the clusters of the consensus clustering and the classes of the data set, and  $T(C_j, L_i)$  is the number of points that occur in both  $C_j$  and  $L_i$ .

We approximate the best correspondence  $p$  of cluster labels from the consensus clustering to the data set labels by solving the correspondence problem using the Hungarian algorithm (Munkres, 1957). The “accuracy” of each clustering with respect to the given labels is then computed using this correspondence.

### 8.2.2 Results

Table 8.4 shows the results of applying the above-mentioned ensemble methods to data sets from the UCI Machine Learning Repository Asuncion and Newman (2007), and the MNIST digit recognition database LeCun et al. (1998) which contains 60,000 data in 784 dimensions, categorized into 10 classes. These data sets were chosen to maximize the diversity in dimensionality, numbers of classes, and numbers of instances, to demonstrate the wide applicability of ensemble clustering.

Our method is referred to as SEC in the table. In each column we show results from the application to one data set of SEC and three other state-of-the-art ensemble clustering algorithms, each representative of a different approach to computing the consensus. In the columns we show the error of each algorithm rather than its accuracy to better demonstrate differences in performance. The least error in each case is shown in bold. Additionally, we also provide information such as the mean and minimum error of all clusterings in the ensemble. Finally, we show the running time of SEC on that data set.

As the bolded numbers show, SEC consistently finds a consensus clustering that has the least error rate among all the methods tested. In the case of the Iris data set, three of the four methods arrived at a consensus that has a mis-clustering error lower than the mean error over the ensemble, and equal to the minimum error. The data set Wine has a similar result for SEC. For the Ionosphere and Soybean data sets all methods perform similarly and arrive at clusterings that are comparable to the mean error in the ensemble. In the ISOLET and MNIST data sets, SEC reports an error that is significantly lower than the mean of the ensemble and, more importantly, also lower than the minimum error. In the case of MNIST, the difference is as much as 10.37%, corresponding to an error reduction of 20.58%. This is especially interesting as it indicates that with no further supervision we are able to use information from disparate clusterings to reduce the mis-clustering error well below even the best performing member of the ensemble.

### 8.3 Neuroimaging

The experiments below are designed to yield a better understanding of neural patterns of change in the brain in relation to aging and the presence of risk factors for AD. Chapter 5 contains a full description of the data set and prepro-

cessing steps applied to each image. Here we discuss results from a series of three experiments.

### 8.3.1 Detecting Chronological Order of Scans

We begin with a “simple” classification problem. For longitudinal data, one instance of ground truth is the chronological order in which the data sets were collected. Thus, a natural question is: can we determine this order for a given individual (see Figure 5.8 for an example)? In other words, given two scans, our task is to identify which was taken earlier.

Recall that we represent any region (or collection of regions) of interest in the brain as a point set  $R = \{(v_1, w_1), (v_2, w_2), \dots, (v_N, w_N)\}$  where each  $v_i \in \mathbb{R}^3$  is a voxel position,  $w_i \in \mathbb{R}$  some value of interest at that voxel, and  $N$  is the number of voxels in the region(s). We define another point set  $\Delta R$  corresponding to the same region in a new image constructed from the *difference* of the two brain scans taken from the same subject and time 1 and time 2.  $\Delta R = \{(v_1, \Delta w_1), \dots, (v_N, \Delta w_N)\}$  where  $\Delta w_i$  is the change in FA at voxel  $i$  from time 1 to time 2. We then perform voxel selection according to the  $Q$  and  $\text{CONS}$  criteria outlined in Section 5.6.1

$$\Delta \widehat{R}_Q(\tau) = \{(v_i, \Delta w_i) \mid (v_i, \Delta w_i) \in \Delta R \text{ and } Q(v_i) > \tau\} \quad (8.2)$$

$$\Delta \widehat{R}_{\text{CONS}}(\tau) = \{(v_i, \Delta w_i) \mid (v_i, \Delta w_i) \in \Delta R \text{ and } \text{CONS}(v_i) > \tau\} \quad (8.3)$$

These are the voxels that will be extracted from each scan.

For each of the 118 subjects, we construct two “difference” images. The first subtracts the latter image from the earlier one (the “positive difference image”), and the second by reverses the order of subtraction (the “negative difference image”). This is done so that when given two new images from a single subject with no ordering information, we perform the subtraction in

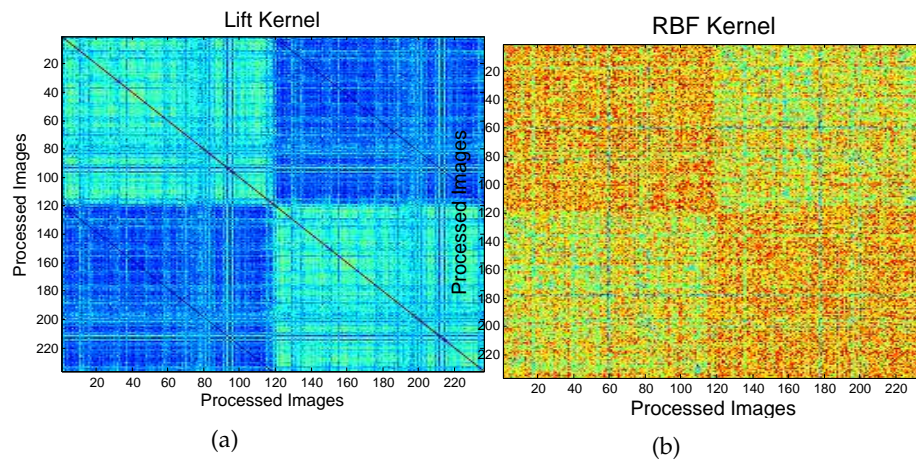


Figure 8.3: **Gram Matrix.** The panels in this image show Gram matrices for the first experiment represented as images. A Gram matrix in this context is composed of kernel values between all instances in the data set. Our data set contains two scans from each of 118 subjects, leading to 236 positive and negative difference images. The Gram matrix therefore contains similarity values for every pair of difference images in this data set. The first 118 entries along the  $x$  and  $y$  axes correspond to the positive difference images, and thereafter the negative difference images. Panels (a) and (b) correspond respectively to Gram matrices constructed using the lift and RBF kernels. As the colors in the image in panel (a) show, the positive images are similar to one another, as are the negative images to one another, but the positive and negative images are not similar to each other (reflected by the darker color). These differences are exploited by a linear SVM classifier to yield the accuracy results in Table 8.6.

an arbitrary manner and compute which set of difference images this new difference image is more “similar” to, using the spatially aware kernels in Chapter 2.

Since there are an equal number of positive and negative difference images, the baseline accuracy for this experiment is 50%.

We trained a support vector machine (SVM) (Shawe-Taylor and Cristianini, 2000) with the lift kernel (Section 2.6) to classify “positive” and “negative” difference images. For comparison purposes to a widely used non-spatially sensitive kernel, we also provide results for the radial basis function (RBF) kernel. Figure 8.3 shows a visualization of the Gram matrices corresponding to these

Region	Label
Corpus Callosum (whole)	101
Corpus Callosum (splenium)	5
Corpus Callosum (genu)	3
Cingulum bundle (R & L)	35
Superior longitudinal fasciculus (R & L)	41
Uncinate fasciculus (R & L)	45
Fornix (column, body, and cres)	6

Table 8.5: **Brain regions.** This table lists the regions in the brain we conducted analysis on. Each region is assigned a label which it is referred to by in the results in Table 8.6.

Region	Mode	$ \Delta\hat{R}_{\text{CONS}}(\tau) $	RBF	Lift
101	FA	1512.7 voxels	79.7%	<b>91.5%</b>
	MD	4803.5 voxels	75.4%	<b>87.3%</b>
5	FA	425.7 voxels	84.7%	<b>93.2%</b>
	MD	2282.8 voxels	69.5%	<b>81.4%</b>
3	FA	170.6 voxels	80.9%	<b>88.1%</b>
	MD	948.2 voxels	74.6%	<b>86.4%</b>
35	FA	1466.2 voxels	84.7%	<b>91.5%</b>
	MD	1090.2 voxels	64.4%	<b>85.6%</b>
41	FA	2032.5 voxels	83.1%	<b>92.4%</b>
	MD	505.6 voxels	63.6%	<b>81.4%</b>
45	FA	124.8 voxels	73.8%	<b>86.4%</b>
	MD	127.3 voxels	64.4%	<b>79.7%</b>
6	FA	19.3 voxels	66.1%	<b>76.3%</b>
	MD	203.2 voxels	71.2%	<b>84.7%</b>

Table 8.6: **Before-After Results.** Classification results using a linear SVM with different kernels for predicting the before image from the later image using seven different WM regions (listed in Table 8.5). The kernels used are radial basis function (RBF) and lift from Chapter 2.  $\tau$  was fixed for all experiments at 0.7 for FA and 0.65 for MD, and the number of voxels reported is the mean cardinality of the set  $|\Delta\hat{R}_{\text{CONS}}|$  across the different folds in each experiment.

different kernels. Accuracy for each region was determined with 10-fold cross validation; experiments for all three kernels were run for each fold. CONS and voxel selection were re-calculated per fold in order to prevent any information leakage from the test set during training. The 10-fold cross-validation accuracies in predicting “before” scans from “after” scans (i.e. “positive” difference images from “negative” difference images) is shown for different WM regions in Table 8.6. As the table shows, approximately 425 well-chosen voxels are sufficient to achieve a classification accuracy of 93%.

### 8.3.2 Predicting APOE Status

In this experiment we ask the question “Is there a difference in the way that WM changes in subjects with different APOE genotypes?” We answer this question by predicting the APOE  $\epsilon 4$  status (i.e., the presence or absence of this allele) based on the changes in FA values. This experiment is similar to the previous one. Rather than have two sets of positive and negative difference images, we take just one (positive difference images) and group them by the APOE  $\epsilon 4$  status of the subjects they correspond to. We transform these images into point sets and apply a slightly different voxel selection scheme than before: within each group we identify the voxels that exhibit increases and decreases most consistently, and take the union across both groups:

$$\Delta \widehat{R}_{\text{CONS}}(\tau) = \Delta_{\text{ApoE -ve}} \widehat{R}_{\text{CONS}}(\tau) \cup \Delta_{\text{ApoE +ve}} \widehat{R}_{\text{CONS}}(\tau)$$

We used the lift kernel in conjunction with an SVM to differentiate between these two classes of point sets. The baseline accuracy for this experiment is 66.95%, since 79 out of 118 subjects are APOE  $\epsilon 4$  negative. The best cross-validated accuracy of 76% was obtained using the whole body of the corpus

Method	Parameters	Accuracy
Lasso logistic regression Friedman et al. (2010)	$\lambda = .011$	<b>70%</b>
SVM, Lifted kernel	$2D = 500, C = 1$	58%
SVM, Gaussian kernel	$\sigma = 1, C = 1$	57%
Baseline Random Guessing		54%

Table 8.7: Classification results for predicting *Speed and Flexibility* from voxels

callosum, with  $\tau = 0.63$ . The non-spatial RBF kernel was not able to achieve more than baseline accuracy.

### 8.3.3 Predicting Direction of Cognitive Change

In this experiment we ask the question: *Can changes in neuroimaging data predict whether a subject's score for some neuropsychological test has increased or decreased?*

In answering this question we will use data from the same voxels that enabled high-accuracy predictions in the before-after experiment above. To better manage the need for constrained variable selection with wide data, we used the coordinate descent approach for lasso and ridge described in Friedman et al. (2010). Lasso logistic regression via coordinate descent run 100 times with 10-fold cross validation achieved a classification accuracy of 70% with shrinkage parameter  $\lambda = .011$ , which corresponds to the  $\lambda$  within one standard error of the minimum. Results for this and other methods are shown in Table 8.7. No significant improvement was seen for other parameters on competing approaches.

In general, we prefer as few explanatory variables in a model as possible. Wide linear models always raise the specter of overfitting and are notoriously difficult to interpret, particularly when constructed with lasso. Following the spatial point set approach in Chapter 2, we cluster the voxels based on spatial proximity and their  $Q$  values. Simple linkage-based clustering connects voxels with their neighbors if their  $Q$  values are within  $\rho$  percent of each other. We

Method	Parameters	Accuracy
Ridge logistic regression(Friedman et al., 2010)	$\lambda = .013$	<b>75%</b>
SVM, Lifted kernel	$2D = 500, C = 1$	55.7%
SVM, Gaussian kernel	$\sigma = 1, C = 1$	58.5%
Baseline Random Guessing		54%

Table 8.8: Classification results for predicting *Speed and Flexibility* from 30 clusters of voxels

typically take  $\rho = 15$  and specify the maximum number of desired clusters as 30.

Because the clusters are internally consistent with respect to  $Q$  values, we used their mean FA values in a ridge logistic regression analysis to predict the sign of the change in the *Speed and Flexibility* score. Because  $p = 30$  here, which is the number of regions, we are no longer dealing with wide data, alleviating many of the concerns that they raise. While one might imagine the clustering process is lossy, the clusters are better predictors than the voxels used in the previous model. Ridge logistic regression via coordinate descent run 100 times with 10-fold cross validation achieved a classification accuracy of 75% for shrinking parameter  $\lambda = 0.13$ , as chosen above. Results for this and other methods are shown in Table 5.4. No significant improvement was seen for other parameters on competing approaches.

## 8.4 Goodness-of-Fit Tests

Goodness-of-fit tests address the question: *is a given data sample consistent with having been drawn from a specified distribution?* Normality testing is by far the most common goodness-of-fit test (see Section 7.3). This is in large part due to the assumption made in linear regression techniques that the residuals be normally distributed. Our hypothesis test uses a test statistic derived from self anti-similarity distance (self- $d_{AS}$ ). It is determined by the difference between a sample's  $d_{AS}$  and the characteristic number for the source distribution. Our



Sample size(n)	Threshold	
	$\alpha = 0.05$	$\alpha = 0.1$
10	.3209	0.1439
20	.2071	0.0876
30	.1436	0.0671
50	.1122	0.0488
100	.0681	0.0308

Table 8.9: This table provides upper bounds at two significance levels for values of  $AS$  for normality testing of different point set sizes . If  $AS$  lies above the corresponding bound for that point set size, we reject the hypothesis that the sample originated from a normal distribution.

anti-similarity-based normality test statistic ( $AS$ ) for a point set  $P$  is defined as:

$$AS = |d_{AS}(P, P; \ell_1) - \sqrt{2}| + |d_{AS}(P, P; \ell_2^2) - 2| \quad (8.4)$$

$AS$  is used to construct a hypothesis test as follows. We estimate the distribution of  $AS$  under the null hypothesis empirically using samples drawn from normal distributions. We selected two significance levels (0.05 and 0.1) and estimated the thresholds for  $AS$  corresponding to each level for different sample sizes. Given our statistic is lower bounded by zero, we use a one-tailed (upper tail) test. Monte Carlo simulations provided the necessary upper bound for obtaining 95% confidence intervals. If the test statistic lies above this bound, we reject the null hypothesis that it came from a normal distribution. Upper bounds for two significance levels are presented for various sized samples drawn from normal distributions in Table 7.1. Similar tables were constructed for other distributions as well. These tables were used to construct goodness-of-fit hypothesis tests based on  $AS$ . Our approach provides a powerful normality test, which is particularly noticeable on small samples. Table 7.2 provides power comparisons of the  $AS$  test for several different sample sizes at the 5% significance level with the Cramer-von Mises, Watson, Kolmogorov-Smirnov, Anderson-Darling, Neyman-type smooth, Lilliefors, and Shapiro-Wilk (W) tests.

Distribution	n	AS	$W^2$	$U^2$	$D$	$A^2$	$\hat{S}_4$	$D'$	$W$
<i>Uniform</i> (-1, 1)	10	<b>8.2</b>	6.9	8.0	6.1	7.4	2.2	6.1	4.7
	20	14.3	14.8	<b>16.8</b>	8.9	16.2	0.1	9.0	9.9
	30	21.9	21.5	25.5	15.1	<b>27.9</b>	0.0	15.7	21.6
	50	46.7	43.5	49.4	26.2	<b>57.4</b>	7.4	26.2	57.1
	100	80.0	85.3	88.4	62.0	95.2	25.4	61.7	<b>98.2</b>
<i>Exponential</i> (1)	10	<b>45.0</b>	38.0	37.4	31.7	40.8	34.5	31.7	35.6
	20	<b>82.5</b>	72.5	70.6	58.5	78.0	62.8	59.1	75.7
	30	<b>95.6</b>	88.5	84.2	77.1	92.1	82.9	77.4	93.1
	50	<b>99.9</b>	99.2	98.4	95.7	99.7	97.6	95.8	99.8
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
<i>Lognormal</i>	10	<b>60.8</b>	55.7	54.4	46.6	58.0	48.8	46.4	53.5
	20	<b>93.6</b>	87.8	86.8	78.0	90.3	84.6	78.5	89.5
	30	<b>99.5</b>	98.5	97.3	94.6	98.9	95.3	94.7	98.6
	50	<b>100.0</b>	99.9	99.8	99.7	<b>100.0</b>	<b>100.0</b>	99.7	<b>100.0</b>
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
$t(1)$	10	56.7	60.1	<b>60.9</b>	57.8	60.1	57.5	57.8	57.7
	20	83.8	<b>88.3</b>	88.2	85.1	<b>88.3</b>	86.5	85.1	87.1
	30	94.9	95.8	<b>96.0</b>	94.3	95.6	95.1	94.3	95.0
	50	99.5	99.3	99.4	98.8	99.4	<b>100.0</b>	98.8	99.3
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
$\chi_8^2$	10	<b>13.3</b>	12.0	11.7	10.9	<b>13.3</b>	<b>13.3</b>	10.9	11.3
	20	<b>26.9</b>	22.7	20.3	18.6	25.5	25.3	18.8	23.4
	30	<b>44.8</b>	31.5	26.2	24.6	35.9	37.2	24.9	34.8
	50	<b>66.0</b>	51.4	43.3	40.2	57.5	54.0	40.5	59.2
	100	91.2	84.0	75.9	71.1	89.6	89.0	71.1	<b>92.4</b>
<i>Beta</i> (2, 1)	10	<b>13.3</b>	12.5	12.8	11.0	13.1	6.7	10.8	8.8
	20	<b>26.2</b>	20.7	20.2	16.1	22.9	8.1	16.1	15.8
	30	<b>51.7</b>	36.4	34.9	26.2	42.4	7.8	26.5	35.3
	50	<b>75.7</b>	60.7	58.4	43.5	71.7	20.0	43.6	72.9
	100	96.2	92.5	90.1	79.9	98.2	84.4	79.8	<b>99.1</b>
<i>Gamma</i> (1, 2)	10	<b>46.9</b>	37.0	36.8	29.0	39.3	34.0	28.8	33.7
	20	<b>81.2</b>	71.7	68.5	56.6	76.5	62.8	56.8	73.0
	30	<b>95.6</b>	90.3	87.5	78.0	93.5	81.6	78.8	92.6
	50	<b>99.8</b>	99.1	98.4	96.1	99.7	96.5	96.1	<b>99.8</b>
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
<i>Cauchy</i> (0, 1)	10	55.2	62.4	<b>62.6</b>	58.4	62.2	57.9	58.3	58.8
	20	83.6	86.9	86.9	82.9	<b>87.8</b>	86.8	82.9	86.4
	30	95.1	96.1	96.3	93.9	<b>96.5</b>	95.6	94.0	96.2
	50	99.6	<b>99.7</b>	<b>99.7</b>	99.6	<b>99.7</b>	99.6	99.6	99.6
	100	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>

Table 8.10: This table contains the statistical powers Cohen (1988) of our test statistic ( $AS$ ) with the Cramer-von Mises ( $W^2$ ), Watson ( $U^2$ ), Kolmogorov-Smirnov ( $D$ ), Anderson-Darling ( $A^2$ ), Neyman-type smooth ( $\hat{S}_4$ ), Lilliefors ( $D'$ ), and Shapiro-Wilk ( $W$ ) tests. These tests were performed for each given sample size on 1000 instances drawn from each specified distribution, with  $\alpha = 0.05$ . The bolded red values indicate the statistically most powerful test for each sample from a given distribution. All tests have approximately 5% statistical power on normal samples.

## 8.5 Document Classification

In this section we demonstrate how  $S_{IM}$  can be used to solve a document classification problem. The task is to determine which of two newsgroup sources a given document came from. By modeling the *topic* of a document as a *shape*, we use the idea of spatial overlap to model document similarity. We do this by mapping the words in each message to points in a “semantic space” so that similar sets of words (documents) have similar shapes (see Pado and Lapata (2007) for an overview of work on semantic spaces). Messages (seen as collections of words) can then be compared by the similarity  $S_{IM}$  between their point set representation in this space. We will compare the accuracy of  $S_{IM}$  with C4.5 (Quinlan, 1993), random forests (Breiman, 2001), Naive Bayes, and support vector machines (Shawe-Taylor and Cristianini, 2000) for classification of this data set.

### 8.5.1 Semantic Space Construction

The semantic space we will represent documents in is constructed from a set of reference words occurring in documents that have high mutual information with the labels (i.e. the newsgroup). Let  $\mathcal{D}$  be a collection of documents, with each document being a collection of words, and  $\mathcal{V}$  the collection of distinct words occurring in those documents. Between any two words  $w, v \in \mathcal{V}$ , the pointwise mutual information (PMI) between  $w$  and  $v$  is defined as

$$\text{PMI}(w, v) = \log \frac{P(w, v)}{P(w)P(v)},$$

where  $P(w)$  is the probability that word  $w$  occurs in a document and  $P(w, v)$  is the probability that words  $w$  and  $v$  both occur in a document. PMI, in this context, can be thought of as a measure of word similarity; many such measures

have been proposed (Terra and Clarke, 2003), but Terra and Clarke (2003) found PMI to be more effective than numerous competitors.

We fix a set  $\{w_1, \dots, w_p\} \subseteq \mathcal{W}$  of “reference words” having high mutual information with the labels. We then define a map  $f : \mathcal{V} \rightarrow \mathbb{R}^p$  taking each word to the vector of its PMI with respect to each reference word, i.e.,  $f(w) = (\text{PMI}(w, w_1), \dots, \text{PMI}(w, w_p))$ . Every word is thus mapped to a vector consisting of its similarities with each of these reference words; similarity between two words  $w$  and  $v$  being defined by their pointwise mutual information. Words that have similar PMI with the reference words will be located near each other in this “semantic space,” and messages involving similar words will have similar point set shapes. Compared to the most common representation of documents for text classification as “bag of word” (BOW) vectors, this construction has a distinct advantage because it makes use of semantic relations between words.

### 8.5.2 Experiment and Results

We present the results of an experiment on the 20 Newsgroups data set, a collection of UseNet articles compiled by Ken Lang (Lang, 1995). For our experiment, we chose 30 articles at random from each of two newsgroups, alt.atheism and sci.med. We applied simple preprocessing to each article: tokenization, downcasing, stopword and punctuation removal, and removal of words occurring only once in the collection; 2015 distinct words remained. We selected 6 reference words (*christian, doctor, god, medical, say, atheists*) having high expected mutual information with the newsgroup label. To estimate the PMI between words, we recorded the number of hits  $c_w$  and  $c_{w,v}$  reported by Google for each word  $w$  individually and for each pair of words  $(w, v)$ , and we set  $\hat{P}(w, v) = c_{w,v}/N$ ,  $\hat{P}(w) = c_w/N$ , where  $N$  is a normalizing constant (Turney

	Classifier	Accuracy	Precision	Recall	F <sub>1</sub> -Score
Baseline (bag-of-words)	C4.5 (J48)	73.33%	0.763	0.733	0.726
	Naive Bayes	75.00%	0.789	0.750	0.741
	Random forest	78.33%	0.784	0.783	0.783
	SVM (RBF kernel)	76.67%	0.800	0.767	0.760
	SVM (poly. kernel)	83.33%	0.847	0.833	0.832
Semantic space	SVM (Pyramid match)	75.36%	0.742	0.719	0.730
	1-NN (SIM)	85.00%	0.860	0.850	0.849
	2-, 3-, 4-NN (SIM)	85.00%	0.854	0.850	0.850
	5-NN (SIM)	81.67%	0.835	0.817	0.814
	<b>SVM (SIM kernel)</b>	<b>92.75%</b>	<b>0.909</b>	<b>0.938</b>	<b>0.923</b>

Table 8.11: **Document Classification.** This table presents 10-fold cross-validated result metrics on a document classification task. There are 30 documents each in two classes. Each document is represented in a semantic space defined by six representative words. Classification was performed using different learning algorithms such as decision trees, naive Bayes, random forest, and support vector machines with various kernels. Results using SIM are shown in red, the best of which is in bold face.

and Littman, 2005). We estimated

$$\widehat{\text{PMI}}(w, v) = \log \frac{\hat{P}(w, v)}{\hat{P}(w)\hat{P}(v)} = \log \frac{c_{w,v}}{c_w c_v} + \text{const},$$

and set  $\text{const} = 0$  for convenience. Thus, in this experiment, the map from words into the semantic space is

$$\hat{f}(w) = (\widehat{\text{PMI}}(\text{christian}, w), \dots, \widehat{\text{PMI}}(\text{atheists}, w)).$$

We performed classification using  $k$ -nearest neighbors ( $k$ -NN (Cover and Hart, 1967)) and support vector machines (Shawe-Taylor and Cristianini, 2000) using pyramid match kernel and SIM to compare documents. To establish a baseline, we also performed classification over the original indicator bag-of-words vectors. We used the C4.5 (Quinlan, 1993), naive Bayes, random forest (Breiman, 2001), and SVM (with both radial basis function and polynomial kernels) classification algorithms. All baseline experiments were performed

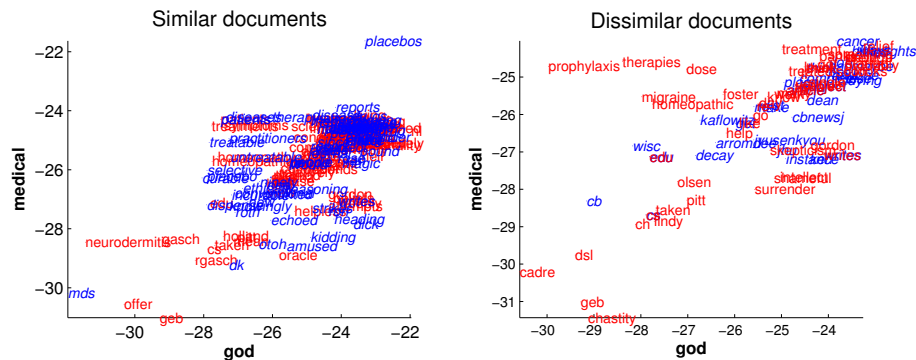


Figure 8.4: In the example above, point sets corresponding to two documents are plotted in the semantic subspace defined by *god* and *medical*. In each plot, one document is displayed in a blue italic font and the other is displayed in a red non-italic font. On the left, the two documents are from the same newsgroups. On the right, the documents are from different newsgroups. *SIM* captures the intuitive notion of spatial overlap corresponding to these classifications. (Note that although the *SIM* computations in semantic space are performed in  $\mathbb{R}^6$ , only two dimensions are visualized here.)

in the Weka Explorer Hall et al. (2009) using default settings. We did not use all common classification algorithms, e.g., decision trees, for the baseline comparisons, as some are not amenable to semantic space representations.

Classification metrics in Table 1 show that *SIM* is able to exploit semantic relationships between words (reflected by their mutual information) to successfully classify samples in this experiment. Average classification accuracy, precision, recall, and  $F_1$ -score are all consistently higher in the experiments using *SIM* than in the baseline bag-of-words experiments. Additionally, *SIM* provides an easy way to visualize and understand the results, something which is uncommon in many classification tasks; an example is shown in Figure 8.4.

## 8.6 Object Classification in Images

In this experiment we compare the performance of three spatially aware point set comparison kernels on an image classification task on a subset of the publicly available ETH-80 data set (Leibe and Schiele, 2003). We use the data set and

experimental setup of Grauman and Darrell (2007). The data set consists of 8 object classes, 10 objects in each class, and 5 views of each object. We performed two experiments, one using SIFT descriptors Lowe (2004) sampled from each image (total of 256 descriptors in 128 dimensions per image) and the other using a dimensionally reduced version of SIFT in 10 dimensions. Following Grauman and Darrell (2007) we train an SVM classifier using a variety of kernels on the following problem: how well can the category of a holdout object be identified after training on the rest of the data including other instances of objects from that category? Validation is performed against all 5 available views of an object that are held out. The baseline accuracy of this experiment is therefore 12.5%. All experiments were conducted using C implementations on a 3.16 GHz machine with 8 GB memory. Classification accuracy results are shown in Tables 8.12 and 8.13. Using all 128 features in the data set, density overlap kernel and SIM were able to achieve accuracies above 93%, corresponding to over 7% and 5% improvements respectively in total cross-validated accuracy over pyramid match kernel (and nearly 4% improvement over the best accuracy over all considered methods cited in Grauman and Darrell (2007)).

## 8.7 Protein Structure Similarity

A fundamental problem in protein structure analysis is determining whether two proteins have similar folded conformations, especially when they have low sequence homology. There are a variety of algorithms to compute an alignment that optimizes some measure of distance between two protein conformations (Singh and Brutlag, 2000). A popular method is to minimize the root mean-squared distance between the 3-D coordinates of each protein's constituent atoms (assuming a correspondence between their backbone carbon atoms) using the Kabsch algorithm (Kabsch, 1976). When correspondences

Kernel	Accuracy	Time
$-d_{KW}$	93.75%	22743s
Density overlap (exact)	93.25%	2271s
Density overlap ( $t = 3$ )	89.25%	774s
Density overlap ( $t = 4$ )	90.5%	1178s
Density overlap ( $t = 5$ )	92.75%	1746s
Pyramid match	86%	407s

Table 8.12: Classification accuracy results on ETH-80 data set. There are a total of 400 images in the data set, consisting of views of objects from eight different classes. In each fold of the experiment, a classifier is tested on five views of a held-out object. The total accuracies using a support vector machine with different learning kernels is shown in this table. All 128 features generated from SIFT are used in this experiment.

Kernel	Accuracy	Time
$-d_{KW}$	89%	21546s
Density overlap (exact)	86.75%	294s
Density overlap ( $t = 3$ )	84%	112s
Density overlap ( $t = 4$ )	85.75%	167s
Density overlap ( $t = 5$ )	86.25%	227s
Pyramid match	81%	108s

Table 8.13: Classification accuracy results on ETH-80 data set. There are a total of 400 images in the data set, consisting of views of objects from eight different classes. In each fold of the experiment, a classifier is tested on five views of a held-out object. The total accuracies using a support vector machine with different learning kernels is shown in this table. The experiment here used 10 features derived via principal components analysis applied on the original 128.

between constituent atoms are unknown the problem is no longer convex and approximate algorithms such as *softassign* can be used (Rangarajan et al., 1997).

We approach this problem by representing protein molecules as weighted point sets of their constituent atoms. Given two proteins, we measure of *structural similarity* by computing  $S_{IM}$  between their point set representations.

The first step in this process is spatially aligning the proteins to compute their  $S_{IM}$  value. We perform this alignment using simulated annealing over gradient descent, guided by the value of  $d_{KW}$  between the two structures. Once the closest structural match has been found, we measure  $S_{IM}$  between



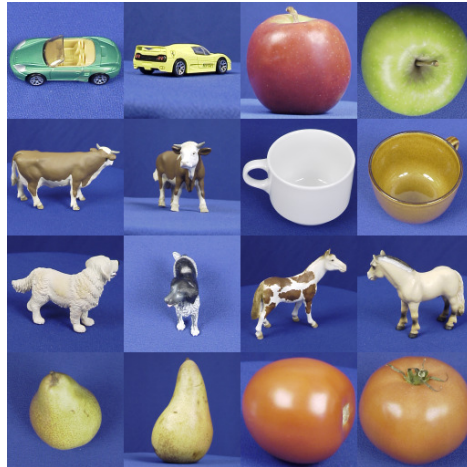


Figure 8.5: Example images from the ETH-80 data set. Two instances from each of the 8 classes are shown.

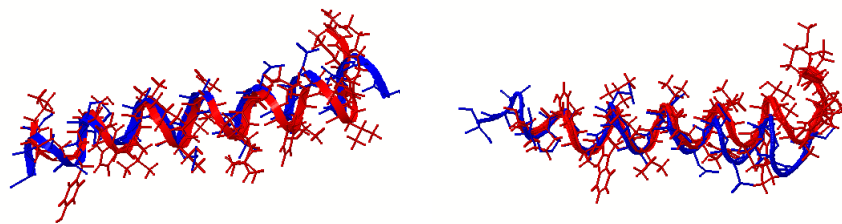


Figure 8.6: The result of aligning the 3-D representations of two proteins 1BCT and 2IFO. The figure on the left displays the alignment obtained by using our method and has a  $S_{IM}$  of 0.814. The figure on the right shows the alignment using Kabsch and has  $S_{IM} = 0.568$ . We only show the first 25 backbone carbons and their residues for effective visualization. The first alignment is “tighter” and thus has a higher  $S_{IM}$ .

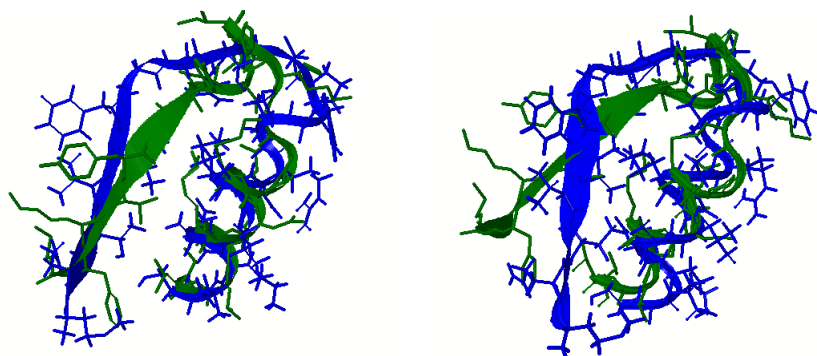


Figure 8.7: The result of aligning the 3-D representations of two proteins 1ABA and 1GRX. The figure on the left displays the alignment obtained by using our method and has a  $S_{IM}$  of 0.764. The figure on the right shows the alignment using Kabsch and has  $S_{IM} = 0.707$ . In the second alignment we see the left ends of the protein chain diverging, leading to a lower value for  $S_{IM}$ . As in Figure 8.6 we restrict visualization to the first 25 units.

the two proteins. In the example shown in Figure 8.6, we align and compare two protein structures with Protein Data Bank (PDB (Bernstein et al., 1978)) IDs '1ABA' and '1GRX'. These two proteins are functionally similar and belong to the Glutaredoxin subgroup; however, they come from different organisms and have different amino acid sequences. We perform the alignment on the backbone atoms (alpha carbons), and then apply the transformation to the residues corresponding to those atoms. The result of this alignment is shown in Figure 8.6. The value of  $S_{IM}$  between the aligned point sets is 0.764, indicating a structural homology. For visualization purposes we only show the first 25 carbon atoms and residues. In contrast, when alignment is performed using the Kabsch algorithm the alignment is not as good (shown in Figure 8.6(b)), indicated by a lower  $S_{IM}$  value of 0.707.

In Figure 8.7, a similar result is shown with proteins '1BCT' and '2IFO.' The result of an alignment based on the Kabsch algorithm is shown in Figure 8.7(b), with a  $S_{IM}$  value of 0.568. With our alignment, the  $S_{IM}$  value is 0.814, corresponding to better overlap, illustrated in Figure 8.7(a). Between non-

similar protein structures  $S_{IM}$  averages at 0.30. This type of analysis can be used to automatically determine remote homologs in a database of protein structures.

We were able to find this surprising result because similarity is determined between entire protein structures; the biologically interesting question is how well do two proteins' folded conformations overlap, as similar structure is often indicative of similar function. The magnitude of  $S_{IM}$  can be seen as a measure of how closely the atoms in one structure mirror those in the other.

## 9 CONCLUSION

---

In this dissertation I have presented a thorough study of the theoretical aspects of the point set representation and its diverse applicability. Point sets enable a lossless representation of data by allowing the inclusion of fuller descriptions of each component within a data instance and by allowing for variable numbers of components within an instance. This representation closely models many real-world knowledge representation needs that benefit from its flexibility. In the remainder of this section I conclude by providing a summary of the previous chapters along with a discussion of my contributions.

### 9.1 Summary

Chapter 2 provides a detailed study of point sets and extant comparison measures. Coen (2006) first introduced the idea of defining a measure between point sets based on spatial overlap; we built on a similar idea and derived another novel measure between point sets called density overlap kernel. The chapter concludes with a comparison of many of these measures on a simple point set classification experiment; spatially-aware measures are shown to be clearly more effective at this task. The proceeding chapters discuss a diverse array of application areas of this paradigm and show the different ways in which these ideas can benefit new and traditional machine learning tasks.

In Chapter 3, we considered the problem of designing a measure to compare the outputs of clustering algorithms that in addition to the cluster assignments also takes into account the spatial locations of differences between them. Our measure `CDISTANCE` is unique in enabling comparisons between clusterings that differ in their data sets, number of points, and number of clusters. We also discussed how `CDISTANCE` may be used to enable stability analysis for clustering algorithms and data sets. Stability is a measure of how robust an algorithm-

data set combination is to perturbations in the input. CDISTANCE is particularly suited for this task because it is able to effectively compare clusterings generated from subsamples of the data set.

We built upon the idea of a spatial comparison of clusters and clusterings in Chapter 4 to introduce a new ensemble clustering algorithm. This algorithm combines information from a diverse set of candidate clusterings (the ensemble) to produce a new “consensus” clustering that combines information from all members of the ensemble. The consensus clustering is generated so that it maximizes a criterion of internal spatial consistency with respect to the individual clusters comprising the ensemble members. One of the main benefits of ensemble methods is that they are able to generate solutions to a task that may lie outside the search spaces of the algorithms generating the ensemble. Experimental validation of this algorithm on a diverse collection of data sets showed that it was able to recover partitions of data sets with lower error than the best-performing member of the ensemble. In these experiments error was measured with respect to a known ground truth partitioning of the data set. In one particularly difficult data set, our algorithm generated a partition that reduced the error of the best-performing individual ensemble member by over 20%.

In Chapter 5 we examined the problem of tracing subtle changes in the brain corresponding to natural aging, risk factor disposition, and cognitive performance of individuals. The goal was to advance current understanding of the progress of neural decay and regeneration in the brain and to find imaging markers for early diagnosis of Alzheimer’s disease. This is an important problem with significant clinical relevance to neuroscience practitioners. Brain imaging techniques yield a three-dimensional view of the brain, assigning measures of interest such as gray matter volume or directionality of diffusion to each voxel. These measures are expected to be consistent in local regions of the

brain. The brain therefore is well-suited to being modelled as a point set. In the experiments conducted in this chapter we were able to identify regions in the brain that are predictive of aging, risk factor status, and cognitive performance. In the process we discovered surprising patterns of *increase* in white matter in certain regions of the brain.

Chapter 6 explores an exciting new property of well-studied probability distributions called characteristic numbers. This property arises from a variant of the transportation distances discussed in Chapter 2. For many families of distributions, characteristic numbers are provably independent of parameters and dimensionality. The quantity used to derive characteristic numbers has a natural discrete counterpart that can be applied to discrete point samples. We have found empirically that over samples this quantity approaches the characteristic number of the source distribution. This insight leads to a new and powerful goodness-of-fit testing framework described and evaluated in Chapter 7. These tests are unique in that they use the coordinates of each point directly, rather than relying on summary statistics such as mean, variance, kurtosis, moments, or skewness. Our normality test performs exceptionally well — especially for small sample sizes — in terms of statistical power (being able to differentiate between samples from distinct probability distributions) with respect to other widely used state-of-the-art techniques.

Chapter 8 summarizes results from previous chapters in one place and presents results from three other experiments on image classification, document classification, and protein similarity detection. In each of these we demonstrated improved results on standard tasks when using spatially aware techniques.

The common thread underlying all the above applications is the idea of using the geometric relationships between data points in order to better inform learning algorithms. There is a significant amount of information contained in these relationships that conventional representations, kernels, and algorithms

are not well-positioned to exploit. I have chosen a diverse set of application domains to demonstrate how spatial analysis can help derive insights and improve learning outcomes such as predictive accuracy in a way that other methods in use today cannot. In most cases above, the best results were obtained not simply by replacing a spatially insensitive kernel with a spatially sensitive one (in most cases, the choice of representation will not permit this), but rather by thinking from the paradigm of spatial locality and determining the best way to incorporate spatial relationships into the learning algorithm. In some cases this may involve a choice between different transportation distances, as in Chapter 3, or a modification to them, as in Chapters 4 and 6.

## 9.2 Contributions

This dissertation contains original contributions that are paradigmatic, theoretical, and application-oriented. At a paradigmatic level, it brings together isolated ideas in optimization theory, knowledge representation, density estimation, and clustering under a common framework and provides a thorough investigation of its theoretical underpinnings. This is a comprehensive work dealing with point set representations of data in multiple domains, containing both a theoretical treatment of the resulting complications and a demonstration of its applicability in a wide variety of domains.

The theoretical contributions of this dissertation include:

- A novel method of measuring the spatial similarity of point sets based on kernels (Section 2.5). The main idea is that discrete point sets can be turned into a continuous distribution by placing Gaussian kernels at each point, and summing up the contributions of each kernel at each point. Two point sets can then be compared by measuring the overlap of these distributions. We showed an analytical way of computing this measure

as well as approximation methods with guaranteed error bounds that cut the running time from  $\Omega(n^2D)$  down to  $O(nD \log(n))$ , where  $D$  is the dimensionality of the feature space.

- A thorough analysis of existing point set to point set similarity measures. I discussed three methods (SIM, density overlap kernel, and lift distance) in detail and analyzed each one's strengths and weaknesses. In addition I also summarized and analyzed other methods such as pyramid match kernel (Grauman and Darrell, 2007) and others.
- A discrete formulation of a point set similarity measure SIM (first proposed as a distance in Coen (2006)). I analyzed its behavior, running time, and define variants that capture different aspects of spatial overlap. I also investigated the efficacy of approximation techniques and how quickly the approximation degrades in different scenarios.
- A novel method of measuring the similarity or dissimilarity between two *clusterings* that takes into account both the partitional differences between them as well as the geometric locations of points of disagreements between them. The measure is smooth and does not have discontinuous jumps at cluster boundaries.
- A complete ensemble clustering algorithm that makes use of the clustering distance measure above.
- A definition of *anti-similarity*, a quantity defined for any discrete point set, and a series of proofs of its asymptotic value for samples from various distributions. This quantity finds application in a powerful goodness-of-fit test.

I have applied the above techniques to frame and solve both standard and new problems in a number of diverse domains:



**Neuroimaging:** Brain scans obtained via magnetic resonance imaging can be seen as a three-dimensional image. The contributions below are an outcome of formulating and solving problems in the context of characterizing patterns of change in populations with elevated risk of Alzheimer’s disease:

- A method to rank voxels according to their information content for a given classification problem.
- A classifier that can be trained to identify the earlier of two scans for a given subject without any temporal information.
- Identification of regions in the brain that increase and decrease in matter with age with large probability.
- Prediction of the presence of a gene type based on neural change and identification of regions in the brain that change maximally differently based on the presence or absence of this gene.
- Prediction of improvement or decline in neuropsychological test scores of a subject purely based on changes in their brain scans.

In addition to the above, I have also discussed applications of the point set representation to problems in the domains of clustering, goodness-of-fit testing, image classification, protein structure similarity determination, and document classification.

### 9.3 Closing Remarks

The goal of this work has been to contribute towards the utility and efficacy of the point set representation which seems perhaps far simpler than other forms of knowledge representation in ML, and yet its lossless fidelity provides advantages difficult to achieve otherwise. It is hoped this humble contribution enables further exploration of spatially aware machine learning techniques.

## REFERENCES

- 
- Adluru, N., C. Hinrichs, M.K. Chung, J.-E. Lee, V. Singh, E.D. Bigler, N. Lange, J.E. Lainhart, and A.L. Alexander. 2009. Classification in DTI using shapes of white matter tracts. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, 2719–2722.
- Alzheimer's, Association. 2015. 2015 Alzheimer's disease Facts and Figures. *Alzheimer's & Dementia: the Journal of the Alzheimer's Association* 11(3):332.
- Alzheimer's Disease Neuroimaging Initiative. 2003. Alzheimer's Disease Neuroimaging Initiative (ADNI): <http://www.adni-info.org>.
- Ambroise, Christophe, and Geoffrey J McLachlan. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences* 99(10):6562–6566.
- Andoni, Alexandr, Khanh Do Ba, Piotr Indyk, and David Woodruff. 2009. Efficient sketches for earth-mover distance, with applications. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, 324–330. IEEE.
- Ansari, M Hidayath, Michael H Coen, Barbara B Bendlin, Mark A Sager, and Sterling C Johnson. 2014. A spatially sensitive kernel to predict cognitive performance from short-term changes in neural structure. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
- Ansari, M Hidayath, Nathanael Fillmore, and Michael H Coen. 2010. Incorporating spatial similarity into ensemble clustering. In *Proceedings of MultiClust: Discovering, Summarizing, and Using Multiple Clusterings: International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Ashford, J. 2004. APOE genotype effects on Alzheimer's disease onset and epidemiology. *Journal of Molecular Neuroscience* 23(3):157–165.
- Asuncion, A., and D.J. Newman. 2007. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/>. [Online; accessed 26-July-2015].
- Bae, Eric, James Bailey, and Guozhu Dong. 2006. Clustering similarity comparison using density profiles. In *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence*, 342–351. Springer LNCS.

Bartzokis, George, Po H Lu, Daniel H Geschwind, Kathleen Tingus, Danny Huang, Mario F Mendez, Nancy Edwards, and Jim Mintz. 2007. Apolipoprotein E affects both myelin breakdown and cognition: implications for age-related trajectories of decline into dementia. *Biological Psychiatry* 62(12):1380–1387.

Basser, P.J., and C. Pierpaoli. 1996. Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI. *Journal of Magnetic Resonance, Series B* 111(3):209–219.

Ben-Hur, A., A. Elisseeff, and I Guyon. 2002. A Stability Based Method for Discovering Structure in Clustered Data. In *Pacific Symposium on Biocomputing*, 6–17.

Bendlin, Barbara B, Michele E Fitzgerald, Michele L Ries, Guofan Xu, Erik K Kastman, Brent W Thiel, Howard A Rowley, Mariana Lazar, Andrew L Alexander, and Sterling C Johnson. 2010a. White matter in aging and cognition: A cross-sectional study of microstructure in adults aged eighteen to eighty-three. *Developmental Neuropsychology* 35(3):257–277.

Bendlin, Barbara B., Michele L. Ries, Elisa Canu, Aparna Sodhi, Mariana Lazar, Andrew L. Alexander, Cynthia M. Carlsson, Mark A. Sager, Sanjay Asthana, and Sterling C. Johnson. 2010b. White matter is altered with parental family history of Alzheimer's disease. *Alzheimer's & Dementia : the Journal of the Alzheimer's Association* 6(5):394–403.

Benitez, Andreana, Els Fieremans, Jens H Jensen, Maria F Falangola, Ali Tabesh, Steven H Ferris, and Joseph A Helpert. 2014. White matter tract integrity metrics reflect the vulnerability of late-myelinating tracts in Alzheimer's disease. *NeuroImage: Clinical* 4:64–71.

Bennett, Ilana J, David J Madden, Chandan J Vaidya, Darlene V Howard, and James H Howard. 2010. Age-related differences in multiple measures of white matter integrity: A diffusion tensor imaging study of healthy aging. *Human Brain Mapping* 31(3):378–390.

Bentley, Jon Louis. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9):509–517.

Berikov, Vladimir. 2014. Weighted ensemble of algorithms for complex data clustering. *Pattern Recognition Letters* 38:99–106.

Bernstein, Frances C, Thomas F Koetzle, Grahame JB Williams, Edgar F Meyer, Michael D Brice, John R Rodgers, Olga Kennard, Takehiko Shimanouchi, and Mitsuo Tasumi. 1978. The Protein Data Bank: a computer-based archival file for macromolecular structures. *Archives of Biochemistry and Biophysics* 185(2): 584–591.

Beyer, Kevin, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is “nearest neighbor” meaningful? In *Database Theory — ICDT '99*, 217–235. Springer.

Birdsill, Alex C, Rebecca L Kosciak, Erin M Jonaitis, Sterling C Johnson, Ozioma C Okonkwo, Bruce P Hermann, Asenath LaRue, Mark A Sager, and Barbara B Bendlin. 2013. Regional white matter hyperintensities: aging, Alzheimer’s disease risk, and cognitive function. *Neurobiology of Aging*.

Botev, Z. I., J. F. Grotowski, and D. P. Kroese. 2010. Kernel density estimation via diffusion. *Annals of Statistics* 38(5):2916–2957.

Bowley, Michael P, Howard Cabral, Douglas L Rosene, and Alan Peters. 2010. Age changes in myelinated nerve fibers of the cingulate bundle and corpus callosum in the rhesus monkey. *Journal of Comparative Neurology* 518(15):3046–3064.

Breiman, Leo. 2001. Random forests. *Machine Learning* 45(1):5–32.

Bron, Esther, Marion Smits, Wiro Niessen, and Stefan Klein. 2015. Feature Selection Based on the SVM Weight Vector for Classification of Dementia. *IEEE Journal of Biomedical and Health Informatics* PP(99):1.

Burges, Christopher JC. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2(2):121–167.

Cai, C. Z., W. L. Wang, L. Z. Sun, and Y. Z. Chen. 2003. Protein function classification via support vector machine approach. *Mathematical Biosciences* 185(2):111–122.

Camastra, Francesco, and Alessandro Verri. 2005. A novel kernel method for clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27(5): 801–805.

Cambanis, S., G. Simons, and W. Stout. 1976. Inequalities for  $E_k(X, Y)$  When the Marginals are Fixed. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiet* 36:285–294.

- Campbell, William M, Douglas E Sturim, and Douglas A Reynolds. 2006. Support vector machines using GMM supervectors for speaker verification. *Signal Processing Letters, IEEE* 13(5):308–311.
- Canu, Elisa, Federica Agosta, Edoardo G Spinelli, Giuseppe Magnani, Alessandra Marcone, Elisa Scola, Monica Falautano, Giancarlo Comi, Andrea Falini, and Massimo Filippi. 2013. White matter microstructural damage in Alzheimer’s disease at different ages of onset. *Neurobiology of Aging* 34(10): 2331–2340.
- Carlesimo, Giovanni A., Andrea Cherubini, Carlo Caltagirone, and Gianfranco Spalletta. 2010. Hippocampal mean diffusivity and memory in healthy elderly individuals. *Neurology* 74(3):194–200.
- Cho, Minkyong, and David M. Mount. 2008. Embedding and similarity search for point sets under translation. In *SCG ’08: Proceedings of the Twenty-Fourth Annual Symposium on Computational Geometry*, 320–327. New York, NY, USA: ACM.
- Chung, M. K., N. Adluru, E. J. Lee, M. Lazar, J. E. Lainhart, and A. L. Alexander. 2010. Cosine series representation of 3D curves and its application to white matter fiber bundles in diffusion tensor imaging. In *Statistics and Its Interface*, vol. 3, 69–80.
- Cleveland, W. S., and R. McGill. 1985. Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science* 229:828–833.
- . 1987. Graphical Perception: The Visual Decoding of Quantitative Information on Statistical Graphs (with Discussion). *Journal of the Royal Statistical Society Series A* 150:192–229.
- Coen, Michael H. 2006. Multimodal dynamics: Self-supervised learning in perceptual and motor systems. Ph.D. thesis, Massachusetts Institute of Technology.
- Coen, Michael H., M. Hidayath Ansari, and Barbara B. Bendlin. 2013. Predicting Short-Term Cognitive Change from Longitudinal Neuroimaging Analysis. In *3rd NIPS Workshop on Machine Learning and Interpretation in Neuroimaging*.
- Coen, Michael H., M. Hidayath Ansari, and Nathanael Fillmore. 2010. Comparing clusterings in space. In *ICML 2010: Proceedings of the 27th International Conference on Machine Learning*.

- . 2011. Learning from spatial overlap. In *AAAI '11: Proceedings of the 25th National Conference on Artificial Intelligence*, 177–182. AAAI Press.
- Coen, Michael H., M. Hidayath Ansari, Marissa Phillips, and Timothy S. Chang. 2012. Goodness-of-fit testing via optimization. In *ORS 2012: Proceedings of the 2nd Annual International Conference on Operations Research and Statistics*.
- Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. 2nd ed. Routledge Academic.
- Corouge, Isabelle, P. Thomas Fletcher, Sarang Joshi, Sylvain Gouttard, and Guido Gerig. 2006. Fiber tract-oriented statistics for quantitative diffusion tensor MRI analysis. *Medical Image Analysis* 10(5):786–798.
- Cour, Timothee, Stella Yu, and Jianbo Shi. 2004. Normalized cut segmentation code. <http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html>. Accessed: 27-July-2015.
- Cover, Thomas M, and Peter E Hart. 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* 13(1):21–27.
- Cuingnet, Rémi, Emilie Gerardin, Jérôme Tessieras, Guillaume Auzias, Stéphane Lehéricy, Marie-Odile Habert, Marie Chupin, Habib Benali, Olivier Colliot, Alzheimer's Disease Neuroimaging Initiative, et al. 2011. Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database. *Neuroimage* 56(2):766–781.
- D'Agostino, R.B., and M.A. Stephens. 1986. *Goodness-of-fit Techniques*. Statistics, textbooks and monographs, M. Dekker.
- Dalal, Navneet, and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 886–893. IEEE.
- Dang, Xuan-Hong, and James Bailey. 2010. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge discovery and data mining*, 573–582. KDD '10, New York, NY, USA: ACM.
- Davatzikos, Christos, Priyanka Bhatt, Leslie M Shaw, Kayhan N Batmanghelich, and John Q Trojanowski. 2011. Prediction of MCI to AD conversion, via MRI, CSF biomarkers, and pattern classification. *Neurobiology of Aging* 32(12): 2322.e19–2322.e27.

- Day, William HE. 1981. The complexity of computing metric distances between partitions. *Mathematical Social Sciences* 1(3):269–287.
- Desikan, Rahul S, Howard J Cabral, Christopher P Hess, William P Dillon, Christine M Glastonbury, Michael W Weiner, Nicholas J Schmansky, Douglas N Greve, David H Salat, Randy L Buckner, et al. 2009. Automated MRI measures identify individuals with mild cognitive impairment and Alzheimer’s disease. *Brain* 132(8):2048–2057.
- Deza, Michel M., and Elena Deza. 2009. *Encyclopedia of Distances*. Springer.
- Di Paola, M, F Di Iulio, A Cherubini, C Blundo, AR Casini, G Sancesario, D Passafiume, C Caltagirone, and G Spalletta. 2010. When, where, and how the corpus callosum changes in MCI and AD. A multimodal MRI study. *Neurology* 74(14):1136–1142.
- Dietterich, Thomas G, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89(1):31–71.
- Do Ba, Khanh, Huy L Nguyen, Huy N Nguyen, and Ronitt Rubinfeld. 2011. Sublinear time algorithms for earth mover’s distance. *Theory of Computing Systems* 48(2):428–442.
- Dongen, Stijn. 2000. Performance criteria for graph clustering and markov cluster experiments. Tech. Rep., National Research Institute for Mathematics and Computer Science in The Netherlands, Amsterdam, The Netherlands.
- Dowling, N Maritza, Bruce Hermann, Asenath La Rue, and Mark A Sager. 2010. Latent structure and factorial invariance of a neuropsychological test battery for the study of preclinical Alzheimer’s disease. *Neuropsychology* 24(6): 742.
- Duda, Richard O, Peter E Hart, and David G Stork. 2012. *Pattern Classification*. John Wiley & Sons.
- Dudoit, Sandrine, and Jane Fridlyand. 2003. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 19(9):1090–1099.
- Dueck, D., and J. Frey, B. 2007. Non-metric affinity propagation for unsupervised image categorization. In *ICCV 2007: IEEE 11th International Conference on Computer Vision*, 1–8. IEEE Computer Society.

Dyrba, M., M. Ewers, M. Wegrzyn, I. Kilimann, C. Plant, A. Oswald, T. Kirste, and S. Teipel et al. 2012. Combining DTI and MRI for the automated detection of Alzheimer's disease using a large European multicenter dataset. In *Multimodal Brain Image Analysis*, vol. 7509 of *Lecture Notes in Computer Science*. Nice, France: Springer Berlin / Heidelberg.

Dyrba, Martin, Michael Ewers, Martin Wegrzyn, Ingo Kilimann, Claudia Plant, Annahita Oswald, Thomas Meindl, Michela Pievani, Arun LW Bokde, Andreas Fellgiebel, et al. 2013. Robust automated detection of microstructural white matter degeneration in Alzheimer's disease using machine learning classification of multicenter DTI data. *PLoS One* 8(5):e64925.

Falahati, Farshad, Eric Westman, and Andrew Simmons. 2014. Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging. *Journal of Alzheimer's disease: JAD* 41(3):685–708.

Farris, James S. 1973. On comparing the shapes of taxonomic trees. *Systematic Biology* 22(1):50–54.

Fern, Xiaoli Z, and Wei Lin. 2008. Cluster ensemble selection. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 1(3):128–141.

Fern, Xiaoli Zhang, and Carla E. Brodley. 2003. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *Proceedings of the 20th International Conference on Machine Learning*, 186–193.

———. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, 36. New York, NY, USA: ACM.

Fields, R. Douglas. 2008. White Matter. *Scientific American* 298(3):54 – 61.

Fisher, R.A. 1918. The correlation between relatives on the supposition of Mendelian inheritance. *Transactions of the Royal Society of Edinburgh* 52:399–433.

Fletcher, P., Ran Tao, Won-Ki Jeong, and Ross Whitaker. 2007. A Volumetric Approach to Quantifying Region-to-Region White Matter Connectivity in Diffusion Tensor MRI. In *Information Processing in Medical Imaging*, ed. Nico Karssemeijer and Boudewijn Lelieveldt, vol. 4584 of *Lecture Notes in Computer Science*, 346–358. Springer Berlin / Heidelberg.



- Folstein, Marshal F, Susan E Folstein, and Paul R McHugh. 1975. "Mini-mental state": A practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research* 12(3):189–198.
- Fowlkes, E.B., and C.L. Mallows. 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78:553–569.
- Franek, Lucas, and Xiaoyi Jiang. 2014. Ensemble clustering by means of clustering embedding in vector spaces. *Pattern Recognition* 47(2):833–842.
- Fred, Ana. 2001. Finding Consistent Clusters in Data Partitions. In *Proceedings of the 3rd International Workshop on Multiple Classifiers*, 309–318. Springer.
- Fred, Ana L. N., and Anil K. Jain. 2002. Data Clustering Using Evidence Accumulation. In *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 4, 276–280.
- Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33(1):1.
- Gelman, A., J.B. Carlin, H.S. Stern, and D.B. Rubin. 2003. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. 2nd ed. Chapman and Hall/CRC.
- Ghaemi, Reza, Md. Nasir Sulaiman, Hamidah Ibrahim, and Norwati Mustapha. 2009. A survey: Clustering ensembles techniques. *World Academy of Science, Engineering, and Technology* 3.
- Gibbs, A. L., and F. E. Su. 2002. On choosing and bounding probability metrics. *International Statistical Review* 7(3):419–435.
- Goodall, Colin. 1991. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)* 53(2):285–339.
- Grauman, Kristen, and Trevor Darrell. 2007. The pyramid match kernel: Efficient learning with sets of features. *The Journal of Machine Learning Research* 8:725–760.
- Gray, Alexander G. 2003. Nonparametric density estimation: toward computational tractability. In *Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining*, 203–211.

- Gray, Katherine R, Paul Aljabar, Rolf A Heckemann, Alexander Hammers, Daniel Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. 2013. Random forest-based similarity measures for multi-modal classification of Alzheimer's disease. *Neuroimage* 65:167–175.
- Grydeland, Håkon, Lars T Westlye, Kristine B Walhovd, and Anders M Fjell. 2013. Improved prediction of Alzheimer's disease with longitudinal white matter/gray matter contrast changes. *Human Brain Mapping* 34(11):2775–2785.
- Haasdonk, Bernard, and Claus Bahlmann. 2004. Learning with distance substitution kernels. In *Pattern Recognition*, 220–227. Springer.
- Hadjitodorov, Stefan T., Ludmila I. Kuncheva, and Ludmila P. Todorova. 2006. Moderate diversity for better cluster ensembles. *Information Fusion* 7(3):264–275.
- Hagmann, Patric, Lisa Jonasson, Philippe Maeder, Jean-Philippe Thiran, Van J Wedeen, and Reto Meuli. 2006. Understanding Diffusion MR Imaging Techniques: From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond. *Radiographics* 26(suppl\_1):S205–S223.
- Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11.
- Haller, Sven, Pascal Missonnier, FR Herrmann, Cristelle Rodriguez, M-P Deiber, Duy Nguyen, Gabriel Gold, K-O Lovblad, and Panteleimon Giannakopoulos. 2013. Individual classification of mild cognitive impairment subtypes by support vector machine analysis of white matter DTI. *American Journal of Neuroradiology* 34(2):283–291.
- Holmes, Michael P., Alexander G. Gray, and Charles Lee Isbell. 2008. Ultrafast Monte Carlo for Kernel Estimators and Generalized Statistical Summations. In *Advances in Neural Information Processing Systems*, vol. 21.
- Huber-Carol, C. 2002. *Goodness-of-fit tests and model validity*. Statistics for Industry and Technology, Birkhäuser.
- Hubert, L., and P. Arabie. 1985. Comparing Partitions. *Journal of Classification* 2:193–218.

- Hubo, Erik, Tom Mertens, Tom Haber, and Philippe Bekaert. 2008. Special Section: Point-Based Graphics: Self-similarity based compression of point set surfaces with application to ray tracing. *Computer Graphics* 32(2):221–234.
- Ishii, Kazunari, Takashi Kawachi, Hiroki Sasaki, Atsushi K Kono, Tetsuya Fukuda, Yoshio Kojima, and Etsuro Mori. 2005. Voxel-based morphometric comparison between early-and late-onset mild Alzheimer’s disease and assessment of diagnostic performance of z score images. *American Journal of Neuroradiology* 26(2):333–340.
- J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. 1983. *Graphical Methods for Data Analysis*. New York: Chapman and Hall.
- Jain, Anil K., and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Jain, Anil K, M Narasimha Murty, and Patrick J Flynn. 1999. Data Clustering: A Review. *ACM Computing Surveys (CSUR)* 31(3):264–323.
- Joachims, Thorsten. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.
- Juneja, Mayank, Andrea Vedaldi, C. V. Jawahar, and Andrew Zisserman. 2013. Blocks that shout: Distinctive parts for scene classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 923–930. IEEE.
- Kabsch, Wolfgang. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32(5):922–923.
- Kantorovich, L. V. 2006. On the translocation of masses. *Journal of Mathematical Sciences* 133(4):1381–1382. (The original paper was published in Dokl. Akad. Nauk SSSR, 37, No 7-8, pp. 227-229).
- Kaufman, Leonard, and Peter J Rousseeuw. 2009. *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. John Wiley & Sons.
- Kerchner, Geoffrey A, Caroline A Racine, Sandra Hale, Reva Wilhelm, Victor Laluz, Bruce L Miller, and Joel H Kramer. 2012. Cognitive processing speed in older adults: Relationship with white matter integrity. *PloS One* 7(11):e50425.

- Khurd, P., S. Baloch, R. Gur, C. Davatzikos, and R. Verma. 2007. Manifold Learning Techniques in Image Analysis of High-dimensional Diffusion Tensor Magnetic Resonance Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007: CVPR '07*, 1–7.
- Kinnunen, Tomi, and Haizhou Li. 2010. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication* 52(1):12–40.
- Klee, V., and G. I. Minty. 1972. How good is the simplex algorithm. In *Inequalities III: Proceedings of the Third Symposium*, 159–175. New York: Academic Press.
- Klöppel, Stefan, Cynthia M Stonnington, Carlton Chu, Bogdan Draganski, Rachael I Scahill, Jonathan D Rohrer, Nick C Fox, Clifford R Jack, John Ashburner, and Richard SJ Frackowiak. 2008. Automatic classification of MR scans in Alzheimer’s disease. *Brain* 131(3):681–689.
- Knuth, D.E. 1969. *The Art of Computer Programming, Volume II: Seminumerical Algorithms*. Addison-Wesley Professional.
- Kolmogorov, A. 1933. Sulla determinazione empirica di una legge di distribuzione. *Giorna. Ist. Attuari.* 4:83–91.
- Kondor, Risi, and Tony Jebara. 2003. A kernel between sets of vectors. In *ICML 2003: Proceedings of the 20th International Conference on Machine Learning*.
- Kononenko, Igor, Edvard Simec, and Marko Robnik-Sikonja. 1997. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence* 7(1):39–55.
- Lang, Ken. 1995. Newsweeder: Learning to filter netnews. In *ICML 1995: Proceedings of the 12th International Conference on Machine Learning*, 331–339.
- Lange, Tilman, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. 2004. Stability-based validation of clustering solutions. *Neural Computation* 16(6): 1299–1323.
- Le Bihan, Denis, Jean-François Mangin, Cyril Poupon, Chris A. Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. 2001. Diffusion Tensor Imaging: Concepts and Applications. *Journal of Magnetic Resonance Imaging* 13(4): 534–546.

- LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.
- Lee, Dongryeol, and Alexander Gray. 2006. Faster Gaussian Summation: Theory and Experiment. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*.
- . 2009. Fast High-dimensional Kernel Summations Using the Monte Carlo Multipole. In *Advances in Neural Information Processing Systems*, vol. 21.
- Lehmann, E.L., and J.P. Romano. 2005. *Testing Statistical Hypotheses*. 3rd ed. Springer Texts in Statistics, New York: Springer.
- Leibe, B., and B. Schiele. 2003. Analyzing appearance and contour based methods for object categorization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*, vol. 2, 409–415.
- Levina, E., and P. Bickel. 2001. The Earth Mover’s Distance is the Mallows Distance: Some insights from statistics. *IEEE International Conference on Computer Vision* 2:251–256.
- Li, Shi. 2010. On constant factor approximation for earth mover distance over doubling metrics. *arXiv abs/1002.4034*.
- Lindemer, Emily R, David H Salat, Eric E Smith, Khoa Nguyen, Bruce Fischl, Douglas N Greve, Alzheimer’s Disease Neuroimaging Initiative, et al. 2015. White Matter Signal Abnormality Quality Differentiates MCI that Converts to Alzheimer’s Disease from Non-converters. *Neurobiology of Aging*.
- Liu, Cheng-Lin, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. 2003. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition* 36(10):2271–2285.
- Liu, Huan, and Hiroshi Motoda. 1998. *Feature selection for knowledge discovery and data mining*. Norwell, MA, USA: Kluwer Academic Publishers.
- Lövdén, M., N. C. Bodammer, S. Kühn, Jörn Kaufmann, H. Schütze, C. Tempelmann, Hans-Jochen Heinze, E. Düzel, F. Schmiedek, and U. Lindenberger. 2010. Experience-dependent plasticity of white-matter microstructure extends into old age. *Neuropsychologia* 48(13):3878–3883.

- Lowe, David G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.
- MacQueen, James, et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297. Oakland, CA, USA.
- Madden, David J, Ilana J Bennett, Agnieszka Burzynska, Guy G Potter, Nankuei Chen, and Allen W Song. 2012. Diffusion tensor imaging of cerebral white matter integrity in cognitive aging. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease* 1822(3):386–400.
- Magnin, Benoît, Lilia Mesrob, Serge Kinkingnéhun, Mélanie Péligrini-Issac, Olivier Colliot, Marie Sarazin, Bruno Dubois, Stéphane Lehericy, and Habib Benali. 2009. Support vector machine-based classification of Alzheimer’s disease from whole-brain anatomical MRI. *Neuroradiology* 51(2):73–83.
- Meila, Marina. 2005. Comparing clusterings: An axiomatic view. In *ICML 2005: Proceedings of the 22nd International Conference on Machine Learning*, 577–584. New York, NY, USA: ACM.
- Mitchell, T. M. 1980. The need for biases in learning generalizations. Tech. Rep. CBM-TR-117, Rutgers University, New Brunswick, NJ.
- Mitchell, Thomas M. 1997. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc.
- Monge, G. 1781. Mémoire sur la théorie des déblais et de remblais. *Mém. Math. Phys. Acad. Royale Sci.* 666–704.
- Mori, S. 2007. *Introduction to Diffusion Tensor Imaging*. Elsevier Science.
- Mukhopadhyay, N., J.K. Ghosh, and James O. Berger. 2005. Some Bayesian predictive approaches to model selection. *Statistics & Probability Letters* 73(4): 369–379.
- Munkres, James. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5(1):32–38.
- Munkres, James R. 2000. *Topology*. 2nd ed. Prentice Hall.

- Mwangi, Benson, Khader M Hasan, and Jair C Soares. 2013. Prediction of individual subject's age across the human lifespan using diffusion tensor imaging: A machine learning approach. *Neuroimage* 75:58–67.
- Neyman, J. 1937. Smooth tests for goodness of fit. *Skand. Aktuar.* 20:150–199.
- Neyman, J., and E.S. Pearson. 1933. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A* 231:289–337.
- Ng, Andrew Y, Michael I Jordan, Yair Weiss, et al. 2002. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* 2:849–856.
- Nguyen, Nam, and Rich Caruana. 2007. Consensus clusterings. In *Seventh IEEE International Conference on Data Mining: ICDM 2007*, 607–612. IEEE.
- Nister, David, and Henrik Stewenius. 2006. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2161–2168. IEEE.
- Odone, Francesca, Annalisa Barla, and Alessandro Verri. 2005. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing* 14(2):169–180.
- Oishi, K., K. Zilles, K. Amunts, and S. Mori et al. 2008. Human brain white matter atlas: Identification and assignment of common anatomical structures in superficial white matter. *Neuroimage* 43(3):447 – 457.
- Osada, Robert, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2001. Matching 3D Models with Shape Distributions. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, 154. Washington, DC, USA: IEEE Computer Society.
- Pado, Sebastian, and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* 33(2):161–199.
- Pearson, K. 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can reasonably be supposed to have arisen from random sampling. *Philosophical Magazine* 5(50):157–175.

- Pele, O., and M. Werman. 2009. Fast and robust earth mover's distances. In *IEEE 12th International Conference on Computer Vision, 2009*, 460–467.
- Philbin, James, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition: CVPR '07*, 1–8. IEEE.
- Phillips, Jeff M., Parasaran Raman, and Suresh Venkatasubramanian. 2011. Generating a Diverse Set of High-Quality Clusterings. In *2nd Workshop on Discovering, Summarizing, and Using Multiple Clusterings (MultiClust 2011)*.
- Pontil, Massimiliano, and Alessandro Verri. 1998. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(6):637–646.
- Pordes, R., D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Wurthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick. 2008. The open science grid status and architecture. *Journal of Physics: Conference Series* 119.
- Prince, Martin, Renata Bryce, and Cleusa Ferri. 2011. World Alzheimer's Report. Tech. Rep., Alzheimer's Disease International.
- Purves, D. 2012. *Neuroscience*. Sinauer Associates.
- Quesenberry, C.P., and F.L. Miller. 1977. Power studies of tests for uniformity. *Communications in Statistics — Simulation and Computation* 5:169–191.
- Quinlan, J Ross. 1993. C4.5: Programming for Machine Learning. *Morgan Kauffmann*.
- Raghavan, V. V. 1982. Approaches for measuring the stability of clustering methods. *SIGIR Forum* 17(1):6–20.
- Rahimi, Ali, and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems* 20:1177–1184.
- Raman, Parasaran, Jeff M. Phillips, and Suresh Venkatasubramanian. 2011. Spatially-aware comparison and consensus for clusterings. In *Proceedings of SIAM International Conference on Data Mining (SDM)*.



- Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 846–850.
- Rangarajan, Anand, Haili Chui, and Fred L. Bookstein. 1997. The Softassign Procrustes Matching Algorithm. In *IPMI '97: Proceedings of the 15th International Conference on Information Processing in Medical Imaging*, 29–42. London, UK: Springer-Verlag.
- Rayner, J. C. W., and D. J. Best. 1986. Neyman-type smooth tests for location-scale families. *Biometrika* 73(2):437–446.
- Rayner, O., J.C.W. Thas, and D.J. Best. 2009. *Smooth Tests of Goodness of Fit: Using R*. Hoboken: John Wiley and Sons Inc.
- Reitan, Ralph M, and Deborah Wolfson. 2009. The Halstead–Reitan Neuropsychological Test Battery for Adults—Theoretical, Methodological, and Validational Bases. *Neuropsychological Assessment of Neuropsychiatric and Neuromedical Disorders* 3–24.
- Roobaert, Danny, and Marc M Van Hulle. 1999. View-based 3D object recognition with support vector machines. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, 77–84. IEEE.
- Ropper, Allan H, et al. 2009. *Adams and Victor's principles of neurology*. McGraw-Hill Medical New York.
- Rosenbloom, Margaret, Edith V Sullivan, and Adolf Pfefferbaum. 2003. Using magnetic resonance imaging and diffusion tensor imaging to assess brain damage in alcoholics. *Alcohol Research and Health* 27(2):146–152.
- Roses, A. D., M. W. Lutz, H. Amrine-Madsen, A. M. Saunders, D. G. Crenshaw, S. S. Sundseth, M. J. Huentelman, K. A. Welsh-Bohmer, and E. M. Reiman. 2010. A TOMM40 variable-length polymorphism predicts the age of late-onset Alzheimer's disease. *The Pharmacogenomics Journal* 10(5):375–384.
- Rubin, D. B. 1984. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics* 12(4):1151–1172.
- Rubner, Y., C. Tomasi, and L. Guibas. 2000. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2):99–121.

- Sager, Mark A, Bruce Hermann, and Asenath La Rue. 2005. Middle-aged children of persons with Alzheimer's disease: APOE genotypes and cognitive function in the Wisconsin Registry for Alzheimer's Prevention. *Journal of Geriatric Psychiatry and Neurology* 18(4):245–249.
- Schnack, Hugo G, Mireille Nieuwenhuis, Neeltje EM van Haren, Lucija Abramovic, Thomas W Scheewe, Rachel M Brouwer, Hilleke E Hulshoff Pol, and René S Kahn. 2014. Can structural MRI aid in clinical classification? A machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects. *Neuroimage* 84:299–306.
- Scholz, Jan, Miriam C Klein, Timothy EJ Behrens, and Heidi Johansen-Berg. 2009. Training induces changes in white-matter architecture. *Nature Neuroscience* 12(11):1370–1371.
- Shamir, Ohad, and Naftali Tishby. 2010. Stability and model selection in k-means clustering. *Machine Learning* 80(2-3):213–243.
- Shapiro, S. S., and M. B. Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52(3/4):pp. 591–611.
- Shawe-Taylor, John, and Nello Cristianini. 2000. Support vector machines. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* 93–112.
- Shi, Jianbo, and J. Malik. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(8):888 –905.
- Singh, Amit P, and Douglas L Brutlag. 2000. Protein structure alignment: A comparison of methods. *Bioinformatics*.
- Smirnov, N.V. 1948. Tables for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics* 19(2):279–281.
- Smith, S. M., M. Jenkinson, H. Johansen-Berg, D. Rueckert, T. E. Nichols, C. E. Mackay, K. E. Watkins, O. Ciccarelli, M. Z. Cader, P. M. Matthews, and T. E.J. Behrens. 2006. Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. *Neuroimage* 31(4):1487 – 1505.
- Sokal, Robert R, and F James Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxonomy* 33–40.

- Sperling, Reisa A, Paul S Aisen, Laurel A Beckett, David A Bennett, Suzanne Craft, Anne M Fagan, Takeshi Iwatsubo, Clifford R Jack, Jeffrey Kaye, Thomas J Montine, et al. 2011. Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimer's & Dementia* 7(3):280–292.
- Stephens, M.A. 1974. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association* 69(347):730–737.
- Stolcke, A., S. Kajarekar, and L. Ferrer. 2008. Nonparametric feature normalization for SVM-based speaker verification. In *ICASSP 2008: IEEE International Conference on Acoustics, Speech and Signal Processing*, 1577–1580.
- Strehl, Alexander, and Joydeep Ghosh. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.
- Sun, Zhuo, Yong Fan, Boudewijn PF Lelieveldt, and Martijn van de Giessen. 2015. Detection of Alzheimer's disease using group lasso SVM-based region selection. In *Proceedings of SPIE 9414, Medical Imaging 2015: Computer-Aided Diagnosis*. International Society for Optics and Photonics.
- Szummer, Martin, and Rosalind W Picard. 1998. Indoor-outdoor image classification. In *IEEE International Workshop on Content-Based Access of Image and Video Database, 1998*, 42–51.
- Terra, Egidio, and Charles LA Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 165–172. Association for Computational Linguistics.
- Thas, O. 2009. *Comparing Distributions*. 1st ed. Springer.
- Thode, H.C. 2002. *Testing for Normality*. 1st ed. New York: Marcel Dekker, Inc.
- Topchy, A., A.K. Jain, and W. Punch. 2005. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(12):1866–1881.

- Topchy, Alexander, Anil K. Jain, and William Punch. 2004. A mixture model for clustering ensembles. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 379–390.
- Trener, Max R, B Crosson, J DeBoe, and WR Leber. 1989. *Stroop Neuropsychological Screening Test Manual*. Psychological Assessment Resources.
- Tukey, J.W. 1977. *Exploratory Data Analysis*. Addison-Wesley.
- Turney, Peter D., and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning* 60(1–3):251–278.
- Vega-Pons, Sandro, and José Ruiz-Shulcloper. 2011. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* 25(03):337–372.
- Vemuri, Prashanthi, Jeffrey L Gunter, Matthew L Senjem, Jennifer L Whitwell, Kejal Kantarci, David S Knopman, Bradley F Boeve, Ronald C Petersen, and Clifford R Jack. 2008. Alzheimer’s disease diagnosis in individual subjects using structural MR images: validation studies. *Neuroimage* 39(3):1186–1197.
- Vinh, Nguyen Xuan, and Julien Epps. 2009. A Novel Approach for Automatic Number of Clusters Detection in Microarray Data Based on Consensus Clustering. In *Proceedings of the 2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering*, 84–91. BIBE ’09, Washington, DC, USA: IEEE Computer Society.
- Wasserman, Larry. 2013. *All of Statistics: A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer Science & Business Media.
- Wasserstein, L. N. 1969. Markov processes over denumerable products of spaces describing large systems of automata. *Probl. Inform. Transmission* 5: 47–52.
- Watanabe, S. 1969. *Knowing and guessing: A quantitative study of inference and information*. New York: Wiley.
- Wilk, M. B., and R. Gnanadesikan. 1968. Probability plotting methods for the analysis of data. *Biometrika* 55(1):1–17.
- Wolpert, D. H. 1996. The existence of a priori distinctions between learning algorithms. *Neural Computation* 8:1341–1390.

Zelnik-Manor, Lihi, and Pietro Perona. 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, 1601–1608.

Zhou, D., J. Li, and H. Zha. 2005. A new Mallows distance based metric for comparing clusterings. In *ICML 2005: Proceedings of the 22nd International Conference on Machine Learning*, 1028–1035. New York, NY, USA: ACM.

Ziegler, David A, Olivier Piguet, David H Salat, Keyma Prince, Emily Connally, and Suzanne Corkin. 2010. Cognition in healthy aging is related to regional white matter integrity, but not cortical thickness. *Neurobiology of Aging* 31(11): 1912–1926.

## A CHARACTERISTIC NUMBERS USING $\ell_2^2$

---

In Section 6.3 we computed values of  $d_{AS}$  for univariate normal, uniform, and exponential distributions using  $\ell_1$  ground distance. This appendix contains the derivations of  $d_{AS}$  values for the same distributions using the non-metric ground distance  $\ell_2^2$ . Table 6.2 shows  $d_{AS}$  for both these ground distances.

### A.1 Uniform Distributions

Let  $P$  be a uniform distribution between  $a$  and  $b$  ( $a, b \in \mathbb{R}$ ) with probability density function  $f(x) = \frac{1}{b-a}$  for  $x$  between  $a$  and  $b$  and 0 otherwise. Let  $m = \frac{b+a}{2}$ . To calculate  $d_{AS}$  we must first compute  $d_{AT}$  and  $d_{NT}$ . The ground distance in this and following section is  $d(x, y) = (x - y)^2$ . Anti-transportation distance for  $P$  can be computed as follows:

$$\begin{aligned}
 d_{AT}(P, P) &= \int_a^b 4(x - m)^2 f(x) dx \\
 &= \int_a^b \frac{4(x - m)^2}{b - a} dx \\
 &= \frac{4}{b - a} \int_a^b (x^2 + m^2 - 2mx) dx \\
 &= \frac{4}{3(b - a)} ((b - a)(a^2 + ab + b^2) + 3m^2(b - a) - 3m(b - a)(b + a)) \\
 &= \frac{4}{3} (a^2 + ab + b^2 + 3m^2 - 3m(b + a)) \\
 &= \frac{4}{3} (a^2 + ab + b^2 - 3m^2) \quad (\text{since } a + b = 2m) \\
 &= \frac{1}{3} (4(a^2 + ab + b^2) - 3(a + b)^2) \\
 &= \frac{1}{3} (a^2 - 2ab + b^2) \\
 &= \frac{(b - a)^2}{3}
 \end{aligned} \tag{A.1}$$

Naive distance:

$$\begin{aligned}
 d_{NT}(P, P) &= \int_a^b \int_a^b (x - y)^2 f(x) f(y) dy dx \\
 &= \frac{1}{(b - a)^2} \int_a^b \int_a^b (x - y)^2 dy dx
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{(b-a)^2} \int_a^b \left( x^2 \int_a^b dy + \int_a^b y^2 dy - x \int_a^b 2y dy \right) dx \\
&= \frac{1}{3(b-a)^2} \int_a^b \left( 3x^2(b-a) + (b-a)(a^2 + ab + b^2) \right. \\
&\quad \left. - 3x(b-a)(a+b) \right) dx \\
&= \frac{1}{6(b-a)} \int_a^b (6x^2 + 2(a^2 + ab + b^2) - 6x(a+b)) dx \\
&= \frac{1}{6(b-a)} \left( 2(b-a)(a^2 + ab + b^2) + 2(b-a)(a^2 + ab + b^2) \right. \\
&\quad \left. - 3(b-a)(a+b)^2 \right) \\
&= \frac{1}{6} (4(a^2 + ab + b^2) - 3(a+b)^2) \\
&= \frac{1}{6} (a^2 - 2ab + b^2) \\
&= \frac{(b-a)^2}{6} \tag{A.2}
\end{aligned}$$

From Equations A.1 and A.2 we have

$$d_{AS}(P, P) = \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{(b-a)^2/3}{(b-a)^2/6} = 2$$

**The self-anti-similarity distance of any univariate uniform distribution using the  $\ell_2^2$  ground distance is therefore 2.**

## A.2 Normal Distributions

Let  $P$  be a normal distribution with mean  $\mu$  and variance  $\sigma^2$  and probability density function  $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  for  $x \in \mathbb{R}$ . The anti-transportation distance for this normal distribution using  $\ell_2^2$  ground distance can be computed

as:

$$\begin{aligned}
 d_{AT}(P, P) &= \int_{-\infty}^{\infty} (2(x - \mu))^2 f(x) dx \\
 &= \int_{-\infty}^{\infty} \frac{4(x - \mu)^2}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\
 &= \frac{4\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y^2 e^{-\frac{y^2}{2}} dy \quad \text{substituting } y = \frac{x - \mu}{\sigma} \tag{A.3}
 \end{aligned}$$

$$= \frac{4\sigma^2}{\sqrt{2\pi}} \left[ \left[ -ye^{-\frac{y^2}{2}} \right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} -e^{-\frac{y^2}{2}} dy \right] \quad (\text{integrating by parts})$$

$$= \frac{4\sigma^2}{\sqrt{2\pi}} \left[ 0 + \sqrt{2\pi} \right] \tag{A.4}$$

$$= 4\sigma^2 \tag{A.5}$$

Naive distance for this normal distribution:

$$\begin{aligned}
 d_{NT}(P, P) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - y)^2 f(x) f(y) dx dy \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{(x - y)^2}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2 + (y-\mu)^2}{2\sigma^2}} dx dy \\
 &= \frac{\sigma^2}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x' - y')^2 e^{-\frac{x'^2 + y'^2}{2}} dx' dy' \\
 &\quad (\text{substituting } x' = \frac{x - \mu}{\sigma}, y' = \frac{y - \mu}{\sigma})
 \end{aligned}$$

Expanding the square and noting that  $xe^{-\frac{x^2}{2}}$  is an odd function, with an integral of 0 between  $-\infty$  and  $\infty$  we have

$$\begin{aligned}
 d_{NT}(P, P) &= \frac{\sigma^2}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x'^2 + y'^2) e^{-\frac{x'^2 + y'^2}{2}} dx' dy' \\
 &= \frac{\sigma^2}{2\pi} \left[ \int_{-\infty}^{\infty} x'^2 e^{-\frac{x'^2}{2}} dx' \int_{-\infty}^{\infty} e^{-\frac{y'^2}{2}} dy' \right. \\
 &\quad \left. + \int_{-\infty}^{\infty} y'^2 e^{-\frac{y'^2}{2}} dy' \int_{-\infty}^{\infty} e^{-\frac{x'^2}{2}} dx' \right] \\
 &= \frac{\sigma^2}{2\pi} \left[ \sqrt{2\pi} \int_{-\infty}^{\infty} x'^2 e^{-\frac{x'^2}{2}} dx' + \sqrt{2\pi} \int_{-\infty}^{\infty} y'^2 e^{-\frac{y'^2}{2}} dy' \right] \\
 &= \frac{\sigma^2}{2\pi} [2\pi + 2\pi] \quad (\text{using the result from A.3 and A.4}) \\
 &= 2\sigma^2 \tag{A.6}
 \end{aligned}$$



From Equations A.5 and A.6 we have

$$d_{AS}(P, P) = d \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{4\sigma^2}{2\sigma^2} = 2$$

**The self-anti-similarity distance of any univariate normal distribution using the  $\ell_2^2$  ground distance, regardless of its mean and variance, is therefore 2.**

### A.3 Exponential Distributions

Let  $P$  be an exponential distribution with rate parameter  $\lambda$  and probability density function  $f(x) = \lambda e^{-\lambda x}$  for  $x \geq 0$ . Since  $P$  is not symmetric, we need its cumulative distribution function  $F(x) = 1 - e^{-\lambda x}$  as well. The anti-transportation distance for this exponential distribution using  $\ell_2^2$  ground distance can be computed as:

$$\begin{aligned} d_{AT}(P, P) &= \int_0^\infty (x - F^{-1}(1 - F(x)))^2 f(x) dx \\ &= \int_0^\infty \lambda \left(x + \frac{\log(1 - e^{-\lambda x})}{\lambda}\right)^2 e^{-\lambda x} dx \\ &= \frac{1}{\lambda^2} \int_0^\infty (z + \log(1 - e^{-z}))^2 e^{-z} dz \\ &= \frac{1}{\lambda^2} \left[ \int_0^\infty z^2 e^{-z} dz + \int_0^\infty 2z \log(1 - e^{-z}) e^{-z} dz \right. \\ &\quad \left. + \int_0^\infty \log^2(1 - e^{-z}) e^{-z} dz \right] \\ &= \frac{1}{\lambda^2} \left[ 2 + 2 \int_0^1 (-\log(1 - y)) \log(y) dy + \int_0^1 \log^2(y) dy \right] \\ &\text{(substituting } y = 1 - e^{-z}\text{)} \\ &= \frac{1}{\lambda^2} \left[ 2 - 2 \int_0^1 \log(1 - y) \log(y) dy + \int_0^1 w^2 e^w dw \right] \\ &\text{(substituting } w = \log(y)\text{)} \\ &= \frac{1}{\lambda^2} \left[ 2 - 2 \int_0^1 \log(1 - y) \log(y) dy + 2 \right] \end{aligned} \tag{A.7}$$

Now consider

$$\begin{aligned}
\int_0^1 \log(1-y) \log(y) dy &= [y \log(y) \log(1-y)]_0^1 \\
&\quad - \int_0^1 y \left( \frac{-\log(y)}{1-y} + \frac{\log(1-y)}{y} \right) dy \text{ (by parts)} \\
&= 0 - \int_0^1 \left( \frac{((1-y)-1) \log(y)}{1-y} + \log(1-y) \right) dy \\
&= - \int_0^1 \log(1-y) dy - \int_0^1 \log(y) dy + \int_0^1 \frac{\log(y)}{1-y} dy \\
&= -2 \int_0^1 \log(y) dy + \int_0^1 \frac{\log(1-y)}{y} dy \\
&= 2 + \int_0^1 \frac{\log(1-y)}{y} dy \tag{A.8}
\end{aligned}$$

Using the Taylor expansion of  $\log(1-y)$  between 0 and 1 we have

$$\begin{aligned}
\int_0^1 \frac{\log(1-y)}{y} dy &= \int_0^1 -\frac{1}{y} \sum_{k=1}^{\infty} \frac{y^k}{k} dy \\
&= - \sum_{k=1}^{\infty} \int_0^1 \frac{y^{k-1}}{k} dy
\end{aligned}$$

Since each term is finite, positive, and integrable

$$\begin{aligned}
&= - \sum_{k=1}^{\infty} \left[ \frac{y^k}{k^2} \right]_0^1 = - \sum_{k=1}^{\infty} \frac{1}{k^2} \\
&= -\pi^2/6 \tag{A.9}
\end{aligned}$$

so that finally, from Equations A.7, A.8, and A.9

$$\begin{aligned}
d_{\text{AT}}(P, P) &= \frac{1}{\lambda^2} [2 - 2(2 - \pi^2/6) + 2] \\
&= \frac{\pi^2}{3\lambda^2} \tag{A.10}
\end{aligned}$$

Naive distance for this exponential distribution:

$$\begin{aligned}
d_{\text{NT}}(P, P) &= \int_0^{\infty} \int_0^{\infty} (x-y)^2 f(x) f(y) dx dy \\
&= \int_0^{\infty} \int_0^{\infty} \lambda^2 (x-y)^2 e^{-\lambda(x+y)} dx dy
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\lambda^2} \int_0^\infty \int_0^\infty (x' - y')^2 e^{-(x'+y')} dx' dy' \text{ where } x' = \lambda x, y' = \lambda y \\
\lambda^2 d_{NT}(P) &= \int_0^\infty \int_0^\infty x'^2 e^{-(x'+y')} dx' dy' - \int_0^\infty \int_0^\infty 2x'y' e^{-(x'+y')} dx' dy' + \\
&\quad \int_0^\infty \int_0^\infty y'^2 e^{-(x'+y')} dx' dy' \\
&= \int_0^\infty x'^2 e^{-x'} dx' \int_0^\infty e^{-y'} dy' - 2 \int_0^\infty x' e^{-x'} dx' \int_0^\infty y' e^{-y'} dy' + \\
&\quad \int_0^\infty e^{-x'} dx' \int_0^\infty y'^2 e^{-y'} dy' \\
&= (2)(1) - 2(1)(1) + (1)(2) \text{ (integrating by parts)} \\
&= 2 \tag{A.11}
\end{aligned}$$

From Equations A.10 and A.11 we have

$$d_{AS}(P, P) = \frac{d_{AT}(P, P)}{d_{NT}(P, P)} = \frac{\pi^2/3\lambda^2}{2/\lambda^2} = \pi^2/6 \approx 1.645.$$

**The self-anti-similarity distance of any exponential distribution using the  $\ell_2^2$  ground distance, regardless of its rate parameter, is therefore  $\pi^2/6$ .**