Mitigating the Risks of Thresholdless Metrics in Machine Learning Evaluation

by

Kendrick Boyd

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

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2014

Date of final oral examination: August 1, 2014

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

C. David Page Jr., Professor, Biostatistics and Medical Informatics
Mark Craven, Professor, Biostatistics and Medical Informatics
Jeffrey Naughton, Professor, Computer Sciences
Jude Shavlik, Professor, Computer Sciences
Xiaojin Zhu, Associate Professor, Computer Sciences

To my family and the future, wherever it takes us.

ACKNOWLEDGMENTS

As always, there have been numerous people who have inspired and helped me in myriad ways during my time in graduate school.

First, I want to thank to my mentor and advisor, David Page, without whom none of this would have been possible. David's boundless enthusiasm and optimism combine perfectly with his intellectual pursuit of deep questions to make him an excellent advisor, teacher, coauthor, and friend.

I wish to thank my thesis committee members: Mark Craven, Jeffrey Naughton, Jude Shavlik, and Xiaojin Zhu. Besides taking informative and thought-provoking classes at various times from all of them, their insights and piercing questions have vastly improved my research and the breadth and depth of my knowledge.

I am grateful to my colleagues and coauthors: Aubrey Barnard, Debbie Chasman, Vitor Santos Costa, Jesse Davis, Kevin Eng, Finn Kuusisto, Eric Lantz, Jie Liu, Deborah Muganda, Jeremy Weiss, and whoever else I am forgetting. From writing papers together, invigorating conversations on machine learning and more, visiting Europe with Finn and Jeremy, and trivia on Monday's at Union South, they have all contributed to making my time in graduate school both productive and memorable.

I want to thank all of the musicians that I have played bassoon with around Madison over the past six years. In particular, to Mindy Taranto and Larry Bevic, for inviting me to be part of the Middleton Community Orchestra and providing me with so many opportunities to create beautiful music.

Thanks to Debbie Chasman, Mike Dartt, Lauren Meyer, Sarah Morton, Deborah Muganda, and Gina Wentling for reading my dissertation, providing many helpful suggestions, and trying to keep my run-on sentences and comma abuse from getting out of control. The remaining paragraph-length sentences, misapplied commas, and mistakes are entirely my fault.

I wish to recognize the University of Wisconsin Computer Sciences Department Alumni Scholarship, the Computation and Informatics in Biology and Medicine Training Program through NIH grant 5T15LM007359, NIGMS grant R01GM097618, and NLM grant R01LM011028-01 for providing funding during graduate school.

And last, but by no means least, I am eternally thankful to my family for their love and support. To Scott and Laura, thanks for nurturing an intellectual curiousity in me in my youth that has ultimately led to this dissertation so many years later. I am especially grateful for the many evening skype conversations with Lewis, Scott, and Laura that have helped keep me (mostly) sane during graduate school. And to my more distant relatives, from farm meetings in Indiana to skiing in the Rockies (acute compartment syndrome included), thanks for all the wonderful experiences in our times, though always too short, together.

My heartfelt thanks and gratitude to all who have touched my life.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This dissertation concerns machine learning evaluation: the process of assessing an algorithm's or model's performance on a test set. Despite being an integral part of machine learning, evaluation is often overlooked and taken for granted. Through investigation of the privacy and properties of thresholdless evaluation methods, this dissertation provides a better understanding of thresholdless measures and demonstrates the dangers of improper application of evaluation methods.

Precision-recall (PR) curves provide an assessment of a scoring model over a range of decision thresholds. PR curves are often preferred over the more well-known ROC curves in highly skewed tasks. We prove that not all points in PR space are achievable. Thus, there is a region that a PR curve cannot go through. The fact that this region changes depending on the test set leads to several important considerations and potential pitfalls for machine learning practitioners, which are discussed.

An additional concern when performing PR analysis is precisely how the PR curve or the area under it is calculated. A number of methods to calculate point estimates and confidence intervals of the area under the PR curve exist in the literature, but there has been minimal investigation into their performance. This dissertation includes an extensive empirical evaluation of these existing methods. The results suggest that average precision, lower trapezoid, and interpolated median are the most robust point estimates. For confidence intervals, the commonly used cross-validation and bootstrap approaches do not provide the advertised coverage for small data sets and should be used with caution. We show that easily calculable parametric confidence intervals do provide the guaranteed coverage.

Differential privacy provides powerful guarantees that individuals incur minimal additional risk by including their personal data in a database. Existing work has focused on producing differentially private models,

counts, and histograms. Nevertheless, even with a differentially private model, directly reporting the model's performance on a database has the potential for disclosure. Thus, differentially private computation of evaluation metrics for machine learning is an important research area. This dissertation presents effective mechanisms for releasing area under the ROC curve and average precision.

1 INTRODUCTION

Evaluating learning algorithms is a critical aspect of machine learning. The machine learning community is largely focused on prediction, so evaluating a model on a separate test set where the correct predictions are known is the gold standard for assessing algorithm performance. The test sets used are as large as possible, given the available resources. Sizes range from tens or hundreds to millions of examples. Therefore, some process of summarizing performance on the test set is required. These summaries to evaluate a model take many forms, from root-mean-square error for real-valued predictions to accuracy and ROC curves for classification tasks.

We will focus on binary classification, where the task is to discriminate between two categories. These categories or labels are often referred to as positive and negative. If a model outputs one of the two labels as the prediction, known as a *dichotomous* output or model, a simple summary for evaluating a model's performance is accuracy: the proportion of the predicted labels that match the true labels. While an attractive approach due to its simplicity, using accuracy alone suffers from several drawbacks (Provost et al., 1998).

One of the drawbacks of accuracy is that it makes no distinction between *true positives* and *true negatives* or between *false positives* and *false negatives* (defined in Table 2.1). While true positives and true negatives both describe a correct prediction, obtaining the correct prediction may be more or less important for one category compared to the other. Similarly, the type of misclassification may be relevant. A false negative may be much worse for a particular task than a false positive would be. Accuracy only makes a distinction between correct and incorrect predictions and ignores the further divisions of false positive versus false negative and true positive versus true negative. So in tasks where the mislabeling costs (cost of false positive compared to cost of false negative) are different, accuracy

is not an ideal measure.

Since the distinction between false positives and false negatives can be critical, another common way to summarize a model's performance is by how many false positives, false negatives, true positives, and true negatives a model predicts for a test set. These counts are typically presented in a *confusion matrix* (also known as a contingency table or, for binary classification in particular, a 2×2 table). The confusion matrix layout used in this document is given in Table 2.2. A confusion matrix provides a compact summary of a model's predictions and contains sufficient information to calculate many other evaluation measures, e.g., accuracy, precision, recall, false positive rate (defined in Table 2.3).

A confusion matrix gives all the information required for most analyses of dichotomous outputs. However, many models assign a probability that each example is of a particular class (often the positive class). More generally, a *scoring* model simply outputs a real number, with larger values indicating the example is more likely to be of a particular class. If analyses using accuracy or other measures derived from a confusion matrix are desired, not only must such a model be learned, but a threshold for the decision boundary must also be selected. There are numerous methods to choose a threshold (Elkan, 2001; Hernández-Orallo et al., 2013), but here we are primarily interested in a group of evaluation techniques that do not require the selection of a specific threshold.

Instead, these techniques provide a summary of an algorithm's performance over a range of possible thresholds. We call this type of analysis *thresholdless* to distinguish it from dichotomous analysis of a single confusion matrix. The most well-known thresholdless method is ROC analysis (Provost et al., 1997; Pepe, 2004; Fawcett, 2006), but there are other techniques that analyze many thresholds simultaneously, e.g., precision-recall curves (Raghavan et al., 1989; Manning and Schütze, 1999), lift curves

(Piatetsky-Shapiro and Masand, 1999; Giudici, 2003), cost curves (Drummond and Holte, 2006), Brier curves (Ferri et al., 2011). While these thresholdless measures are often preferred over accuracy (Provost et al., 1998), their use accrues additional risks that many people are not aware of, including mistaken intuitions about results across different tasks, combining multiple results, and high variance and bias of certain estimates.

The aforementioned metrics are always calculated by applying a model to some test set of labeled data. It has long been known that machine learning models can reveal information about the data used to train them. In the extreme case, a nearest neighbor model might store the data set itself, but more subtle disclosures occur with all types of models. Even small changes in the training set can produce detectable changes in the model. This fact has motivated work to preserve the privacy of the training set by making it difficult for an adversary to discern information about the training data. One popular framework is differential privacy (Dwork, 2006), which sets bounds on the amount of change that can occur when any one training data set row is modified.

Several authors have modified existing machine learning algorithms such that the models satisfy differential privacy (Chaudhuri and Monteleoni, 2008; Friedman and Schuster, 2010; Zhang et al., 2012). In doing so, the models can be released to the public, and the privacy risk to the owners of the rows in the database is tightly bounded, even if the adversary has auxiliary information. However, these protections only cover the training data set, not any latter uses of the model on other data sets.

Consider a scenario in which multiple hospitals are collaborating to predict disease onset but are prevented by policy or law from sharing their data with one another. They may instead attempt to produce a model using data from one institution and test the model at other sites in order to evaluate how well the model generalizes. The institution generating the model might use a differentially private algorithm to create the model in

order to protect their own patients and then distribute the model to the other hospitals. These hospitals in turn run the model on their patients and produce an evaluation of the model's performance, such as the area under the ROC curve (AUCROC). The test data sets at the latter institutions are not covered by any privacy protection that might have been used during training. The problem remains even if the training and test data sets exist at the same institution. While releasing an evaluation metric may seem to be a limited potential privacy breach, it has been demonstrated that data about patients can be reconstructed from ROC curves if the adversary has access to a subset of the test data (Matthews and Harel, 2013).

Thus, an additional risk (though it is more general than just thresholdless evaluation) is the potential leakage of private information, even through the summarization of a thresholdless evaluation method. This dissertation presents, discusses, and makes proposals to address these risks of mistaken intuitions about results, high variance and bias, and privacy for thresholdless metrics.

1.1 Thesis Statement

Evaluating models is an integral aspect of machine learning that is too often taken for granted. Of particularly wide use in machine learning are thresholdless methods such as ROC curves, areas under the ROC curve (AUCROC), precision-recall (PR) curves, areas under the PR curve (AUCPR), the closely related mean average precision (MAP), and error bounds on all of these. This dissertation provides evidence for the following thesis: *Not all methods of generating thresholdless metrics are created equal, and potential pitfalls and benefits accrue based on which methods are chosen.* Specific contributions that follow from the evidence provided include:

- The existence of an unachievable region in PR space that varies with class skew, which implies that AUCPR and MAP estimates and

comparisons of these for different methods should take into account class skew.

- Some widely-used methods for computing confidence bounds on AUCPR and MAP are substantially better than others in a way not previously recognized.
- Publication of AUCROC, AUCPR, and other related metrics computed on private data can violate privacy under a precise, widely-used definition, but algorithms exist to add noise in a way that maintains utility of the estimates while providing guaranteed privacy protection.

1.2 Contributions

While there have been several papers characterizing PR curves (Davis and Goadrich, 2006; Goadrich et al., 2006; Clémençon and Vayatis, 2009), we expand the theoretical and empirical understanding of PR space and curves in Chapters 3 and 4. Chapter 3 concerns the unachievable region of PR space.

- We prove theorems about the location and size of the unachievable region.
- We propose AUCNPR - a modification of AUCPR to account for the unachievable region.
- We discuss the impact of the unachievable region on cross-validation, aggregation across multiple tasks, downsampling, and F_β score.

In Chapter 4, we investigate methods of estimating AUCPR and providing confidence intervals.

- We perform an extensive empirical analysis of the performance of different point estimates and confidence interval methods for AUCPR on simulated data.
- We find that the AUCPR estimators behave quite differently and recommend lower trapezoid, average precision, or interpolated median as three estimators with reasonable performance.
- We find that the commonly used cross-validation and bootstrap approaches to confidence intervals are not satisfactory on small data sets and recommend using binomial or logit intervals instead.

In the final section, we turn to the question of protecting the privacy of the test set in Chapter 5.

- We discuss the need for differential privacy in evaluation, not just for training or data set release.
- We describe algorithms for differentially private AUCROC and average precision.
- We show that these algorithms provide both utility and privacy through experiments on two real-world data sets.

2 BACKGROUND

We review the foundations of evaluation of model's with dichotomous outputs in Section 2.1. Moving to thresholdless measures, we describe and discuss related work on ROC analysis in Section 2.2 and the closely related PR analysis in Section 2.3. Finally, we provide an overview of differential privacy in Section 2.4.

2.1 Confusion Matrices and Related Metrics

This work focuses on evaluation for a binary classification task on a test set with N total examples. We refer to the two classes of examples as *positive* and *negative*, where the positive class often represents the item of interest. For example, in an information retrieval task, the relevant documents would be labeled positive and the irrelevant ones labeled negative. An important property of a test set is the skew, denoted by π, which is the proportion of positive examples. Following Bamber (1975), we denote the number of positive examples by n (not to be confused with the total number of examples N) and the number of negative examples by m. Thus, $n + m = N$ and $\pi = \frac{n}{N}$.

If a model outputs one of two possible values, we say the model has *dichotomous* outputs. With two possible predicted values, the predicted and actual labels can combine in four ways, shown in Table 2.1. The number of occurrences of each type can be compactly described in a confusion matrix (also known as a contingency table or 2×2 table), as in Table 2.2. A variety of performance measures for dichotomous models can be calculated from a confusion matrix. Such measures used in this document, as well as some others included for completeness, are defined in Table 2.3.

Dichotomous outputs, for which a model must make a hard choice between positive and negative, are the simplest type of output for a binary

Table 2.1: The four possible results of a model with dichotomous outputs for a binary classification task.

Predicted	Actual	Name	Variable
Positive	Positive	True Positive	tp
Negative	Positive	False Negative	fn
Positive	Negative	False Positive	fp
Negative	Negative	True Negative	tn

Table 2.2: Confusion matrix: a concise presentation of the number of true positives, false positives, false negatives, and true negatives of a model on some data set. We use the row to denote the prediction and the column to denote the actual label.

	Actual	
Predicted	Positive	Negative
Positive	tp	fp
Negative	fn	tn
Total	$\mathfrak{n}$	$\mathfrak{m}$

classification task. However, most models, including logistic regression, SVMs, and Bayesian networks, internally calculate a score or probability for each example. Then a decision threshold is used to predict positive if the score is larger than the threshold and negative if the score is smaller. Since the choice of decision threshold has an enormous impact on the confusion matrix produced, in the next section we turn to ROC analysis. ROC analysis is a thresholdless evaluation method that does not require a threshold to be chosen and instead simultaneously evaluates a model over all possible thresholds.

Table 2.3: Definitions of several machine learning measures that can be calculated from a confusion matrix.

Name	Formula	Description
Accuracy	$\frac{tp+tn}{tp+tn+fp+fn}$	Proportion of all examples correctly labeled
True Positive Rate TPR Sensitivity Recall	$r = \frac{tp}{tp+fn}$	Proportion of positive examples correctly labeled positive
False Positive Rate FPR 1 - Specificity	$\frac{fp}{fp+tn}$	Proportion of negative examples incorrectly labeled positive
Specificity 1 - FPR	$\frac{tn}{fp+tn}$	Proportion of negative examples correctly labeled negative
Precision Positive Predictive Value PPV	$p = \frac{tp}{tp+fp}$	Proportion of positively labeled examples that are actually positive
Negative Prediction Value NPV	$\frac{tn}{tn+fn}$	Proportion of negatively labeled examples that are actually negative
F_1 score	$\frac{2pr}{p+r}$	Harmonic mean of precision and recall
F_β score	$\frac{(1+\beta^2)pr}{\beta^2 p+r}$	Weighted combination of precision and recall (generalization of F_1 score)

2.2 ROC Analysis

Receiver operating characteristic (ROC) curves were originally developed for signal detection theory in the 1940s. Most famously, ROC curves were used to evaluate the ability of radar receivers to detect enemy aircraft during World War II (Lobo et al., 2008). Subsequently, ROC curves have been used in a variety of fields, including psychophysics (Green and Swets, 1966), evaluation of medical diagnostic tests (Swets and Pickett, 1982; Pepe, 2004), and machine learning (Provost et al., 1998; Flach, 2003; Fawcett, 2006).

ROC analysis summarizes a model's performance using the true positive rate, also known as sensitivity, and false positive rate, equivalent to 1 - specificity (Fawcett, 2006). These two measures are frequently visualized in ROC space. ROC space is the unit square ($[0,1] \times [0,1]$) with false positive rate on the x-axis and true positive rate on the y-axis. The performance of a model with dichotomous outputs can be visualized in ROC space as a point, (x, y), where x is the false positive rate and y is the true positive rate (defined in Table 2.3). Some notable points in ROC space include $(0,0)$, where all examples are labeled negative; $(1,1)$, where all examples are labeled positive; and the ideal point, $(0,1)$, where all examples are correctly labeled. In general, a point is better the closer it is to the ideal point at $(0,1)$, i.e., the higher and farther left it is.

ROC Curves

As we move from a model with dichotomous outputs to one with ordered or real-valued outputs, we can create many different dichotomous models by choosing different thresholds for splitting the ordered outputs into two sets, one labeled positive and the other labeled negative. By connecting points from adjacent thresholds with a line, we obtain the ROC curve for the model. Note that the thresholds that label all examples positive or all

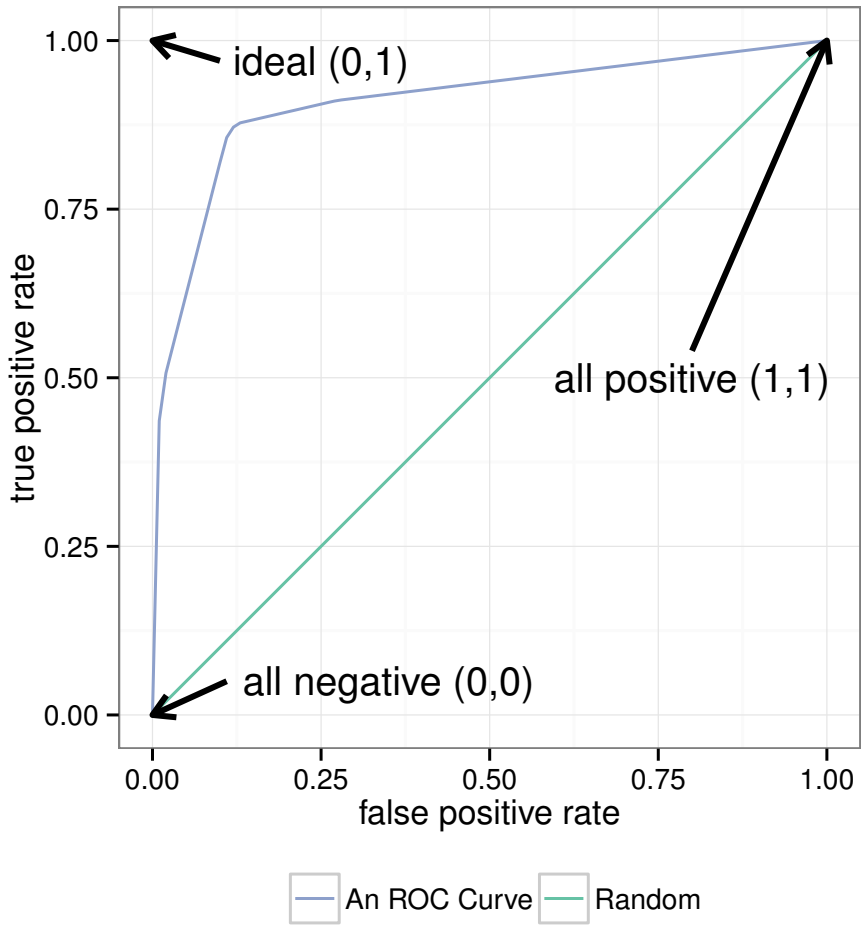


Figure 2.1: A sample ROC curve and the random guessing curve.

examples negative are always possible, so an ROC curve should always start at $(0,0)$ and end at $(1,1)$. See Figure 2.1 for a sample ROC curve with annotations of the notable points in ROC space.

ROC curves have a variety of properties that make them attractive for evaluating binary classification. A model that randomly guesses (e.g., by

outputting a random value between 0 and 1 for each example) has an expected ROC curve of a diagonal line with $y = x$. An ROC curve below the line $y = x$ indicates a model that is worse than random guessing. Thus, ROC curves are typically above the diagonal, with the ideal curve going from $(0, 0)$ to $(0, 1)$ to $(1, 1)$.

ROC curves are insensitive to the prevalence of positive examples, denoted by $\pi = \frac{n}{n+m}$ and often referred to as class skew in machine learning. Changing the ratio of positive to negative examples does not change the true or false positive rates. This can be a particularly useful property when evaluating medical tests where the prevalence in the test data, due to the sampling from case-control studies, may not match the true prevalence.

Another critical property of ROC space is that it grants the ability to linearly interpolate between two points. If two points, A and B, in ROC space are achieved by two models, then any point on the line between A and B can be achieved by a model that randomly chooses between the output of A and B with the appropriate probability (Bamber, 1975). This validates using the convex hull of an ROC curve or set of points in ROC space as the maximum achievable ROC curve (Provost and Fawcett, 2001; Davis and Goadrich, 2006; Fawcett and Niculescu-Mizil, 2007).

Area Under the ROC Curve

The area under the ROC curve is often used as a summary measure for an ROC curve (Bamber, 1975; Hanley and McNeil, 1982; Pepe, 2000, 2004; Fawcett, 2006). Area under the ROC curve is traditionally abbreviated with AUC, but we will use AUCROC to distinguish the area under the ROC curve from areas under other curves. AUCROC ranges from 0 for the worst possible model to 1 for the ideal model. Random guessing has an expected AUCROC of 0.5.

AUCROC is a well-studied quantity in statistics. It can be estimated

using parametric assumptions on the ROC curve (Metz and Kronman, 1980; Swets and Pickett, 1982) or nonparametrically with the Mann-Whitney U-statistic (Hanley and McNeil, 1982). Additionally, there are hypothesis tests for determining if two AUCROCs are significantly different using the DeLong method (DeLong et al., 1988). Bamber (1975) demonstrates an intriguing equivalence that underlies the relationship with the Mann-Whitney U-statistic: AUCROC is equal to the probability that a model will correctly order a randomly drawn positive and randomly drawn negative example. Therefore, if the random variable X denotes the scores of positive examples and Y denotes the scores of negative examples, then

$$\text{AUCROC} = P(X > Y) + \frac{1}{2}P(X = Y).$$

This characterization of AUCROC is binomial if there are no ties (such as when X and Y have continuous distributions and thus $P(X = Y) = 0$) and justifies the use of t-tests for comparing AUCROCs because, for large sample sizes, AUCROC is approximately normally distributed (Bamber, 1975).

The relationship between AUCROC and the Mann-Whitney U-statistic also provides a simple plug-in estimator for calculating AUCROC:

$$\text{AUCROC} = \frac{1}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbb{1}[x_i < y_j] \tag{2.1}$$

where x_i for $1 \leqslant i \leqslant m$ are the scores on the negative examples in the test set and y_j for $1 \leqslant j \leqslant n$ are the scores on the positive examples. Note that neither the x_is nor the y_js must be ordered.

ROC Analysis Drawbacks

ROC analysis is ubiquitous and well-understood; however, when one class is rare, it can be misleading. The true positive rate and false positive rate characterize performance on actual positive and actual negative examples separately. They compare true positives to false negatives and false positives to true negatives, but do not make other critical comparisons such as between true positives and false positives. In many applications, a comparison between true positives and false positives may be even more important than the true positive rate or the false positive rate.

For example, in a medical diagnosis task such as predicting if a mammogram contains a malignant tumor or not, the precision and negative predictive value (defined in Table 2.3) may be the most relevant measures for a patient. Given that a patient has a mammogram that is predicted to be malignant, she is most concerned about the likelihood that it is, in fact, malignant. This is exactly what precision captures: the probability of an actual positive given a positive label. When positives are scarce, as is the case in mammography because malignant tumors are fortunately rare, good performance on true positive rate and false positive rate do not necessarily lead to good precision. When the positive class is rare, even with a high true positive rate and a low false positive rate, the number of false positives can still be much larger than the number of true positives. This leads to low precision despite good results from ROC analysis. A confusion matrix illustrating this phenomenon is given in Table 2.4. Despite obtaining a false positive rate of just 0.1 and a true positive rate of 0.9, the precision is only 0.08 because positive examples are rare ($\pi = 0.01$). Therefore, other thresholdless methods are still of interest, particularly for evaluating tasks with low prevalence.

Table 2.4: Confusion matrix for a highly skewed data set with $\pi = 0.01$. Using ROC analysis, this confusion matrix looks very good, with a false positive rate of 0.1 and a true positive rate of 0.9. However, precision is only 0.08, so the probability that a positively labeled example is actually positive is only 0.08.

	Actual	
Predicted	Positive	Negative
Positive	90	990
Negative	10	8910
Total	100	9900

2.3 PR Analysis

Precision-recall (PR) analysis is similar to ROC analysis, but it uses precision and recall for the axes instead of the true and false positive rates. PR space is defined by the unit square with recall on the x-axis and precision on the y-axis. As in ROC space, a confusion matrix maps to a single point in PR space (with some corner cases when precision is undefined). A model with ordered outputs produces a set of points that can be connected to create a PR curve. The proper method of connecting two points in PR space, however, is not linear interpolation.

Linear interpolation in PR space leads to overly optimistic PR curves (Goadrich et al., 2006; Davis and Goadrich, 2006). To obtain the correct interpolation in PR space, Davis and Goadrich (2006) noted that points in PR space can be mapped to ROC space. The interpolation in PR space can therefore be done by mapping to ROC space, performing a linear interpolation there, and then mapping back to PR space. This produces a nonlinear interpolation in PR space. A critical aspect of the mapping between PR space and ROC space is that the class skew must be known. For

a particular ROC curve, the corresponding PR curve changes depending on the class skew. This nonlinear interpolation in PR space is investigated further in Section 4.3.

Similar to ROC space, several properties of PR space are known. The expected random guessing PR curve is a horizontal line with $y = \pi$. The ideal point in PR space is (1,1) and a model that always assigns a positive label obtains the point (1,π). See Figure 2.2 for a sample PR curve with annotations on notable points in PR space. A model that labels everything negative is a bit problematic because precision is undefined when recall is 0. Loosely speaking, labeling everything negative can be thought of as the point (0,1): no recall but perfect precision. However, in practice, the precision at low recall is highly variable. At low recall, only a few examples are labeled positive, and the precision depends heavily on the exact number of false positives. Small perturbations in the data set, such as removing an example that is predicted as a false positive, can greatly change the precision. At the extreme, if the example with the largest score is positive, then the PR curve starts at $(\frac{1}{n}, 1)$. But if that example is a negative, then the PR curve starts at $(\frac{1}{n}, 0)$. This variability in the PR curve at low recall leads some users of PR analysis to focus only on high recall (Davis et al., 2005).

Finally, as with ROC curves, the area under the PR curve (AUCPR) is often used as a summary statistic. For example, information retrieval (IR) systems are frequently judged by their mean average precision, which is closely related to the mean AUCPR over the queries (Manning et al., 2008). Similarly, AUCPR often serves as an evaluation criterion for machine learning approaches that are typically applied to highly-skewed data, such as statistical relational learning (Kok and Domingos, 2010; Davis et al., 2005; Sutskever et al., 2009; Mihalkova and Mooney, 2007) and information extraction (Ling and Weld, 2010; Goadrich et al., 2006). Some algorithms, such as SVM-MAP (Yue et al., 2007) and SAYU (Davis et al., 2005), explicitly

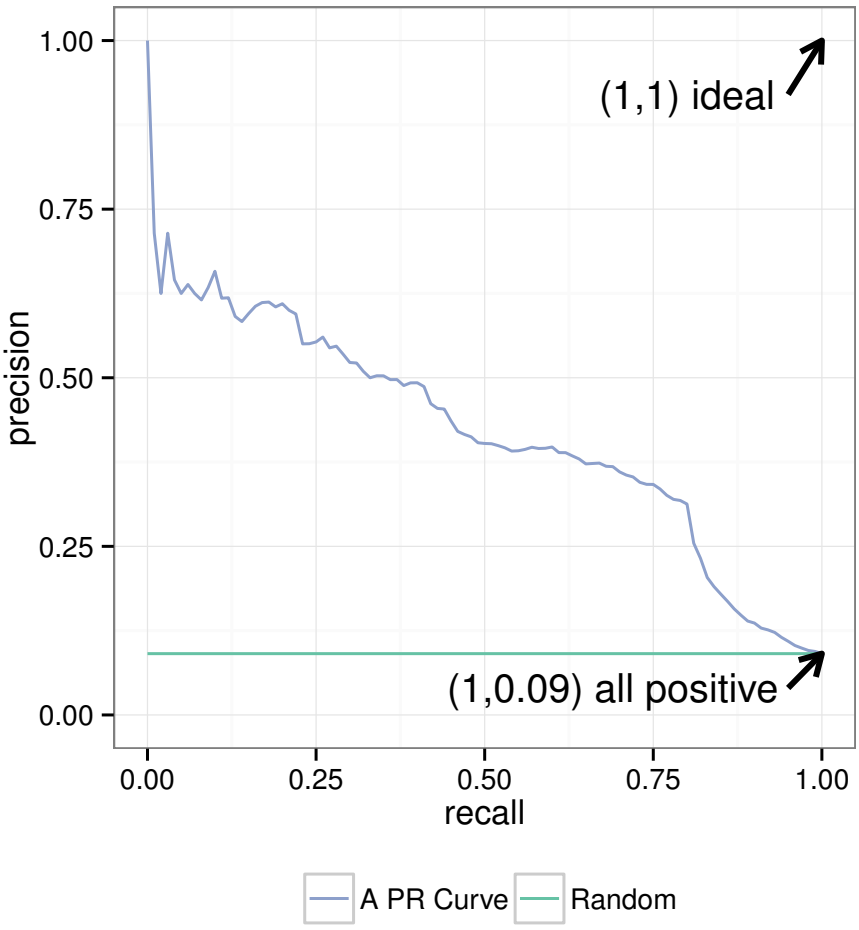


Figure 2.2: A sample PR curve and the random guessing curve for a data set with 10 negatives for every positive example ($\pi = 0.09$).

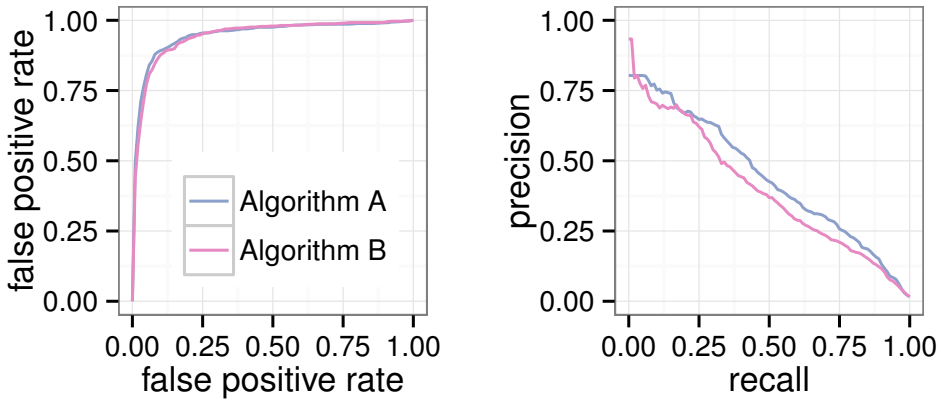
optimize the AUCPR of the learned model.

While sensitivity to class skew may be seen as a drawback to PR analysis, this sensitivity can highlight differences between models in highly skewed data sets that are not as apparent in ROC analysis. A small change

in an already low false positive rate has a minimal impact on ROC analysis: the performance looks good regardless. But in a highly skewed task, a small change in false positive rate can substantially change precision. Figure 2.3 illustrates this using results from two algorithms on the same test set. This test set, and the train set used to learn the models, is highly skewed with $\pi \approx 0.01$. In ROC space, the algorithms are nearly identical, but the difference is more pronounced in PR space. Furthermore, a hypothesis test does not find a statistically significant difference between the AUCROCs for algorithm A and B ($p = 0.30$). But for AUCPR, a statistically significant difference is found with $p < 0.01$ (details are presented in Table 2.5). Thus, PR analysis is often preferred to ROC analysis when there is a large skew in the class distribution (Manning and Schütze, 1999; Bunescu et al., 2005; Davis and Goadrich, 2006). A variety of machine learning applications exhibit a large skew. In information retrieval, only a few documents are relevant to a given query. In medical diagnoses, only a small proportion of the population has a specific disease at any given time. In relational learning, only a small fraction of the possible groundings of a relation are true in a database. PR analysis is increasingly relevant for machine learning as work in these highly skewed data sets continues to grow.

2.4 Differential Privacy

Differential privacy is a framework that guarantees that the presence or absence of an individual's information in the database has little effect on the output of an algorithm. Thus, an adversary can learn limited information about any individual. More precisely, for any databases $D, D' \in \mathbb{D}$, let $d(D, D')$ be the number of rows that differ between the two databases. Differential privacy requires that the probability an algorithm outputs the same result on any pair of neighboring databases (those with $d(D, D') = 1$)



(a) ROC curves

(b) PR curves with $\pi \approx 0.01$

Figure 2.3: ROC curves and PR curves for two algorithms on the same test set. The slight improvement of algorithm A over algorithm B seen in the ROC curves is more pronounced in the PR curves. The PR curve also shows that much room remains for improvement. These curves are drawn from experiments on a medical data set where the task is to predict if a patient will develop breast cancer within one year. The PR curve explicitly shows that the threshold at which 90% of the malignant tumors are identified ($x = 0.9$) provides only about 10% precision. That is, roughly 90% of the patients receiving a malignant diagnosis will not actually develop breast cancer in the next year!

is bounded by a constant ratio. There is not a consensus amongst the differential privacy literature as to whether "neighboring" databases means adding or removing a row, or just changing a row. This leads Kifer and Machanavajjhala (2011) to distinguish between bounded differential privacy, where a neighboring database is obtained by changing the value of exactly one row, and unbounded differential privacy, where a neighboring

Table 2.5: AUCROC for the ROC curves in Figure 2.3(a) and AUCPR for recall above 0.5 for the PR curves in Figure 2.3(b). The p-value is the result of performing a two-tailed, paired t-test on the cross-validated folds to test for statistically significant differences in AUCROC and AUCPR. Even though the PR and ROC curves are derived from the same two algorithms on the same test set, a statistically significant difference is detected in AUCPR but not in AUCROC.

Algorithm	AUCROC	AUCPR ($r > 0.5$)
A	0.948	0.123
B	0.944	0.101
p-value	0.30	< 0.01

database is obtained by adding or removing a row. In this dissertation, we use the bounded differential privacy definition and henceforth will refer to it simply as differential privacy, defined in Definition 2.1. Thus, $\mathbb{D}$ refers to the set of all databases with the same number of rows, N.[1]

Definition 2.1 (ϵ-differential privacy (Dwork, 2006; Kifer and Machanavajjhala, 2011))**.** *For any input database* D*, a randomized algorithm* $f : \mathbb{D} \to \mathtt{Range}(f)$ *is* ϵ*-differentially private iff for any* $\mathcal{S} \subseteq \mathtt{Range}(f)$ *and any database* D' *where* $d(D, D') = 1$*,*

$$\Pr(f(D) \in \mathcal{S}) \leqslant e^{\epsilon} \Pr(f(D') \in \mathcal{S}) \tag{2.2}$$

A commonly used relaxation of Definition 2.1 is (ϵ, δ)-differential privacy, in which an additive constant of δ is allowed in addition to the multiplicative e^{ϵ}.

[1]The most precise notation would be $\mathbb{D}^N$, but we drop the superscript to simplify notation since the size of the database should always be clear from the context.

Definition 2.2 ((ϵ, δ)-differential privacy (Dwork, 2006))**.** *For any input database* D*, a randomized algorithm* $f : \mathbb{D} \to \mathtt{Range}(f)$ *is* (ϵ, δ)*-differentially private iff for any* $\mathcal{S} \subseteq \mathtt{Range}(f)$ *and any database* D' *where* $d(D, D') = 1$*,*

$$\Pr(f(D) \in \mathcal{S}) \leqslant e^{\epsilon} \Pr(f(D') \in \mathcal{S}) + \delta \tag{2.3}$$

The most common approach to ensure differential privacy is to perturb the correct result. To determine how much perturbation is required, we must compute the *sensitivity* of the function we want to privatize. Here, sensitivity is defined as the largest difference between the output of any pair of neighboring databases and not the performance metric $\frac{tp}{n}$.

Definition 2.3 (Global sensitivity (Dwork, 2006))**.** *Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the global sensitivity of* f *is:*

$$GS_f = \max_{d(D,D')=1} |f(D) - f(D')| \tag{2.4}$$

Using Laplace noise to perturb any real-valued query gives the following differentially private method:

Theorem 2.4 (Laplace noise (Dwork, 2006))**.** *Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the computation*

$$f'(D) = f(D) + \mathtt{Laplace}\left(\frac{GS_f}{\epsilon}\right) \tag{2.5}$$

guarantees ϵ*-differential privacy.*

A sequence of differentially private computations also ensures differential privacy. This is called the composition property of differential privacy as stated in Theorem 2.5.

Theorem 2.5 (Composition (Dwork et al., 2006)). *Given a sequence of computations* $\mathbf{f} = f_1,\ldots,f_k$, *with* f_i *guaranteeing* ϵ_i*-differential privacy, then* $\mathbf{f}$ *is* $(\sum_{i=1}^{k} \epsilon_i)$*-differentially private.*

Sometimes we wish to apply differential privacy to domains that are not real-valued, but rather have a number of discrete outcomes. Here, it is unclear how to effectively perturb the output. Instead, an appropriately weighted soft-max called the exponential mechanism can be used.

Theorem 2.6 (Exponential mechanism (McSherry and Talwar, 2007)). *Given a quality function* $q : (\mathbb{D} \times \mathcal{Z}) \to \mathbb{R}$ *that assigns a score to each outcome* $z \in \mathcal{Z}$*, an algorithm that outputs* z *with probability*

$$\Pr(z|D, q) \propto \exp\left(\frac{\epsilon q(D, z)}{2\Delta_q}\right) \tag{2.6}$$

is ϵ*-differentially private.*

McSherry and Talwar (2007) note that the exponential mechanism is also applicable when $\mathcal{Z}$ is continuous. Indeed, using Laplace noise as in Theorem 2.4 is an instance of the exponential mechanism where $q(D, r) = -|f(D) - r|$.

The preceding approaches for obtaining differential privacy use the worst-case, global sensitivity to scale the added noise. For some functions, such as median, the global sensitivity may be large, but the difference between outputs for most neighboring databases is quite small. This motivates the work of Nissim et al. (2007) to explore uses of local sensitivity.

Definition 2.7 (Local sensitivity (Nissim et al., 2007)). *Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the local sensitivity of* f *at* D *is*

$$LS_f(D) = \max_{d(D,D')=1} |f(D) - f(D')|. \tag{2.7}$$

Local sensitivity cannot be directly used to provide differential privacy as the change in the noise scale can actually release information, but a smooth upper bound can be used.

Definition 2.8 (β-smooth sensitivity (Nissim et al., 2007))**.** *For $\beta > 0$, the β-smooth sensitivity of f is*

$$S^*_{f,\beta}(D) = \max_{D' \in \mathbb{D}} LS_f(D') e^{-\beta d(D,D')} \tag{2.8}$$

Using the β-smooth sensitivity and Cauchy-like or Laplace noise provides differential privacy as specified in the following theorem from Nissim et al. (2007).

Theorem 2.9 (Calibrating Noise to Smooth Bounds on Sensitivity (Nissim et al., 2007))**.** *Let $f : \mathbb{D} \to \mathbb{R}$ be any real-valued function and let $S : \mathbb{D} \to \mathbb{R}$ be the β-smooth sensitivity of f, then*

1. *If $\beta \leqslant \frac{\epsilon}{2(\gamma+1)}$ and $\gamma > 1$, the algorithm $f'(D) = f(D) + \frac{2(\gamma+1)S(D)}{\epsilon}\eta$, where η is sampled from the distribution with density $h(z) \propto \frac{1}{1+|z|^\gamma}$, is ϵ-differentially private. Note that when $\gamma = 2$, η is drawn from a standard Cauchy distribution.*

2. *If $\beta \leqslant \frac{\epsilon}{2\ln(\frac{2}{\delta})}$ and $\delta \in (0,1)$, the algorithm $f'(D) = f(D) + \frac{2S(D)}{\epsilon}\eta$, where $\eta \sim \mathtt{Laplace}(1)$, is (ϵ, δ)-differentially private.*

3 UNACHIEVABLE REGION IN PRECISION-RECALL SPACE AND ITS EFFECT ON EMPIRICAL EVALUATION

The material in this chapter was published in Boyd et al. (2012).

3.1 Introduction

With the increased usage of PR curves and AUCPR, the differences between PR analysis and ROC analysis must not be forgotten. PR curves and AUCPR are not a simple substitute for ROC curves and AUCROC in skewed domains. PR curves and ROC curves have different properties, summarized in Sections 2.2 and 2.3, such as the high variability of PR curves at low recall. Additionally, for a given ROC curve, the corresponding PR curve varies with class skew. A related and previously unproven distinction between the two types of curves is that, while any point in ROC space is achievable, not every point in PR space is achievable. Specifically, for a given data set, it is possible to construct a confusion matrix that corresponds to any (false positive rate, true positive rate) pair, but it is *not* possible to do this for every (recall, precision) pair.[1]

We show that this distinction between ROC space and PR space has major implications for the use of PR curves and AUCPR in machine learning. The foremost is that the unachievable points define a minimum PR curve. The area under the minimum PR curve constitutes a portion of AUCPR that any algorithm, no matter how poor, is guaranteed to obtain "for free." Figure 3.1 illustrates this phenomenon. We prove that the size of the unachievable region is *only a function of class skew* and has a simple, closed form.

[1] To be strictly true in ROC space, fractional counts for $\mathtt{tp}$, $\mathtt{fp}$, $\mathtt{fn}$, and $\mathtt{tn}$ must be allowed. The fractional counts can be from a weighted data set or integer counts in an expanded data set.

The unachievable region can influence algorithm evaluation and even behavior in many ways. Even for evaluations using F_1 score, which only consider a single point in PR space, the unachievable region has subtle implications. When averaging AUCPR over multiple tasks (e.g., target predicates in statistical relational learning or queries in information retrieval), the area under the minimum PR curve alone for a non-skewed task may outweigh the total area for all other tasks. A similar effect can occur when the folds used for cross-validation do not have the same skew. Downsampling that changes the skew will also change the minimum PR curve. In algorithms that explicitly optimize AUCPR or MAP during training, algorithm behavior can change substantially with a change in skew. These undesirable effects of the unachievable region can be at least partially offset with straightforward modifications to AUCPR, which we describe.

We explain and characterize the unachievable region in Section 3.2, present modifications to AUCPR in Section 3.3, and discuss the implications of the unachievable region for machine learning evaluation in Section 3.4.

3.2 Achievable and Unachievable Points in PR Space

We first precisely define the notion of an achievable point in PR space. Then we provide an intuitive example to illustrate the concept of an unachievable point. Finally, in Theorems 3.3 and 3.4 we present our central theoretical contributions that formalize the notion of the unachievable region in PR space.

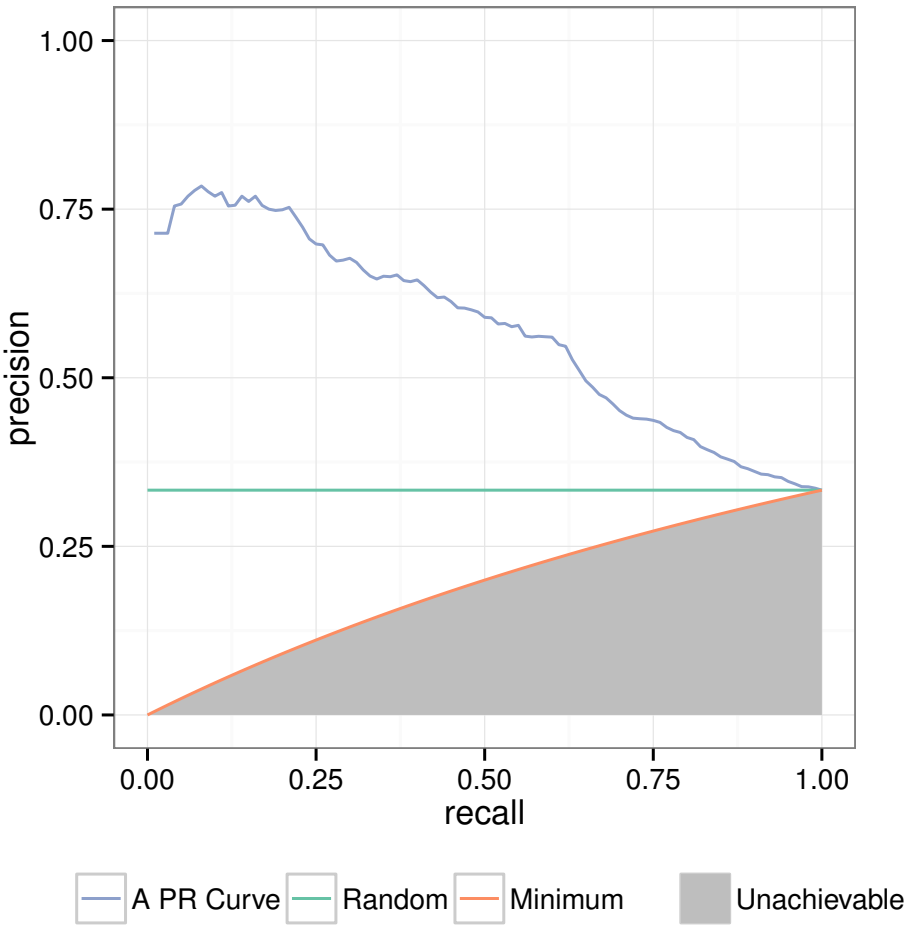


Figure 3.1: Sample PR curve, random guessing curve, and minimum PR curve with $\pi = 0.33$.

Consider a data set D with $N = n + m$ examples, where n is the number of positive examples and m is the number of negative examples. This could be a weighted data set where each example has some weight in $\mathbb{R}^+$, and n and m are the total weight for the positive and negative examples, respectively. We characterize confusion matrices that could actually arise from such a data set in the following definition.

Definition 3.1 (Valid Confusion Matrix)**.** *A* valid *confusion matrix for* $\mathfrak{n}$ *positive and* $\mathfrak{m}$ *negative examples is a tuple* (tp, fp, fn, tn) *such that* $tp, fp, fn, tn \geqslant 0$, $tp + fn = \mathfrak{n}$, *and* $fp + tn = \mathfrak{m}$.

Note that the cell counts in the confusion matrix are not restricted to be integers because we allow weighted data sets.

Achievable points in PR space then are those points that can arise from a valid confusion matrix.

Definition 3.2 (Achievable Point)**.** *For a data set* D, *an* achievable point *in PR space is a point* $(\mathfrak{r}, \mathfrak{p})$ *such that there exists a valid confusion matrix with recall* $\mathfrak{r}$ *and precision* $\mathfrak{p}$. *That is, where*

$$\mathfrak{r} = \frac{tp}{tp + fn} = \frac{tp}{\mathfrak{n}} \tag{3.1}$$

and

$$\mathfrak{p} = \frac{tp}{tp + fp} \tag{3.2}$$

Unachievable Points in PR Space

One can easily show that, as in ROC space, each valid confusion matrix, where $tp > 0$, defines a unique point in PR space. In PR space, both recall and precision depend on the *tp* cell of the confusion matrix, in contrast to the true positive rate and false positive rate used in ROC space. This dependence, together with the fact that a specific data set contains a fixed number of negative and positive examples, imposes limitations on what precisions are possible for a particular recall. Thus, not every point in PR space has a corresponding valid confusion matrix.

To illustrate this effect, consider a data set with $\mathfrak{n} = 100$ and $\mathfrak{m} = 200$. Table 3.1(a) shows a valid confusion matrix with $\mathfrak{r} = 0.2$ and $\mathfrak{p} = 0.2$. Consider holding precision constant while increasing recall. Obtaining $\mathfrak{r} = 0.4$ is possible with $tp = 40$ and $fn = 60$. Notice that keeping $\mathfrak{p} = 0.2$

Table 3.1: For a data set with 100 positive and 200 negative examples, (a) shows a valid confusion matrix with $r = 0.2$ and $p = 0.2$, while (b) is an invalid confusion matrix attempting to obtain $r = 0.6$ and $p = 0.2$.

	Actual	
Predicted	Positive	Negative
Positive	20	80
Negative	80	120
Total	100	200

(a) Valid

	Actual	
Predicted	Positive	Negative
Positive	60	240
Negative	40	**-40**
Total	100	200

(b) Invalid

requires increasing fp from 80 to 160. With a fixed number of negative examples in the data set, increases in fp cannot continue indefinitely. For this data set, $r = 0.5$ with $p = 0.2$ is possible by using all of the negatives as false positives (so $tn = 0$). However, maintaining $p = 0.2$ for any $r > 0.5$ is impossible. Table 3.1(b) illustrates an attempted confusion matrix with $r = 0.6$ and $p = 0.2$. Achieving $p = 0.2$ at this recall requires $fp > m$. This forces $tn < 0$ and makes the confusion matrix invalid.

The following theorem formalizes this restriction on achievable points in PR space.

Theorem 3.3. *An achievable point in PR space with precision (p) and recall (r) must satisfy*

$$p \geqslant \frac{\pi r}{1 - \pi + \pi r} \tag{3.3}$$

where $\pi = \frac{n}{N}$ is the skew.

Proof. From the definition of precision,

$$p = \frac{tp}{tp + fp}. \tag{3.4}$$

But since the number of false positives is limited by the number of negatives because the confusion matrix must be valid, $fp \leqslant (1-\pi)\mathrm{N}$, so

$$\mathrm{p} \geqslant \frac{tp}{tp + (1-\pi)\mathrm{N}}. \tag{3.5}$$

From the definition of recall, $tp = \mathrm{r}\pi\mathrm{N}$, and thus

$$\mathrm{p} \geqslant \frac{\mathrm{r}\pi\mathrm{N}}{\mathrm{r}\pi\mathrm{N} + (1-\pi)\mathrm{N}}. \tag{3.6}$$

We can reasonably assume the data set is non-empty ($\mathrm{N} > 0$), so N cancels out and we are left with

$$\mathrm{p} \geqslant \frac{\mathrm{r}\pi}{\mathrm{r}\pi + 1 - \pi}. \tag{3.7}$$

□

Note that a point's achievability depends solely on the skew and not on a data set's size. Thus, we often refer to achievability in terms of the skew and not in reference to any particular data set.

Unachievable Region in PR Space

Theorem 3.3 gives a constraint that every achievable point in PR space must satisfy. For a given skew, there are many points that are unachievable, and we refer to this collection of points as the *unachievable region* of PR space. In this section we study the properties of the unachievable region.

The constraint on precision and recall in Equation (3.3) makes no assumptions about a model's performance. Consider a model that produces the worst possible ranking, where each negative example is ranked ahead of every positive example. Building a PR curve based on this ranking means placing one PR point at $(0, 0)$ and a second PR point at $(1, \pi)$. Davis and Goadrich (2006) provide a method for interpolating between points in

PR space; interpolation is nonlinear in PR space but is linear between the corresponding points in ROC space. Interpolating between the two known points gives intermediate points with recall of $r_i = \frac{i}{n}$ and precision of $p_i = \frac{\pi r_i}{(1-\pi)+r_i\pi}$, for $0 \leqslant i \leqslant n$. This is the equality case from Theorem 3.3, so Equation (3.3) is a tight lower bound on precision. We call the curve produced by this ranking the *minimum PR curve* because it lies on the boundary between the achievable and unachievable regions of PR space; see Figure 3.2 for examples. For a given skew, all achievable points are on or above the minimum PR curve.

The minimum PR curve has an interesting implication for AUCPR and average precision (AP). Any model must produce a PR curve that lies above the minimum PR curve. Thus, the AUCPR score includes the size of the unachievable region "for free." In the following theorem, we provide a closed form solution for calculating the area of the unachievable region.

Theorem 3.4. *The area of the unachievable region in PR space and the minimum* AUCPR*, for skew π, is*

$$\text{AUCPR}_{\text{MIN}} = 1 + \frac{(1-\pi)\ln(1-\pi)}{\pi} \tag{3.8}$$

Proof. Since Equation (3.3) gives a lower bound for the precision at a particular recall, the unachievable area is the area below the curve $f(r) = \frac{r\pi}{1-\pi+r\pi}$.

$$\begin{aligned}
\text{AUCPR}_{\text{MIN}} &= \int_0^1 \frac{r\pi}{1-\pi+r\pi}\,dr \\
&= \left.\frac{r\pi + (\pi-1)\ln(\pi(r-1)+1)}{\pi}\right|_{r=0}^{r=1} \\
&= \frac{1}{\pi}(\pi + (\pi-1)(\ln(1) - \ln(1-\pi))) \\
&= 1 + \frac{(1-\pi)\ln(1-\pi)}{\pi}
\end{aligned}$$

□

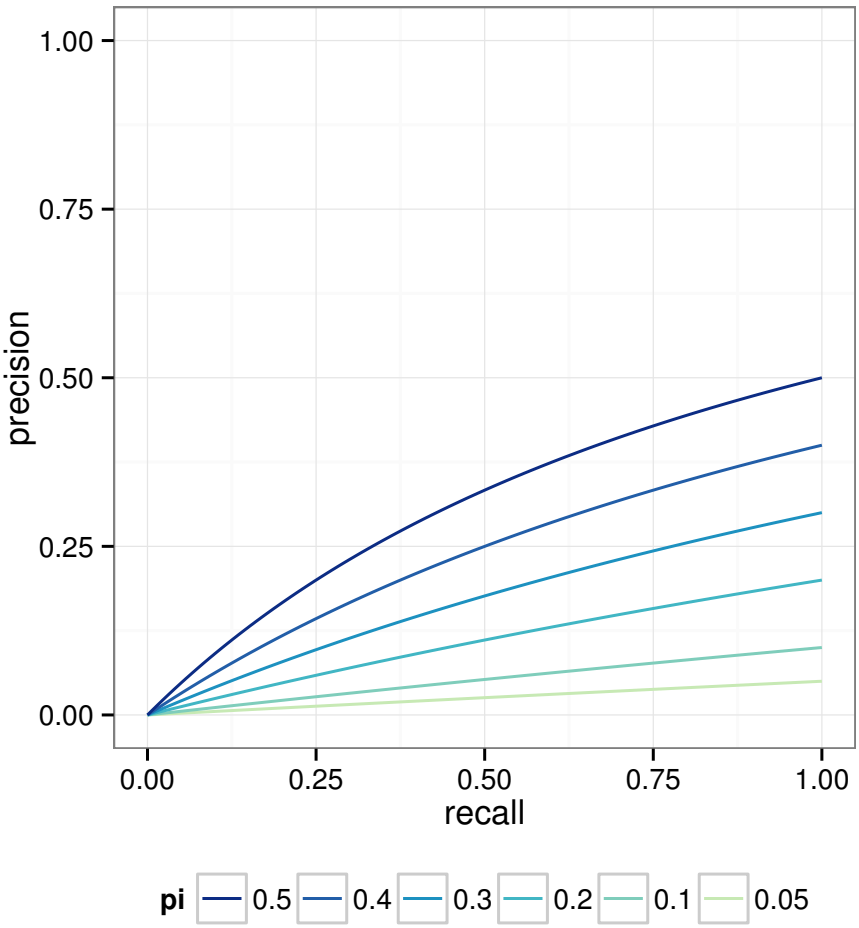


Figure 3.2: Minimum PR curves for several values of π.

A plot of $\text{AUCPR}_{\text{MIN}}$ versus the skew is shown in Figure 3.3(a).

Similar to AUCPR, Equation (3.3) also defines a minimum for AP. Average precision is the mean precision after correctly labeling each positive example in the ranking, so the minimum takes the form of a discrete summation. Unlike AUCPR, which is calculated from interpolated curves, the

minimum AP depends on the number of positive examples because it controls the number of terms in the summation.

Theorem 3.5. *The minimum* AP*, for a data set with* n *positive and* m *negative examples, is*

$$\mathrm{AP}_{\mathrm{MIN}} = \frac{1}{n}\sum_{i=1}^{n}\frac{i}{i+m}$$

Proof.

$$\begin{aligned}
\mathrm{AP}_{\mathrm{MIN}} &= \frac{1}{n}\sum_{i=1}^{n}\frac{\frac{\pi i}{n}}{1-\pi+\frac{\pi i}{n}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\frac{ni}{(n+m)n}}{1+\frac{n}{n+m}(\frac{i}{n}-1)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\frac{i}{n+m}}{\frac{i+m}{n+m}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{i}{i+m}
\end{aligned}$$

□

This precisely captures the natural intuition that the worst AP involves labeling all negatives examples as positive before starting to label the positives.

The existence of the minimum AUCPR and minimum AP can affect the qualitative interpretation of a model's performance. For example, changing the skew of a data set from 0.01 to 0.5 increases the minimum AUCPR by approximately 0.3. This leads to an automatic jump of 0.3 in AUCPR simply by changing the data set and with absolutely no change to the learning algorithm. This type of change in skew is common in data from case-control studies versus observational data or when downsampling the negative examples for computational or learning reasons, as in Sutskever et al. (2009) and Natarajan et al. (2012).

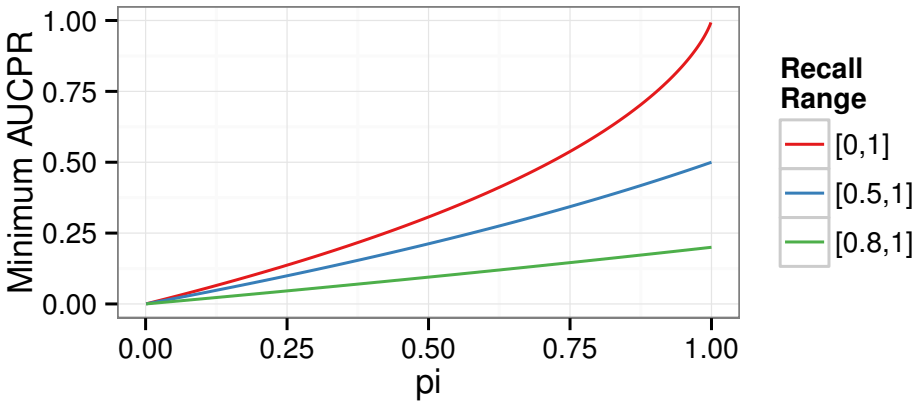
Since the majority of the unachievable region is at higher recalls, the effect of $\text{AUCPR}_{\text{MIN}}$ becomes more pronounced when restricting the area calculation to high levels of recall. Calculating AUCPR for recalls above a threshold is frequently done due to the high variance of precision at low recall or because the learning problem requires high recall solutions (e.g., medical domains such as breast cancer risk prediction). Corollary 3.6 gives the formula for computing $\text{AUCPR}_{\text{MIN}}$ when the area is calculated over a restricted range of recalls. See Figure 3.3(a) for minimum AUCPR when calculating the area over restricted recall. The increased impact of the minimum AUCPR when focusing on high recall is apparent in Figure 3.3(b), where $\text{AUCPR}_{\text{MIN}}$ is scaled to the maximum AUCPR possible in the restricted area. $\text{AUCPR}_{\text{MAX}}$ is the AUCPR achieved by a perfect ranking of the examples. $\text{AUCPR}_{\text{MAX}} = 1$ when working with the entire PR curve and $\text{AUCPR}_{\text{MAX}} = b - a$ when restricting recall to $a \leqslant r \leqslant b$.

Corollary 3.6. *For calculation of* AUCPR *over recalls in* $[a, b]$ *where* $0 \leqslant a < b \leqslant 1$,

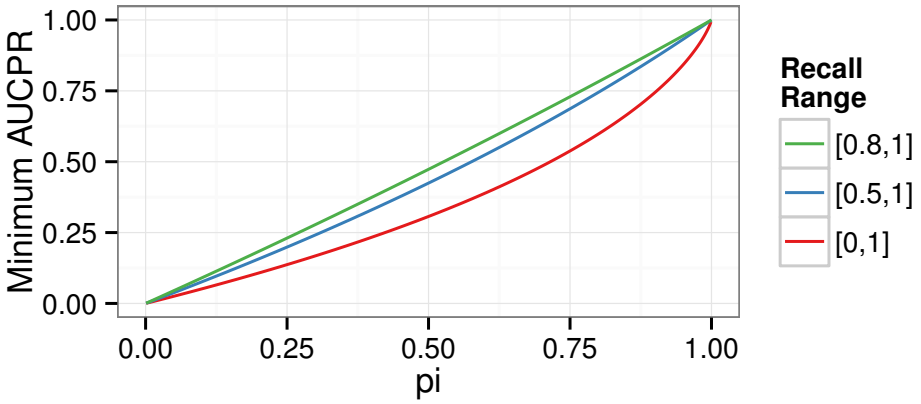
$$\text{AUCPR}_{\text{MIN}} = b - a + \frac{1-\pi}{\pi} \ln\left(\frac{\pi(a-1)+1}{\pi(b-1)+1}\right)$$

Proof. Same as the proof of Theorem 3.4 with limits of a and b in the definite integral instead of 0 and 1. □

Degenerate data sets where $\pi = 0$ and $\pi = 1$ are worth considering briefly because they do sometimes occur and Equation (3.8) for $\text{AUCPR}_{\text{MIN}}$ is undefined when $\pi = 0$ or $\pi = 1$. We propose setting $\text{AUCPR}_{\text{MIN}} = 0$ when $\pi = 0$ and $\text{AUCPR}_{\text{MIN}} = 1$ when $\pi = 1$ since these are the limits of Equation (3.8) as π approaches 0 and 1, respectively. This also has a reasonable interpretation for the area under the curve. When $\pi = 0$, there are no positive examples, so precision is always 0. Therefore, the PR curve must lie on the x-axis and the area under the curve is 0, regardless of the ranking. Analogously, when $\pi = 1$ and all the examples are positive,



(a) Minimum AUCPR versus π



(b) Minimum AUCPR scaled to maximum AUCPR $\left(\frac{\text{AUCPR}_{\text{MIN}}}{\text{AUCPR}_{\text{MAX}}}\right)$ versus π.

Figure 3.3: Minimum AUCPRs for area calculated over entire PR curve [0,1], recall above 0.5 [0.5,1], and recall above 0.8 [0.8,1]. Direct areas are shown in (a) and the area scaled to the maximum AUCPR in (b).

precision must be 1. So the PR curve is always a line at $p = 1$, and AUCPR is 1.

3.3 Modifying AUCPR based on the Unachievable Region

The unachievable region represents a lower bound on AUCPR and it is important to develop evaluation metrics that account for this. We believe that any metric A' that replaces AUCPR should satisfy at least the two properties. First, A' should relate to AUCPR. Assume AUCPR was used to estimate the performance of classifiers $C_1, \ldots, C_n$ on a *single* test set. If $\text{AUCPR}(C_i, \text{test}_D) > \text{AUCPR}(C_j, \text{test}_D)$, then $A'(C_i, \text{test}_D) > A'(C_j, \text{test}_D)$, as test set test_D's skew affects each model equally. Note that this property may not be appropriate or desirable when aggregating scores across *multiple* test sets, as is done in cross validation, because each test set may have a different skew. Second, A' should have the same range for every data set, regardless of skew. This is necessary, though not sufficient, to achieve meaningful comparisons across data sets. AUCPR does not satisfy the second requirement because, as shown in Theorem 3.4, its range depends on the data set's skew.

We propose normalizing the area under the PR curve so the worst ranking has a score of 0 and the best ranking has a score of 1.

Definition 3.7 (AUCNPR)**.** *The* normalized *area under the PR curve is*

$$\text{AUCNPR} = \frac{\text{AUCPR} - \text{AUCPR}_{\text{MIN}}}{\text{AUCPR}_{\text{MAX}} - \text{AUCPR}_{\text{MIN}}}$$

where $\text{AUCPR}_{\text{MAX}} = 1$ *when calculating area under the entire PR curve and* $\text{AUCPR}_{\text{MAX}} = b - a$ *when restricting recall to* $a \leqslant r \leqslant b$.

Regardless of skew, the best possible classifier will have an AUCNPR

of 1 and the worst possible classifier will have an AUCNPR of 0. AUCNPR also preserves the ordering of algorithms on the same test set since $\text{AUCPR}_{\text{MAX}}$ and $\text{AUCPR}_{\text{MIN}}$ are constant for the same data set. Thus, AUCNPR satisfies our proposed requirements for a replacement of AUCPR. Furthermore, by accounting for the unachievable region, it makes comparisons between data sets with different skews more meaningful than when using AUCPR.

AUCNPR measures the proportion of the achievable area in PR space that a classifier attains. In this sense, AUCNPR is properly undefined when $\pi = 0$ or $\pi = 1$ because there is no difference between the minimum and maximum PR curves. At $\pi = 0$ or $\pi = 1$, every ranking of the examples produces the exact same PR curve. As a convention, when a numeric score is required, AUCNPR $= 1$ for $\pi = 1$ and AUCNPR $= 0$ for $\pi = 0$ seem reasonable. For $\pi = 0$, this is exactly what Definition 3.7 gives when assuming $\text{AUCPR}_{\text{MIN}} = 0$ if $\pi = 0$. Additionally, it makes sense for PR analysis, which focuses on the positive examples, to give no credit in a task that has no positive examples. For $\pi = 1$, however, Definition 3.7 is undefined. If a numeric score is required for reporting or aggregation purposes, setting AUCNPR to always be 1 when $\pi = 1$ is a reasonable solution, although arguments could be made for 0 or π depending on the application and goals of evaluation.

We chose to normalize to the minimum AUCPR because it ensures the range of AUCNPR is always the same. One simple alternative is to normalize to the AUCPR for random guessing, which is simply π. While it is simpler, normalizing to π has two drawbacks. First, the range of scores depends on the skew, and therefore is not consistent across different data sets. Second, it can result in a negative score if an algorithm performs worse than random guessing, which seems counter-intuitive for an area *under* a curve.

3.4 Discussion and Recommendations

We believe all practitioners using evaluation scores based on PR space (e.g., PR curves, AUCPR, AP, F_1) should be cognizant of the unachievable region and how it affects their analyses.

Visually inspecting the PR curve or looking at an AUCPR score often gives an intuitive sense for the quality of an algorithm or difficulty of a task or data set. If the skew is extremely large, the effect of the very small unachievable region on PR analysis is negligible. However, there are many instances where the skew is closer to 0.5 and the unachievable area is not insignificant. With $\pi = 0.1$, $\text{AUCPR}_{\text{MIN}} \approx 0.05$, and it increases as π approaches 0.5. AUCPR is used in many applications where $\pi > 0.1$ (Hu et al., 2009; Sonnenburg et al., 2006; Liu and Shriberg, 2007). Therefore, a general awareness of the unachievable region and its relationship to skew is important when casually comparing or inspecting PR curves and AUCPR scores. A simple recommendation that will make the unachievable region's impact on results clear is to *always show the minimum PR curve on PR curve plots*.

Next, we discuss several specific situations where the unachievable region is highly relevant for machine learning.

Aggregation for Cross-Validation

In cross-validation, stratification is typically used to ensure all folds have the same skew. However, particularly in relational domains, this is not always the case. In relational domains, stratification must consider fold membership constraints imposed by links between objects that, if violated, would bias the results of cross-validation. For example, consider the bioinformatics task of protein secondary structure prediction. Putting amino acids from the same protein in different folds has two drawbacks. First, it could bias the results as information about the same protein is

in both the train and test sets. Second, it does not properly simulate the ultimate goal of predicting the structure of entirely novel proteins. Links between examples occur in most relational domains, and placing all linked items in the same fold can lead to substantial variation in the skew of the folds. Because the different skews yield different $\text{AUCPR}_{\text{MIN}}$s, care must be taken when aggregating results to create a single summary statistic of an algorithm's performance.

Cross-validation assumes that each fold is sampled from the same underlying distribution. Even if the skew varies across folds, the merged data set is the best estimate of the underlying distribution and thus the overall skew. Ideally, aggregate descriptions, like a PR curve or AUCPR, should be calculated on a single, merged data set. However, merging directly compares probability estimates for examples in different folds and assumes that the models are calibrated. Unfortunately, this is rarely a primary goal of machine learning and learned models tend to be poorly calibrated (Forman and Scholz, 2010).

With uncalibrated models, the most common practice is to average the results from each fold. For AUCPR, the mean of the AUCPR from each fold is typically used. For a PR curve, vertical averaging of the individual PR curves from each fold provides a summary curve. In both cases, averaging fails to account for any differences in the unachievable region that arise due to variations in class skew. As shown in Theorem 3.4, the range of possible AUCPR values varies according to a fold's skew. Similarly, when vertically averaging PR curves, a particular recall level will have varying ranges of potential precision values for each fold if the folds have different skews. Even a single fold, which has much higher precision values due to a substantially lower skew, can cause a higher vertically averaged PR curve because of its larger unachievable region. Failing to account for fold-by-fold variation in skew can lead to overly optimistic assessments when using straightforward averaging.

We recommend averaging AUCNPR instead of AUCPR when evaluating area under the curve. Averaging AUCNPR, which has the same range regardless of skew, helps reduce (but not eliminate) the skew's effect compared to averaging AUCPR. A technique for creating a summary PR curve from multiple curves with different skews is not known. Summary PR curves are discussed as future work in Section 6.1.

Aggregation among Different Tasks

Machine learning algorithms are commonly evaluated on several different tasks. This setting differs from cross-validation because each task is not assumed to have the same underlying distribution. While the tasks may be unrelated (Tang et al., 2009), they often come from the same domain. For example, the tasks could be the truth values of different predicates in a relational domain (Kok and Domingos, 2010; Mihalkova and Mooney, 2007) or different queries in an IR setting (Manning et al., 2008). Often, researchers report a single, aggregate score by averaging the results across the different tasks. However, the tasks can potentially have very different skews, and therefore different minimum AUCPRs. Therefore, averaging AUCNPR scores, which (somewhat) control for skew, is preferred to averaging AUCPR.

In statistical relational learning, researchers frequently evaluate algorithms by reporting the average AUCPR over a variety of tasks in a single data set (Mihalkova and Mooney, 2007; Kok and Domingos, 2010). As a case study, consider the commonly used `IMDB` data set[2] that describes relationships among movies, actors, and directors. Here, the task is to predict the probability that each possible grounding of each predicate is true. Across all predicates in `IMDB`, the skew of true groundings is relatively low ($\pi = 0.06$), but there is significant variation in the skew of individual predicates. For example, the `gender` predicate has a skew close to $\pi = 0.5$,

[2] Available from `http://alchemy.cs.washington.edu/`.

Table 3.2: Average AUCPR and AUCNPR scores for each predicate in the IMDB data set. Results are for the LSM algorithm from Kok and Domingos (2010). The range of scores shows the difficulty and skews of the prediction tasks vary greatly. By accounting for the (potentially large) unachievable regions, AUCNPR yields a more conservative overall estimate of performance.

Predicate	AUCPR	AUCNPR
actor	1.000	1.000
director	1.000	1.000
gender	0.509	0.325
genre	0.624	0.611
movie	0.267	0.141
workedUnder	1.000	1.000
Mean	0.733	0.680

whereas a predicate such as genre has a skew closer to $\pi = 0.05$. While presenting the mean AUCPR across all predicates is a good first approach, it leads to averaging values that do not all have the same range. The gender predicate's range for AUCPR is $[0.31, 1.0]$ while the genre predicate's range is $[0.02, 1.0]$. Thus, an AUCPR of 0.4 means very different things on these two predicates. For the gender predicate, this score is worse than random guessing, while for the genre predicate, this is a reasonably high score. In a sense, all AUCPR scores of 0.4 are not created equal, but averaging the AUCPR treats them as equals.

Table 3.2 shows AUCPR and AUCNPR for each predicate on a Markov logic network model learned by the LSM algorithm (Kok and Domingos, 2010). Notice the wide range of scores and that AUCNPR gives a more conservative overall estimate. AUCNPR is still sensitive to skew, so an AUCNPR of 0.4 in the aforementioned predicates still does not imply completely comparable performances, but it is closer than AUCPR.

Downsampling

Downsampling is common when learning on highly skewed tasks. Often the downsampling alters the skew on the train set (e.g., subsampling the negatives to facilitate learning, using data from case-control studies) such that it does not reflect the true skew. PR analysis is frequently used on the downsampled data sets (Sonnenburg et al., 2006; Natarajan et al., 2012; Sutskever et al., 2009). The sensitivity of AUCPR and related scores makes it important to recognize, and if possible quantify, the effect of downsampling on evaluation metrics.

The varying size of the unachievable region provides an explanation and quantification of some of the dependence of PR curves and AUCPR on skew. Thus, AUCNPR, which adjusts for the unachievable region, should be more stable than AUCPR to changes in skew. To explore this, we used SAYU (Davis et al., 2005) to learn a model for the `advisedBy` task in the `UW-CSE` domain for several downsampled train sets. The `UW-CSE` data set[3] (Richardson and Domingos, 2006) contains predicates that describe an academic department, e.g., `taughtBy` and `advisedBy`. Table 3.3 shows the AUCPR and AUCNPR scores on a test set downsampled to the same skew as the train set and on the original (i.e., non-downsampled) test set. AUCNPR has less variance than does AUCPR. However, there is still a sizable difference between the scores on the downsampled test set and the original test set. As expected, the difference increases as the ratio approaches 1 positive to 1 negative. At this ratio, even the AUCNPR score on the downsampled data is more than twice the score on the original skew. This is a massive difference and it is disconcerting that it occurs simply by changing the data set skew. An intriguing area for future research is to investigate scoring metrics that either are less sensitive to skew or that permit simple and accurate transformations that facilitate comparisons between different skews.

[3] Available from `http://alchemy.cs.washington.edu/`.

Table 3.3: AUCPR and AUCNPR scores for SAYU on `advisedBy` task in the `UW-CSE` data set for different train set skews. The downsampled columns report scores on a test set with the same downsampled skew as the train set. The original skew columns report scores on the original test set with a ratio of 1 positive to 24 negatives ($\pi = 0.04$).

	Downsampled		Original Skew	
Ratio	AUCPR	AUCNPR	AUCPR	AUCNPR
1:1	0.851	0.785	0.330	0.316
1:2	0.740	0.680	0.329	0.315
1:3	0.678	0.627	0.343	0.329
1:4	0.701	0.665	0.314	0.299
1:5	0.599	0.560	0.334	0.320
1:10	0.383	0.352	0.258	0.242
1:24	0.363	0.349	0.363	0.349

F_β Score

While PR curves allow evaluations without settling on a specific threshold, some single-threshold evaluation measures are closely related to PR analysis. If precision and recall are used in evaluating a confusion matrix, such as with the F_1 score, this corresponds to a point in PR space. Even with a single operating point, the unachievable region still applies and the minimum PR curve and random guessing PR curve are relevant. Thus, the relationship between a point in PR space and the unachievable region is informative.

The most commonly used single-threshold measure impacted by the unachievable region is the F_β family,

$$F_\beta = \frac{(1+\beta^2)pr}{\beta^2 p + r}$$

where $\beta > 0$ is a parameter to control the relative importance of recall and precision (Manning et al., 2008; Carterette and Voorhees, 2011). Most often, the F_1 score ($\beta = 1$) is used. The F_1 score is also the harmonic mean of precision and recall. We focus our discussion on the F_1 score, but similar analysis applies to F_β. Figure 3.4 shows the contours of the F_1 score over PR space.

The unachievable region has a subtle interaction with F_1 score that changes depending on the skew. Because F_1 combines precision and recall into a single, real-valued score, it necessarily loses information. One aspect of this information loss is that PR points with the same F_1 score can have vastly different relationships with the unachievable region. Consider points A and B in Figure 3.4. Both points have an F_1 score of 0.4, but point A has modest recall with good precision while point B is on the minimum PR curve. The F_1 score does not differentiate between two models with these operating points in PR space. However, one model is even worse than random guessing, while the other might be excellent for some tasks.

Whereas losing information is inevitable with a summary like the F_1 score, the problem arises partly because the F_1 score treats recall and precision interchangeably. This is not unique to $\beta = 1$. While F_β changes their relative importance, the assumption remains that precision and recall, appropriately scaled by β, are equivalent for assessing performance. Our findings about the unachievable region show this is problematic, as recall and precision have fundamentally different properties: every recall has a minimum precision, there is a maximum recall for low precision, and there are no constraints on recall otherwise.

We want to investigate how these drawbacks of F_1 might be addressed. Given the F_1 score's popularity, particularly in information retrieval, is there a modification to the formula that would alleviate the problematic interaction with the minimum PR curve and the unachievable region? Or, would it be more informative to combine F_1 with the distance to the

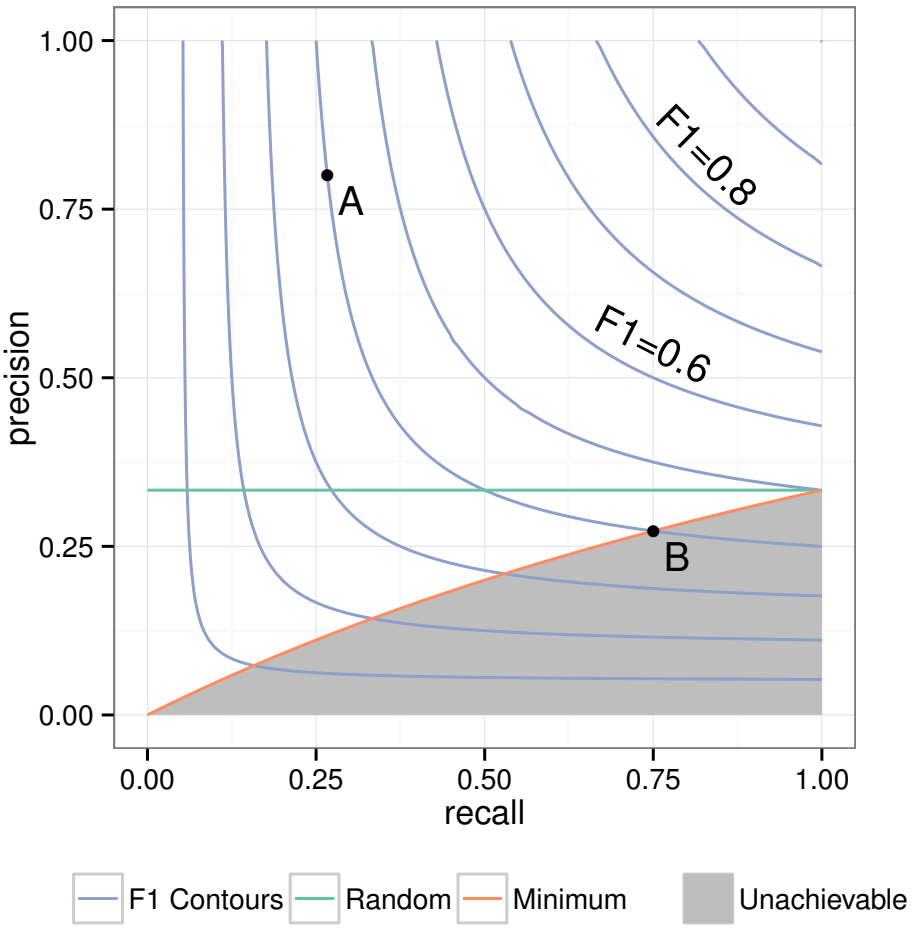


Figure 3.4: Contours of F_1 score in PR space with minimum PR curve and unachievable region for $\pi = 0.33$. Points A and B both have an F_1 score of 0.4, but B is on the minimum PR curve while A has modest recall and high precision. Using F_1 score alone, these two very different performances are indistinguishable.

minimum PR curve?

While a modified F_1 score that is sensitive to the unachievable region

would be useful, an ideal solution may not exist. Consider three simple requirements for a modified F_1 score, f':

$$\forall p \text{ s.t. } 0 < p \leqslant 1, \quad f'(r_1, p) < f'(r_2, p) \quad \text{iff} \quad r_1 < r_2 \tag{3.9}$$

$$\forall r \text{ s.t. } 0 < r \leqslant 1, \quad f'(r, p_1) < f'(r, p_2) \quad \text{iff} \quad p_1 < p_2 \tag{3.10}$$

$$f'(r, p) = 0 \quad \text{if} \quad p = \frac{r\pi}{1 - \pi + r\pi} \tag{3.11}$$

Equations (3.9) and (3.10) capture the expectation that an increase in precision or recall while the other is constant should always increase f' ($p = 0$ and $r = 0$ are excluded so that F_1 satisfies these two conditions). Equation (3.11) ensures $f' = 0$ if the PR point is on the minimum PR curve and is the additional constraint we want to impose.

However, these three properties are impossible to satisfy. Suppose we have an f' that satisfies the three properties. Choose any recall $0 < r < 1$ and let $p = \frac{r\pi}{1-\pi+r\pi}$. Note that $0 < p < \pi$ because $r < 1$. Then,

$$\begin{aligned} 0 = f'(r, p) \quad & \text{from Equation (3.11)} \\ f'(r, p) < f'(r, \pi) \quad & \text{from Equation (3.10)} \\ f'(r, \pi) < f'(1, \pi) \quad & \text{from Equation (3.9)} \\ f'(1, \pi) = 0 \quad & \text{from Equation (3.11)} \end{aligned}$$

Putting these four equations together we have a contradiction: $0 < 0$. So there cannot be an f' that has all three properties.

Relaxing Equations (3.9) and (3.10) to

$$\forall p \text{ s.t. } 0 < p \leqslant 1, \quad f'(r_1, p) \leqslant f'(r_2, p) \quad \text{iff} \quad r_1 \leqslant r_2$$

$$\forall r \text{ s.t. } 0 < r \leqslant 1, \quad f'(r, p_1) \leqslant f'(r, p_2) \quad \text{iff} \quad p_1 \leqslant p_2$$

makes it possible to construct an f' that satisfies the requirements, but

implies $f'(r, p) = 0$ if $p \leqslant \pi$. This seems unsatisfactory because it ignores all distinctions once the performance is worse than random guessing. If assigning 0 whenever $p \leqslant \pi$ is acceptable, one modified F_1 score that satisfies the relaxed requirements is

$$f'(r, p) = \begin{cases} 0 & \text{if } p \leqslant \pi \\ \frac{2(p-\pi)r}{p-\pi+(1-\pi)r} & \text{if } p > \pi \end{cases}$$

which assigns 0 to any PR point worse than random guessing and uses the harmonic mean of recall and a normalized precision ($\frac{p-\pi}{1-\pi}$) otherwise. Extension to a modified F_β for unequal weighting of recall and normalized precision is straightforward.

3.5 Chapter Summary

In this chapter, we demonstrated that a region of precision-recall space is unachievable for any particular ratio of positive to negative examples. With the precise characterization of this unachievable region given in Theorems 3.3 and 3.4, we further the understanding of the effects of downsampling and the impact of the minimum PR curve on score aggregation, downsampling, and F measure. These nuances of precision-recall space, particularly the dependence between precision and recall that leads to unachievable points, inspire us to explore the process of creating PR curves and calculating AUCPR in more depth in the next chapter.

4 AREA UNDER THE PRECISION-RECALL CURVE: POINT ESTIMATES AND CONFIDENCE INTERVALS

After identifying and discussing the unachievable region in the previous chapter, we seek the best methods for estimating AUCPR and creating confidence intervals in this chapter. The interdependence of precision and recall, unlike the true and false positive rates used in ROC space, creates the distinctive "saw-shape" typical of simple plotting of the PR curve. However, there are several other ways of calculating AUCPR and creating a PR curve and we describe and empirically evaluate them in this chapter. Additionally, we look at methods of generating confidence intervals for AUCPR.

The material in this chapter was published in Boyd et al. (2013).

4.1 Introduction

Machine learning researchers build a PR curve by plotting precision-recall pairs, or points, that are obtained using different thresholds on a probabilistic or other continuous-output classifier. This is similar to the way an ROC curve is built by plotting true and false positive rate pairs obtained using different thresholds. After plotting the points in PR space, we next seek to construct a curve, compute its AUCPR, and calculate 95% (or other) confidence intervals (CIs) around the curve or the AUCPR.

However, the best method to construct the curve and calculate area is not readily apparent. The small data set in Table 4.1 produces the PR points in Figure 4.1, and these points give rise to several questions. How should multiple points with the same x-value (recall) be treated (i.e., is the maximum, minimum, or mean representative)? Is linear interpolation in any form appropriate? Davis and Goadrich (2006) showed that using a line to connect the highest points at each recall is overly optimistic, but what

about other schemes for connecting points using lines? Should a convex hull be used, either in ROC space, as suggested by Davis and Goadrich (2006), or in PR space?

Different answers to the these questions lead to at least four distinct methods, with several variations, that have been used in machine learning, statistics, and related areas to compute AUCPR. Additionally, we are interested not just in point estimates of AUCPR, but in the variance as well, and we identify four methods that have been used to construct CIs for AUCPR. This chapter discusses and analyzes eight estimators and four CIs empirically. We provide evidence in favor of computing AUCPR using the *lower trapezoid*, *average precision*, or *interpolated median* estimators and using *binomial* or *logit* CIs rather than other methods, including the more widely-used ten-fold *cross-validation*. The differences in results using these approaches are most striking when data are highly skewed, which is exactly the case when PR curves are most preferred over ROC curves.

Section 4.2 describes our notation for PR curve points and areas, Section 4.3 describes the estimators and CIs we evaluate, and Section 4.4 presents case studies of the estimators and CIs in action.

4.2 Area Under the Precision-Recall Curve

Consider a binary classification task where models produce continuous outputs, denoted by the random variable Z, for each example. Diverse applications are subsumed by this setup, e.g., a medical test to identify diseased and disease-free patients, a document ranker to distinguish between relevant and non-relevant documents for a particular query, and generally any binary classification task. The two categories are often naturally labeled as positive (e.g., diseased, relevant) or negative (e.g., disease-free, non-relevant). Following the literature on ROC curves (Bamber, 1975; Pepe, 2004), we denote the output values for the negative examples by

Table 4.1: Data set with $n = 10$ and $m = 20$ used to generate the PR points in Figure 4.1. Model outputs were sampled from $\mathcal{N}(0, 1)$ for negatives and $\mathcal{N}(0.5, 1)$ for positives. Recall and precision values are for labeling that row and above as positive.

Output	True Label	Recall	Precision
0.95	positive	0.20	1.00
0.90	negative	0.20	0.50
0.85	negative	0.20	0.33
0.80	positive	0.40	0.50
0.75	positive	0.60	0.60
0.70	negative	0.60	0.50
0.65	negative	0.60	0.43
0.60	negative	0.60	0.38
0.55	negative	0.60	0.33
0.50	positive	0.80	0.40
0.45	negative	0.80	0.36
0.40	negative	0.80	0.33
0.35	negative	0.80	0.31
0.30	negative	0.80	0.29
0.25	negative	0.80	0.27
0.20	negative	0.80	0.25
0.15	positive	1.00	0.29
0.10	negative	1.00	0.28
0.05	negative	1.00	0.26
0.00	negative	1.00	0.25

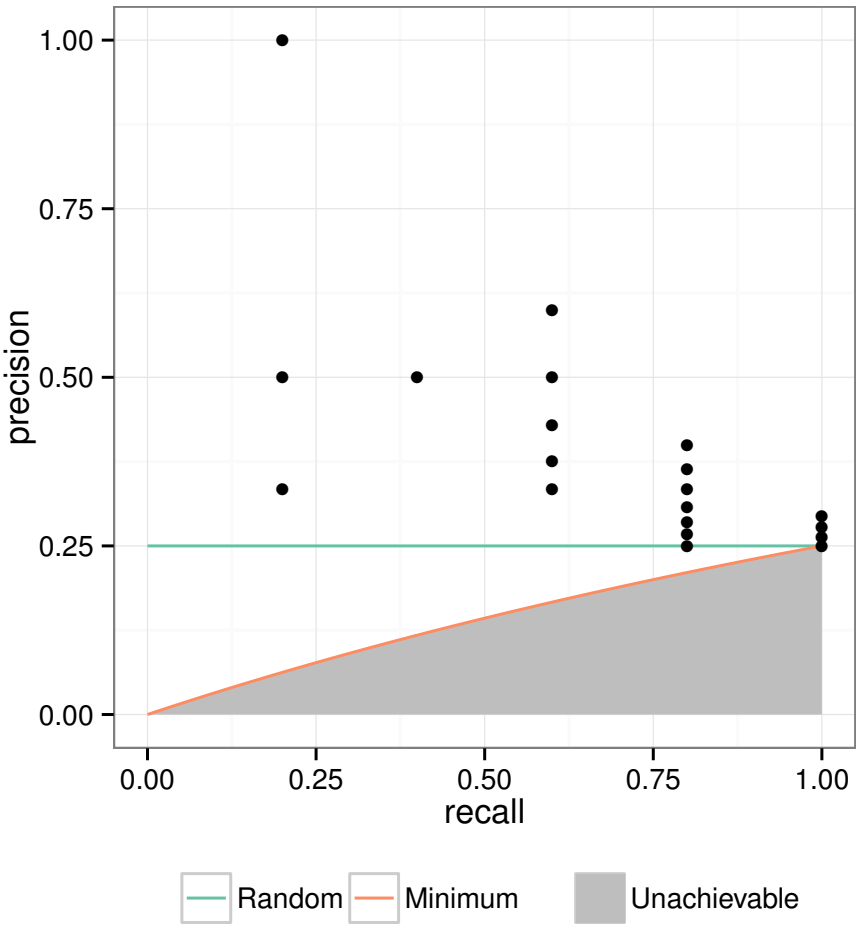


Figure 4.1: Empirical PR points, random and minimum PR curves, and unachievable region for outputs and labels in Table 4.1 where $\pi = 0.25$.

the random variable X and the output values for the positive examples by Y (Z is a mixture of X and Y). These populations are assumed to be independent when the class is known. Larger output values are associated with positive examples, so for a given threshold c, an example is predicted positive if its score is greater than c. We represent the category or class with the indicator random variable D, where $D = 1$ corresponds to positive examples and $D = 0$ to negative examples. An important aspect of a task or data set is the class skew $\pi = P(D = 1)$. Skew is also known as prevalence or a prior class distribution.

Several techniques exist to assess the performance of binary classification across a range of thresholds. While ROC analysis is the most common, we are interested in the related PR curves. A PR curve may be defined as the set of points:

$$PR(\cdot) = \{(Recall(c), Prec(c)), -\infty < c < \infty\}$$

where $Recall(c) = P(Y > c)$ and $Prec(c) = P(D = 1|Z > c)$. Recall is equivalent to true positive rate or sensitivity (the y-axis in ROC curves), while precision is the same as positive predictive value. Because larger output values are assumed to be associated with positive examples, as c decreases, $Recall(c)$ increases to 1 and $Prec(c)$ approaches π. As c increases, $Prec(c)$ becomes highly variable, as discussed in Section 2.3, though we generally think about $Prec(c)$ reaching 1 as $Recall(c)$ approaches 0. The high variance of precision estimates for recall near 0 is a major difficulty of constructing PR curves.

It is often desirable to summarize the PR curve with a single scalar value. One summary is the area under the PR curve (AUCPR), which we will denote by θ. Following the work of Bamber (1975) on ROC curves, AUCPR is an average of the precision weighted by the probability of a

given threshold.

$$\theta = \int_{-\infty}^{\infty} \mathrm{Prec}(c)\mathrm{dP}(Y \leqslant c) \tag{4.1}$$

$$= \int_{-\infty}^{\infty} \mathrm{P}(D = 1|Z > c)\mathrm{dP}(Y \leqslant c). \tag{4.2}$$

By Bayes' rule and using that Z is a mixture of X and Y,

$$\mathrm{P}(D = 1|Z > c) = \frac{\pi \mathrm{P}(Y > c)}{\pi \mathrm{P}(Y > c) + (1 - \pi)\mathrm{P}(X > c)}$$

and we note that $0 \leqslant \theta \leqslant 1$ since $\mathrm{Prec}(c)$ and $\mathrm{P}(Y \leqslant c)$ are bounded on the unit square. Therefore, θ might be viewed as a probability. If we consider Equation (4.2) as an importance-sampled Monte Carlo integral, we may interpret θ as the fraction of positive examples among those whose output values exceed a randomly selected $c \sim Y$ threshold.

4.3 AUCPR Estimators

In this section we summarize point estimators for θ and then introduce CI methods.

Point Estimators

Let $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ represent observed output values from negative and positive examples, respectively. The skew π is assumed to be given or is set to $n/(n+m)$. An empirical estimate of the PR curve, $\widehat{\mathrm{PR}}(\cdot)$, can be derived by the empirical estimates of each coordinate:

$$\widehat{\mathrm{Recall}}(c) = n^{-1} \sum_{i=1}^{n} \mathrm{I}(y_i > c)$$

$$\widehat{\text{Prec}}(c) = \frac{\pi \widehat{\text{Recall}}(c)}{\pi \widehat{\text{Recall}}(c) + (1-\pi) m^{-1} \sum_{j=1}^{m} \mathbb{1}[x_j > c]}$$

where $\mathbb{1}[A]$ is the indicator function for event A.

We review a number of possible estimators for θ. These estimators, either directly or indirectly, correspond to some assumption about how to interpolate between or approximate the empirical PR points. These interpolations and the differences between the estimators are visually shown on a small data set in Figure 4.2.

Trapezoidal Estimators

For fixed $\widehat{\text{Recall}}(t)$, the empirical precision may not be constant, therefore, $\widehat{\text{PR}}(\cdot)$ is often not one-to-one. Multiple precision values for a single recall occur when $y_{(i)} < x_j < y_{(i+1)}$ for some i and j, where $y_{(i)}$ denotes the ith order statistic (ith largest value among the y_i's). As the threshold increases from $y_{(i)}$ to x_j, recall remains constant while precision decreases. Let $r_i = \widehat{\text{Recall}}(y_{(n-i)})$, such that $r_1 \leqslant r_2 \leqslant \cdots \leqslant r_n$, and let p_i^{max} be the largest sample precision value corresponding to r_i. Likewise, let p_i^{min} be the smallest sample precision value corresponding to r_i. This leads immediately to a few choices for estimators based on the empirical curve using trapezoidal estimation (Abeel et al., 2009):

$$\widehat{\theta}_{LT} = \sum_{i=1}^{n-1} \frac{p_i^{min} + p_{i+1}^{max}}{2} (r_{i+1} - r_i) \tag{4.3}$$

$$\widehat{\theta}_{UT} = \sum_{i=1}^{n-1} \frac{p_i^{max} + p_{i+1}^{max}}{2} (r_{i+1} - r_i) \tag{4.4}$$

These correspond to a *lower trapezoid* approximation in Equation (4.3) and an *upper trapezoid* approximation in Equation (4.4). Note that the *upper trapezoid* method uses an overly optimistic linear interpolation (Davis and

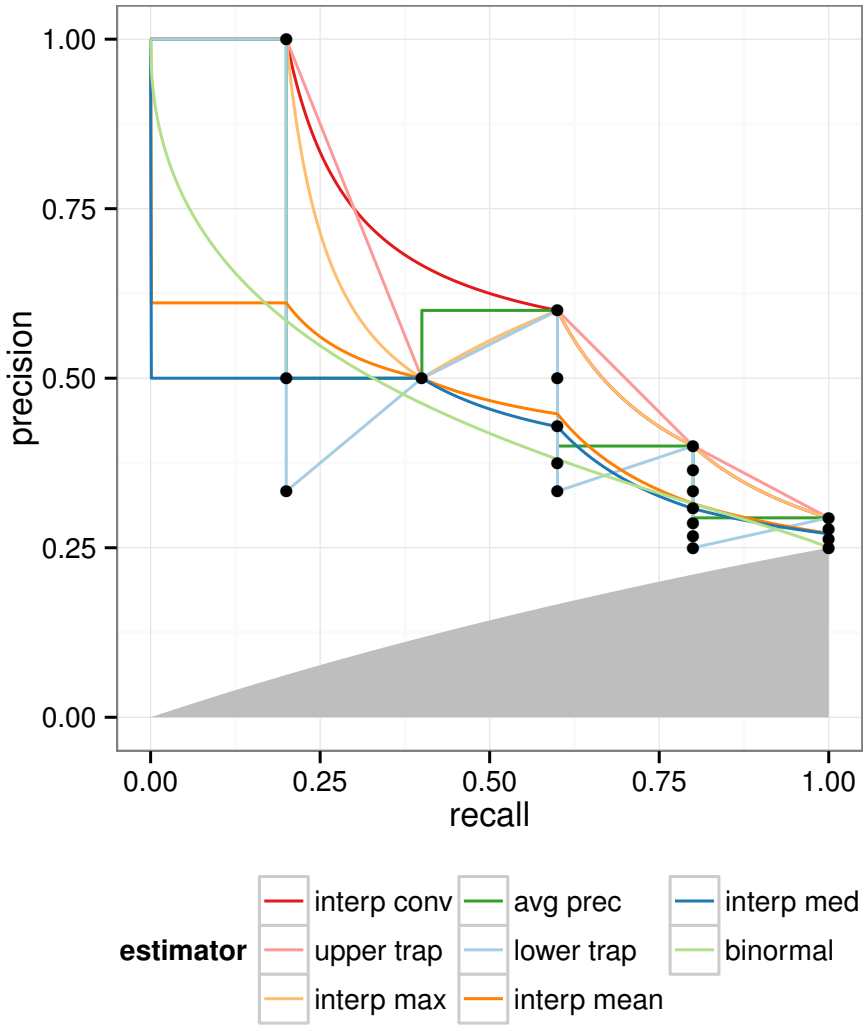


Figure 4.2: PR curves demonstrating the interpolation assumptions of the point estimate methods on the predictions from Figure 4.1 and Table 4.1, where $\pi = 0.25$. The unachievable region is shown in gray.

Goadrich, 2006). We include it for comparison as it is one of the first methods a non-expert is likely to use due to its similarity to estimating area under the ROC curve.

Interpolation Estimators

As suggested by Davis and Goadrich (2006) and Goadrich et al. (2006), we use PR space interpolation as the basis for several estimators. These methods use the non-linear interpolation between known points in PR space derived from a linear interpolation in ROC space.

Davis and Goadrich (2006) and Goadrich et al. (2006) examine the interpolation in terms of the number of true positives and false positives corresponding to each PR point. Here we perform the same interpolation, but use the recall and precision of the PR points directly, which leads to the surprising observation that the interpolation (from the same PR points) does not depend on π.

Theorem 4.1. *For two points in PR space* (r_1, p_1) *and* (r_2, p_2) *(assume WLOG* $r_1 < r_2$*), the interpolation for recall* r' *with* $r_1 \leqslant r' \leqslant r_2$ *is*

$$p' = \frac{r'}{ar' + b} \tag{4.5}$$

where

$$a = 1 + \frac{(1-p_2)r_2}{p_2(r_2 - r_1)} - \frac{(1-p_1)r_1}{p_1(r_2 - r_1)}$$
$$b = \frac{(1-p_1)r_1}{p_1} - \frac{(1-p_2)r_1 r_2}{p_2(r_2 - r_1)} + \frac{(1-p_1)r_1^2}{p_1(r_2 - r_1)}$$

Proof. First, we convert the points to ROC space. Let s_1, s_2 be the false positive rates for the points (r_1, p_1) and (r_2, p_2), respectively. By definition of false positive rate,

$$s_i = \frac{(1-p_i)\pi r_i}{p_i(1-\pi)}. \tag{4.6}$$

A linear interpolation in ROC space for $r_1 \leqslant r' \leqslant r_2$ has a false positive rate of

$$s' = s_1 + \frac{r' - r_1}{r_2 - r_1}(s_2 - s_1). \tag{4.7}$$

Then convert back to PR space using

$$p' = \frac{\pi r'}{\pi r' + (1 - \pi)s'}. \tag{4.8}$$

Substituting Equation (4.7) into Equation (4.8) and using Equation (4.6) for s_1 and s_2, we have

$$\begin{aligned} p' &= \pi r' \left[\pi r' + \frac{\pi(1-p_1)r_1}{p_1} + \frac{\pi(r'-r_1)}{r_2 - r_1}\left(\frac{(1-p_2)r_2}{p_2} - \frac{(1-p_1)r_1}{p_1}\right)\right]^{-1} \\ &= r' \left[r' \left(1 + \frac{(1-p_2)r_2}{p_2(r_2-r_1)} - \frac{(1-p_1)r_1}{p_1(r_2-r_1)}\right) + \right. \\ &\qquad \left. \frac{(1-p_1)r_1}{p_1} - \frac{(1-p_2)r_1r_2}{p_2(r_2-r_1)} + \frac{(1-p_1)r_1^2}{p_1(r_2-r_1)}\right]^{-1} \end{aligned}$$

□

Thus, despite PR space being sensitive to π and the translation to and from ROC space depending on π, the interpolation in PR space *does not* depend on π. One explanation is that each particular PR space point inherently contains the information about π, primarily in the precision value, and no extra knowledge of π is required to perform the interpolation.

The area under the interpolated PR curve between these two points has a closed form.

Theorem 4.2. *The area under the interpolated PR curve from* r_1 *to* r_2 *defined in Theorem 4.1 is*

$$\frac{ar_2 - b\log(ar_2 + b) - ar_1 + b\log(ar_1 + b)}{a^2} \tag{4.9}$$

Proof. The proof is a simple application of calculus and definite integrals.

$$\begin{aligned}\int_{r_1}^{r_2} \frac{r'}{ar' + b}\, dr' &= \left.\frac{ar' - b\log(ar' + b)}{a^2}\right|_{r'=r_1}^{r'=r_2} \\ &= \frac{ar_2 - b\log(ar_2 + b) - ar_1 + b\log(ar_1 + b)}{a^2}\end{aligned}$$ □

With the definite integral to calculate the area between two PR points, the question is which points should be used? The achievable PR curve of Davis and Goadrich (2006) uses only those points (translated into PR space) that are on the ROC convex hull. We also use three methods of summarizing from multiple PR points at the same recall to a single PR point to interpolate through. The summaries we investigate are the max, mean, and median of all p_i for a particular r_i. So we have four estimators using interpolation: *convex*, *max*, *mean*, and *median*.

Average Precision

Avoiding the empirical curve altogether, a plug-in estimate of θ, known in information retrieval as *average precision* (Manning et al., 2008), is

$$\widehat{\theta}_A = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathrm{Prec}}(y_i) \tag{4.10}$$

which replaces the distribution function $P(Y \leqslant c)$ in Equation (4.2) with its empirical cumulative distribution function.

Binormal Estimator

Conversely, a fully parametric estimator may be constructed by assuming that $X_j \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y_j \sim \mathcal{N}(\mu_y, \sigma_y^2)$. In this *binormal* model (Brodersen

et al., 2010), the MLE of θ is

$$\widehat{\theta}_B = \int_0^1 \frac{\pi t}{\pi t + (1-\pi)\Phi\left(\frac{\hat{\mu}_y - \hat{\mu}_x}{\sigma_x} + \frac{\hat{\sigma}_y}{\hat{\sigma}_x}\Phi^{-1}(t)\right)}\, dt \tag{4.11}$$

where $\hat{\mu}_x, \hat{\sigma}_x, \hat{\mu}_y, \hat{\sigma}_y$ are the sample means and variances of X and Y and $\Phi(t)$ is the standard normal cumulative distribution function.

Confidence Interval Estimation

Having discussed AUCPR estimators, we now turn our attention to computing confidence intervals (CIs) for these estimators. Our goal is to determine a simple, accurate interval estimate that is logistically easy to implement. We will compare two computationally intensive methods against two simple statistical intervals.

Bootstrap Procedure

A common approach uses a *bootstrap* procedure to estimate the variation in the data and to either extend a symmetric, normal-based interval about the point estimate or to take the empirical quantiles from the resampled estimates as interval bounds (Efron, 1979). Because the relationship between the number of positive examples n and negative examples m is crucial for estimating PR points and hence curves, we recommend using stratified bootstrap so π is preserved in all replicates. In our simulations, we chose to use empirical quantiles for the interval bounds and perform 1000 bootstrap replicates.

Cross-Validation Procedure

Similarly, a *cross-validation* approach is a wholly data-driven method for simultaneously producing the train/test splits required for unbiased estimation of future performance and producing variance estimates. In

k-fold cross-validation, the available data are partitioned into k folds. $k-1$ folds are used for training while the remaining fold is used for testing. By evaluating the results of each fold separately, k estimates of performance are obtained. A normal approximation of the interval can be constructed using the mean and variance of the k estimates. For more details and discussion of k-fold cross-validation, see Dietterich (1998). For our case studies, we use the standard $k = 10$.

Binomial Interval

Recalling that $0 \leqslant \theta \leqslant 1$, we may interpret $\hat{\theta}$ as a probability associated with some Binomial$(1, \theta)$ variable. If so, a CI for θ can be constructed through the standard normal approximation to the binomial:

$$\hat{\theta} \pm \Phi_{1-\alpha/2}\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{n}}$$

We use n for the sample size as opposed to $n + m$ because n specifies the (maximum) number of unique recall values in $\widehat{PR}(\cdot)$. The *binomial* method can be applied to any $\hat{\theta}$ estimate once it is derived. A weakness of this estimate is that it may produce bounds outside of $[0, 1]$, even though $0 \leqslant \theta \leqslant 1$.

Logit Interval

To obtain an interval that is guaranteed to produce endpoints within $[0, 1]$, we may use the logistic transformation $\hat{\eta} = \log \frac{\hat{\theta}}{(1-\hat{\theta})}$ where $\hat{\tau} = s.e.(\hat{\eta}) = (n\hat{\theta}(1-\hat{\theta}))^{-1/2}$ by the delta method (DeGroot and Schervish, 2001).

On the logistic scale, an interval for η is $\hat{\eta} \pm \Phi_{1-\alpha/2}\hat{\tau}$. This can be converted pointwise to produce an asymmetric *logit* interval bounded in $(0, 1)$:

$$\left[\frac{e^{\hat{\eta}-\Phi(1-\alpha/2)\hat{\tau}}}{1+e^{\hat{\eta}-\Phi(1-\alpha/2)\hat{\tau}}}, \frac{e^{\hat{\eta}+\Phi(1-\alpha/2)\hat{\tau}}}{1+e^{\hat{\eta}+\Phi(1-\alpha/2)\hat{\tau}}}\right].$$

4.4 Case Studies

We use simulated data to evaluate the merits of the candidate point and interval estimates discussed in Section 4.3 with the goal of selecting a subset of desirable procedures.[1] The ideal point estimate is unbiased, robust to various distributional assumptions on X and Y, and has good convergence as $n + m$ increases. A CI should have appropriate coverage, and smaller widths of the interval are preferred over larger widths.

We consider three scenarios for generating output values. We intend to cover representative but not exhaustive cases whose conclusions will be relevant more generally. The densities for these scenarios are plotted in Figure 4.3. The true PR curves (calculated using the cumulative distribution functions of X and Y) for $\pi = 0.1$ are shown in Figure 4.4. Figure 4.4 also contains sampled empirical PR curves that result from drawing data from X and Y. These are the curves the estimators work from, attempting to recover the area under the true curve as accurately as possible.

For unbounded, continuous outputs, the binormal scenario assumes that $X \sim \mathcal{N}(0, 1)$ and $Y \sim \mathcal{N}(\mu, 1)$ where $\mu > 0$. The distance between the two normal distributions, μ, controls the discriminative ability of the assumed model. For test values bounded by $[0, 1]$, such as probabilistic outputs, we replace the normal distribution with a beta distribution. Therefore, the bibeta scenario has $X \sim \text{Beta}(a, b)$ and $Y \sim \text{Beta}(b, a)$ where $0 < a < b$. The larger the ratio between a and b, the better we are able to distinguish between positive and negative examples. Finally, we model an extreme scenario where the support of X and Y is not the same. This offset uniform scenario is given by $X \sim \text{Uniform}(0, 1)$ and $Y \sim \text{Uniform}(\gamma, 1 + \gamma)$ for $\gamma \geqslant 0$. If $\gamma = 0$ there is no ability to discriminate, while $\gamma > 1$ leads to perfect classification of positive and negative examples with a threshold of $c = 1$. All results in this paper use $\mu = 1$, $a = 2$, $b = 5$, and $\gamma = 0.5$. These

[1] R code for the estimators and simulations may be found at `http://pages.cs.wisc.edu/~boyd/projects/2013ecml_aucprestimation/`

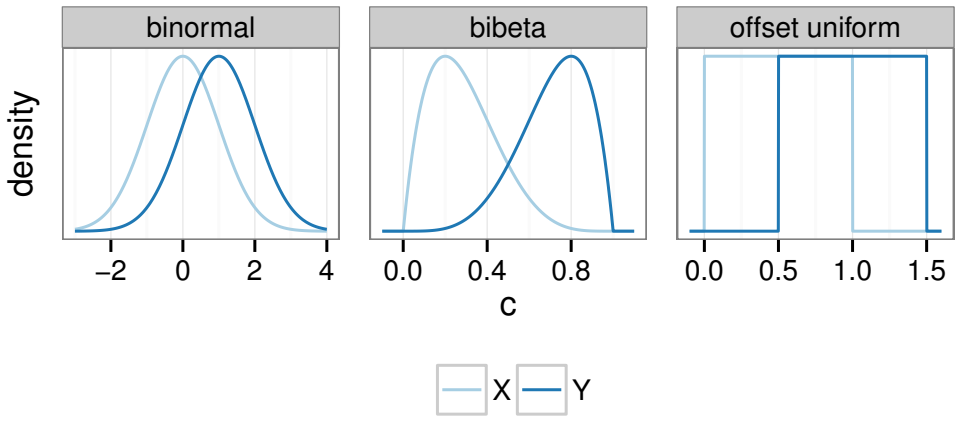


Figure 4.3: Probability density functions for X (negative) and Y (positive) output values for binormal ($X \sim \mathcal{N}(0,1), Y \sim \mathcal{N}(1,1)$), bibeta ($X \sim \text{Beta}(2,5), Y \sim \text{Beta}(5,2)$), and offset uniform ($X \sim \text{Uniform}(0,1), Y \sim \text{Uniform}(0.5,1.5)$) case studies.

were chosen as representative examples of the distributions that produce reasonable PR curves.

This chapter exclusively uses simulated data drawn from specific, known distributions because this allows calculation of the true PR curve (shown in Figure 4.4) and the true AUCPR. Therefore, we have a target value to compare the estimates against and we are able to evaluate the bias of an estimator and the coverage of a CI. This analysis would be difficult or impossible if we used a model's predictions on real data because the true PR curve and AUCPR are unknown.

Bias and Robustness in Point Estimates

For each scenario, we evaluate eight estimators: the nonparametric *average precision*, the parametric *binormal*, two trapezoidal estimates, and four interpolated estimates. Figure 4.5 shows the bias ratio versus $n + m$ where

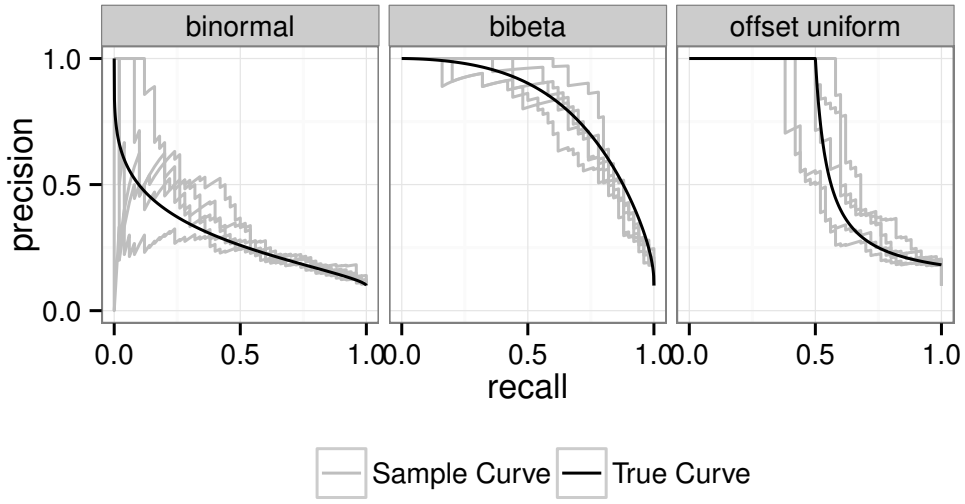


Figure 4.4: True PR curves (calculated from the theoretical density functions) and sampled empirical PR curves at $\pi = 0.1$. Sampled PR curves use $n + m = 500$. The sampled PR curves were generated by connecting PR points corresponding to adjacent thresholds.

$\pi = 0.1$ over 10,000 simulations, and Figure 4.6 shows the bias ratio versus π where $n + m = 1000$. The bias ratio is the mean of the estimated AUCPR divided by the true AUCPR, so an unbiased estimator has a bias ratio of 1.0. Good point estimates of AUCPR should be unbiased as $n + m$ and π increase. That is, an estimator should have an expected value equal to the true AUCPR (calculated by numerically integrating Equation (4.2)).

As $n + m$ grows large, most estimators converge to the true AUCPR in every case. However, the *binormal* estimator shows the effect of model misspecification. When the data are truly binormal, it shows excellent performance, but when the data are bibeta or offset uniform, the *binormal* estimator converges to the wrong value. Interestingly, the bias due to misspecification that we observe for the *binormal* estimate is lessened as

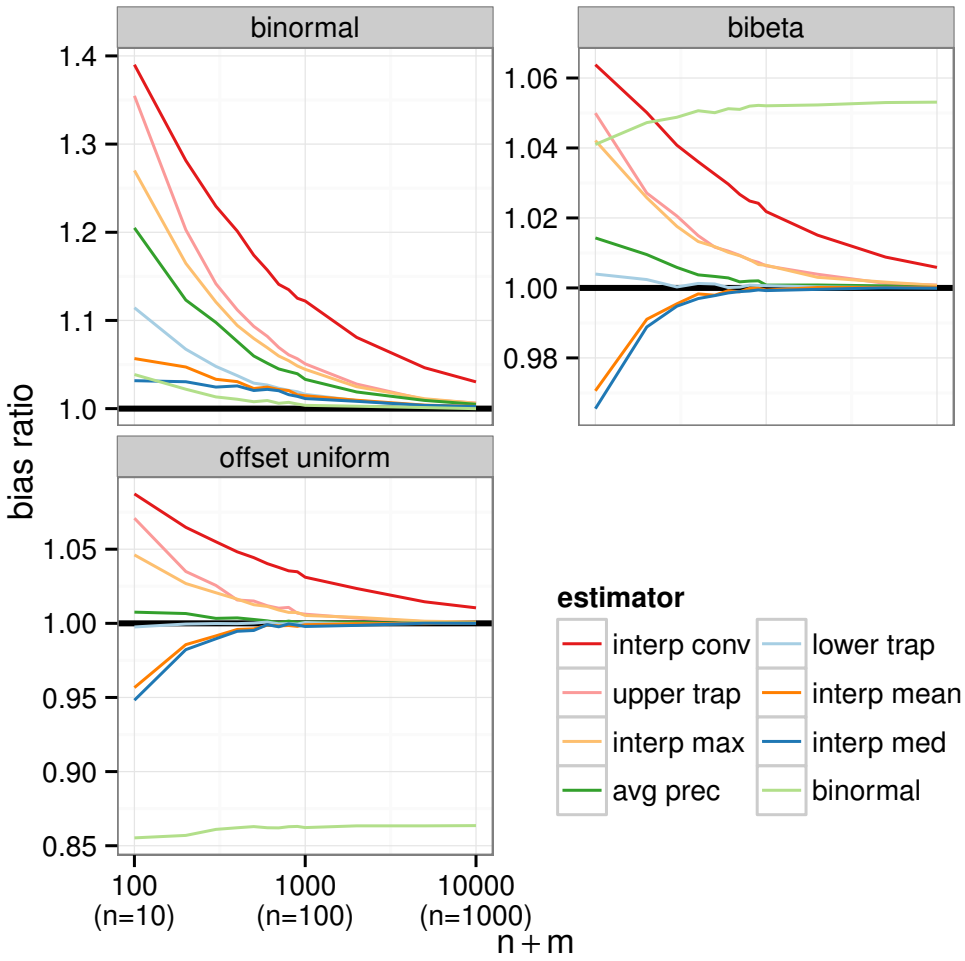


Figure 4.5: Ratio of estimated AUCPR to true AUCPR (bias ratio) versus total number of examples ($n + m$). $\pi = 0.1$ for all cases.

the data become more balanced (π increases).

As predicted by Davis and Goadrich (2006), the *upper trapezoid* estimator consistently overestimates the true AUCPR. Surprisingly, the only estimator that is always higher than the *upper trapezoid* method is the *interpolated convex* estimator. Even when $n + m = 10,000$, the *interpolated*

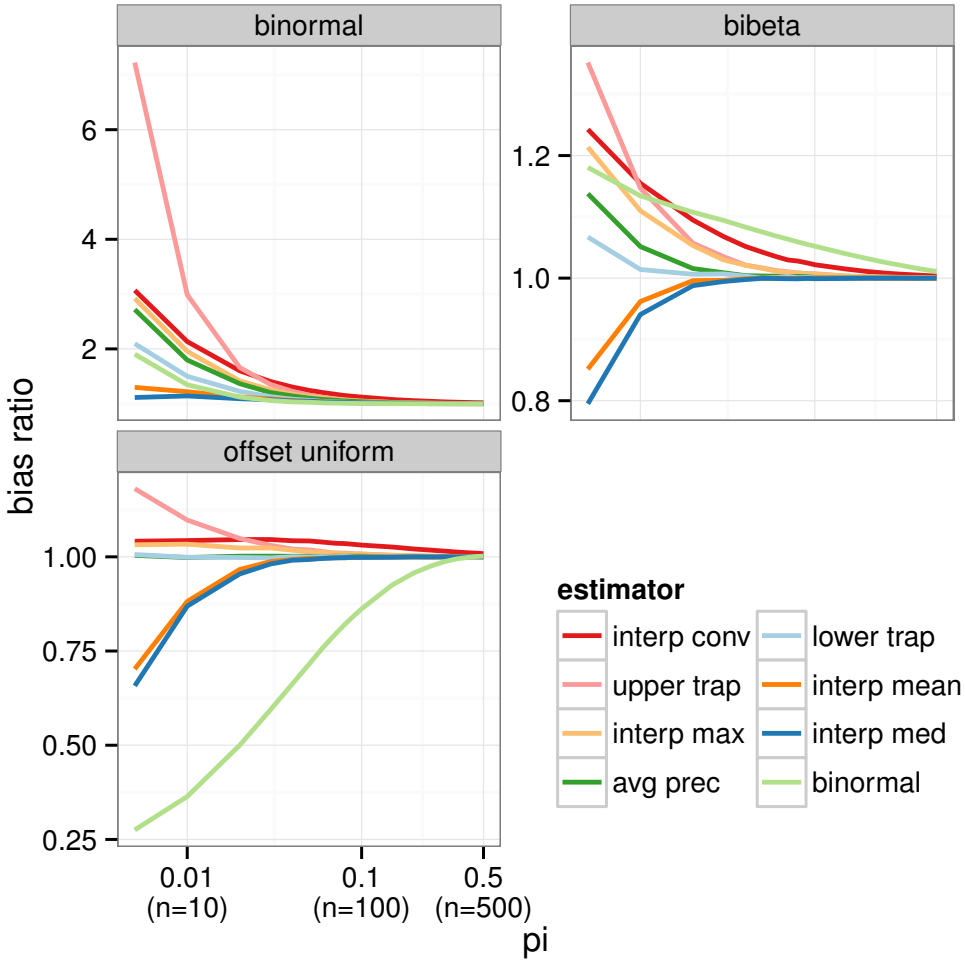


Figure 4.6: Ratio of estimated AUCPR to true AUCPR (bias ratio) versus π. In all cases $n + m = 1000$.

convex estimator is still far from the true value. The poor performance of the *interpolated convex* estimator is unusual because it uses the popular convex hull ROC curve and then converts back to PR space. Because the other interpolated estimators perform adequately, the problem may lie in evaluating the convex hull in ROC space. The convex hull chooses

those particular points that give the best performance on the *test* set. The convex hull procedure is analogous to using the test set during training, causing potential overfitting to the test set and leading to the observed overestimation of AUCPR.

It is important to note that since $\pi = 0.1$ in Figure 4.5, data are sparse at $n + m = 100$; there are only $n = 10$ values of Y from which to make the estimate. In these situations, there is no clear winner across the three scenarios. The estimators tend to overestimate AUCPR when n is small, with a few exceptions where AUCPR is substantially underestimated. Among related estimators, *lower trapezoid* is more accurate than the *upper trapezoid* method and the *mean* or *median interpolation* estimators outperform the *convex* and *max interpolation* estimators. Consequently, we will only consider the *average precision*, *interpolated median*, and *lower trapezoid* estimators because they are unbiased in the limit, less biased for small sample sizes, and robust to model misspecification.

Confidence Interval Evaluation

We use a two-step approach to evaluate confidence intervals (CIs) based on Chapter 7 of Shao (2003). In practice, interval estimates must come with a confidence guarantee: if we say an interval is a $(1-\alpha)\%$ CI, we should be assured that it covers the true value in at least $(1-\alpha)\%$ of data sets (Shao, 2003; Wasserman, 2004; Lehmann and Casella, 1998). It may be surprising to non-statisticians that an interval with slightly low coverage is ruled inadmissible, but this invalidates the guarantee. Additionally, targeting an exact $(1-\alpha)\%$ interval is often impractical for technical reasons, hence the *at least* $(1-\alpha)\%$. A valid interval provides at least $(1-\alpha)\%$ coverage, and this is the first criterion a candidate interval must satisfy.

After identifying valid methods for CIs, the second step is determining the narrowest (or optimal) intervals among the valid methods. The trivial $[-\infty, +\infty]$ interval is a valid 95% CI because it always has at least 95%

coverage (indeed, it has 100% coverage), but it conveys no useful information about the estimate. We therefore seek methods that produce the narrowest, valid intervals.

Confidence Interval Coverage

The first step in CI evaluation is identifying valid CIs with coverage of at least $(1 - \alpha)\%$. In Figure 4.7, we show results of 10,000 simulations for the coverage of the four CI methods described in Section 4.3. These are 95% CIs, so the target coverage of 0.95 is denoted by the thick black line. As mentioned at the end of Section 4.4, we only consider the *average precision*, *interpolated median*, and *lower trapezoid* estimators during our CI evaluation.

A strong pattern emerges from Figure 4.7 where the *bootstrap* and *cross-validation* intervals tend to have coverage below 0.95, though asymptotically approaching 0.95. Because the coverage is below 0.95, the computational intervals are technically invalid. The two formula-based intervals are consistently above the requisite 0.95 level. Thus, the *binomial* and *logit* methods produce valid confidence intervals.

Given the widespread use of *cross-validation* within machine learning, it is troubling that the CIs produced from that method fail to maintain the confidence guarantee. This is not an argument against *cross-validation* in general, only a caution against using it for AUCPR inference. Similarly, *bootstrap* is considered a rigorous (though computationally intensive) fallback for nonparametrically evaluating variance, yet Figure 4.7 shows it is only successful asymptotically as data size increases, and it must be fairly large before the *bootstrap* nears 95% coverage).

Confidence Interval Width

To better understand why *bootstrap* and *cross-validation* are failing, we ask: are the intervals *too* narrow? Since we have simulated 10,000 data sets and

obtained AUCPR estimates on each using the various estimators, we have an empirical distribution from which we can calculate an ideal empirical width for the CIs. When creating a CI, only 1 data set is available, so this empirical width is not available, but we can use it as a baseline, optimal width. Figure 4.8 shows the coverage versus the ratio of mean width to empirically ideal width. As expected, there is a positive correlation between coverage and the width of the intervals. Wider intervals tend to provide higher coverage. For *cross-validation*, the widths tend to be slightly smaller than the *logit* and *binomial* intervals but still larger than the empirically ideal width. However, coverage is frequently too low, suggesting the width of the interval is not the reason for the poor performance of *cross-validation*. But interval width may be part of the issue with *bootstrap*. The *bootstrap* widths are either right at the empirically ideal width or even smaller.

Confidence Interval Location

Another possible cause for poor coverage is that the intervals are for the wrong target value (i.e., the intervals are biased). To investigate this possibility, we analyze the mean location of the intervals. We use the original estimate from the full data set as the location for the *binomial* and *logit* intervals because both intervals are constructed around that estimate, the mid-point of the interval from *cross-validation*, and the median of the *bootstrap* replicates since we use the quantiles to calculate the interval. The ratio of the mean location to the true value (similar to Figure 4.5) is presented in Figure 4.9. The location of the *cross-validation* intervals is much farther from the true estimate than either the *bootstrap* or *binomial* locations are, with *bootstrap* being a bit worse than *binomial*. We believe this targeting of the wrong value for small $n + m$ is the primary explanation for the low coverage of *bootstrap* and *cross-validation* seen in Figure 4.7.

Comments on *Bootstrap* and *Cross-Validation* Intervals

The increased bias in the intervals produced by *bootstrap* and *cross-validation* occurs because these methods use many smaller data sets to produce a variance estimate. k-fold *cross-validation* reduces the effective data set size by a factor of k, while *bootstrap* is less extreme but still reduces the effective size by a factor of roughly 1.5. Since the estimators become more biased with smaller data sets (as demonstrated in Figure 4.5), the point estimates used to construct the *bootstrap* and *cross-validation* intervals are more biased, leading to the misplaced intervals and less than $(1-\alpha)\%$ coverage.

Additionally, the *bootstrap* has no small sample theoretical justification and tends to break down for very small sample sizes (Efron, 1988). When estimating AUCPR with skewed data, the critical number is the number of positive examples n, not the size of the data set $n+m$. Even when the data set itself seems reasonably large with $n+m=200$, there are only $n=20$ positive examples if $\pi=0.1$. With just 20 samples, it is difficult to obtain representative samples during the *bootstrap*. The small sample size contributes to the lower than expected 95% coverage and presents a possible explanation for the *bootstrap* widths being even smaller than the empirically ideal widths seen in Figure 4.8.

We emphasize that both the *binomial* and *logit* intervals are valid and do not require the additional computation that the *cross-validation* and *bootstrap* intervals do. For large sample sizes *bootstrap* approaches $(1-\alpha)\%$ coverage, but it approaches from below, so care should be taken. *Cross-validation* is even more problematic, with proper coverage not obtained even at $n+m=10,000$ for some of our case studies.

4.5 Chapter Summary

Our computational study has determined that simple confidence interval estimators can achieve nearly ideal interval widths while maintaining valid coverage for AUCPR estimation. A key point is that these simple estimates are easily evaluated and do not require resampling nor do they add to the computational workload. Conversely, computationally expensive, empirical procedures (*bootstrap* and *cross-validation*) yield interval estimates that do not provide adequate coverage for small sample sizes and only asymptotically approach $(1-\alpha)\%$ coverage.

We have also tested a variety of point estimates for AUCPR and we determined that the parametric *binormal* estimate is extremely poor when the true generating distribution is not normal. Practically, data may be re-scaled (e.g., the Box-Cox transformation) to make this assumption fit better, but this seems unnecessary because robust, easily accessible, non-parametric estimates exist.

The scenarios we studied are by no means exhaustive, but they are representative, and the conclusions can be further tested in specific cases if necessary. In summary, our investigation concludes that the *lower trapezoid*, *average precision*, and *interpolated median* point estimates are the most robust estimators and we recommend the *binomial* and *logit* methods for constructing interval estimates.

This chapter concludes the section focused on PR space, where we discussed the unachievable region in Chapter 3 and estimators of AUCPR in this chapter. Next, we consider the data from which AUCROCs and AUCPRs are generated and how to protect the privacy of those test sets while still releasing useful performance assessments.

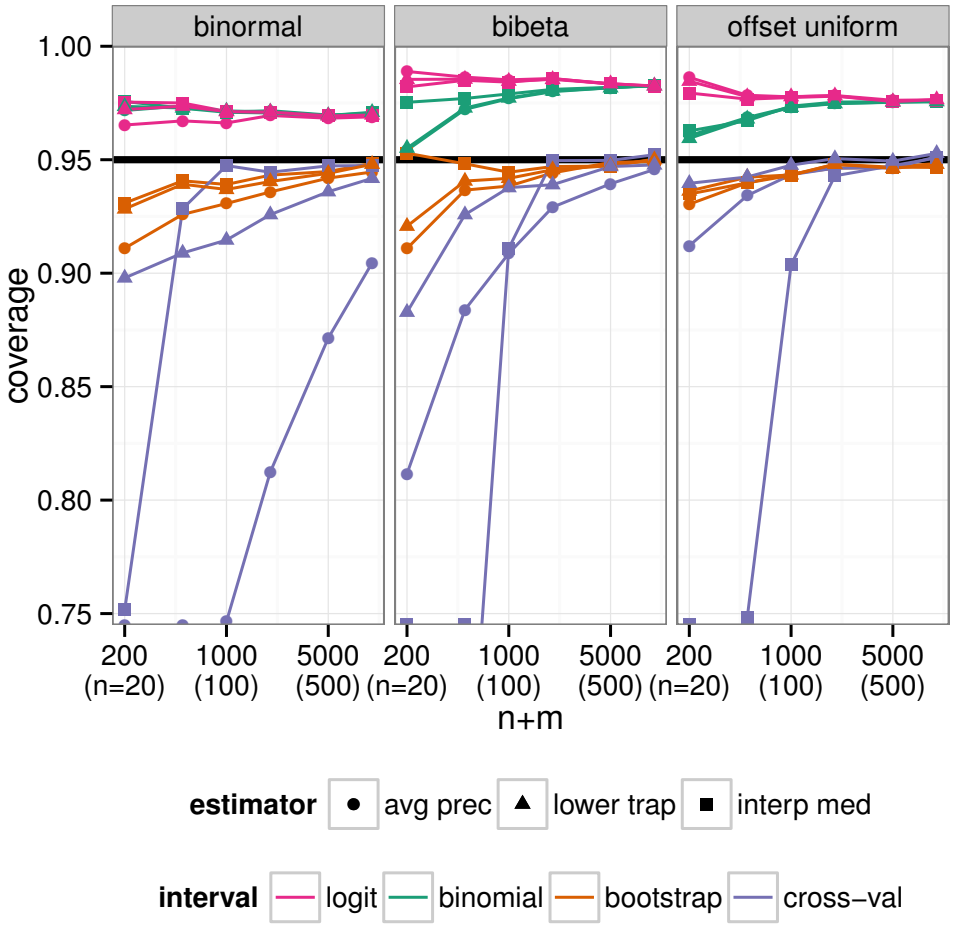


Figure 4.7: Coverage for selected estimators and 95% CIs calculated using the four interval methods. Results for selected $n+m$ are shown for $\pi = 0.1$. To be valid 95% CIs, the coverage should be at least 0.95. Note that the coverage for a few of the *cross-validation* intervals is below 0.75. These points are represented as half-points along the bottom border.

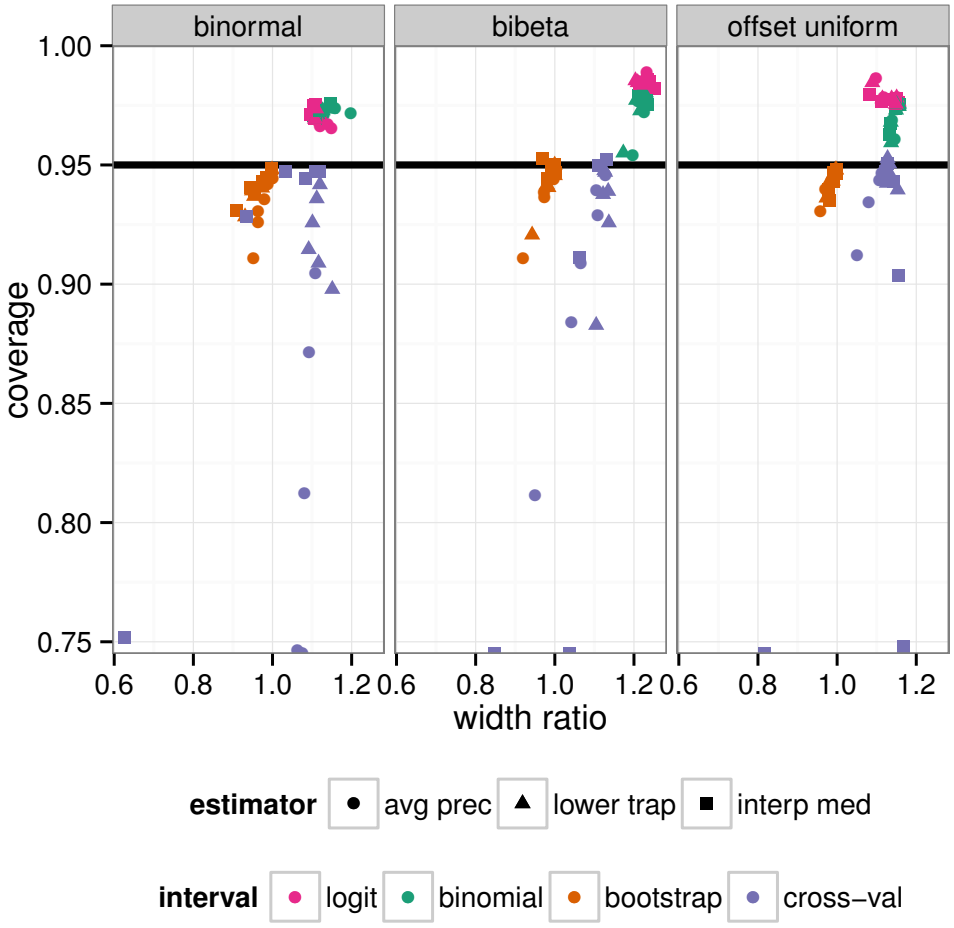


Figure 4.8: Mean normalized width ratio versus coverage for *binomial*, *logit*, *cross-validation*, and *bootstrap* methods. Normalized width is the ratio of the CI width to the empirically ideal width. Width ratios below 1 suggest the intervals are overly optimistic. Results shown for $n + m \in 200, 500, 1000, 5000, 10000$ with $\pi = 0.1$. Note that the coverage for some of the *cross-validation* intervals is below 0.75. These points are represented as half-points along the bottom border.

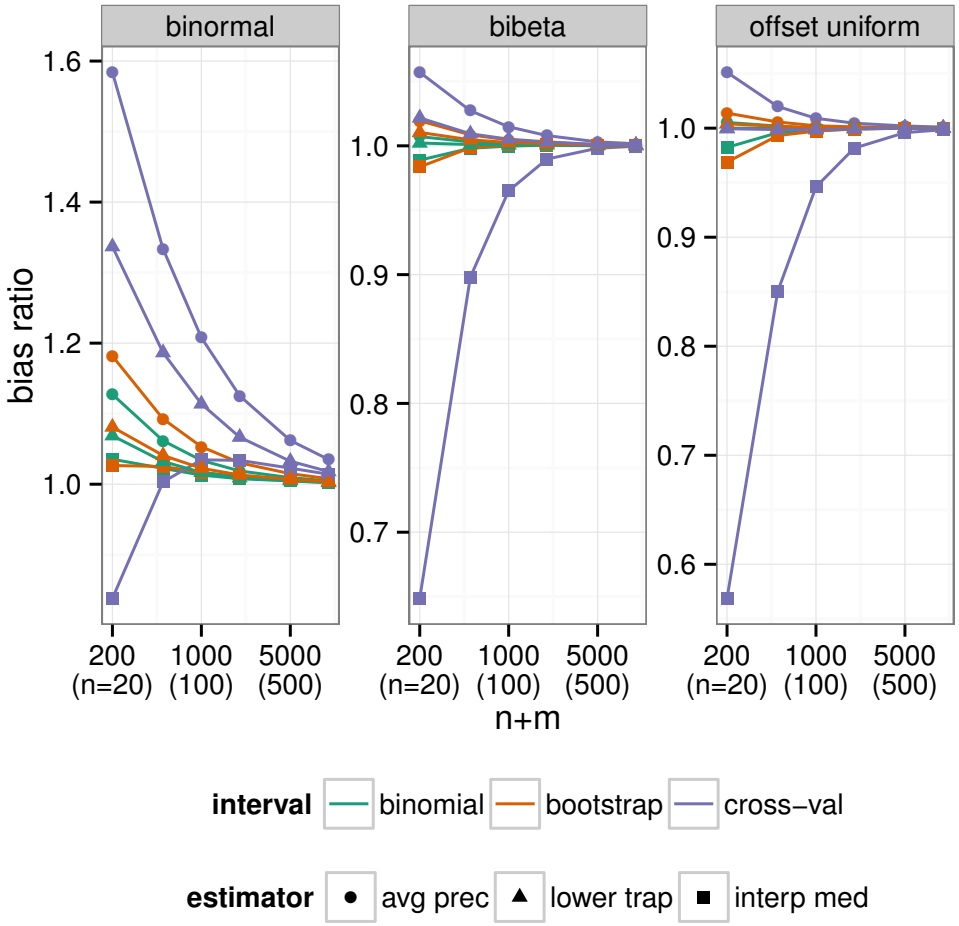


Figure 4.9: Mean location of the intervals produced by the *binomial*, *bootstrap*, and *cross-validation* methods (*logit* is identical to *binomial*). As in Figure 4.5, the y-axis is the bias ratio, the ratio of the location (essentially a point estimate based on the interval) to the true AUCPR. *Cross-validation* is considerably more biased than the other methods are and *bootstrap* is slightly more biased than *binomial* is.

5 DIFFERENTIALLY PRIVATE EVALUATION

After discussing properties and estimators of PR curves in the previous chapters, we consider the data upon which evaluation is performed in this chapter. As data sets and models leverage increasing amounts of information about the examples, there is growing concern about privacy. In particular, privacy is a concern when the examples are people, as is the case in social networking and medical diagnosis tasks. In this chapter, we address the privacy of the test set data.

5.1 Introduction

Our aim in this chapter is to expand the scope of differential privacy in machine learning to include the protection of test data sets beyond the existing work on the protection of training data sets. To our knowledge, this is the first time the privacy of evaluating models, even differentially private models, on new data and the added privacy risk involved has been addressed.

We start by motivating our application of differential privacy to evaluation by discussing potential attacks on ROC analysis in Section 5.2. In Section 5.3, we define the task of differentially private evaluation, and then discuss differentially private algorithms for AUCROC and average precision in Sections 5.4 and 5.5. Finally, in Section 5.6, we perform experiments analyzing the utility and behavior of these algorithms.

5.2 Attacks on ROC Curves and AUCROC

Prior work has demonstrated the vulnerability of data points in ROC curves to reidentification (Matthews and Harel, 2013); we extend that to AUCROC to demonstrate that the problem remains even with the summary

metric. Given the AUCROC of the full data set, consider the problem of identifying the class of one of the examples. Here, the adversary has access to all of the data points but one, and also knows the AUCROC of the full data set. The goal is to predict whether the final example is a member of the positive or negative class. Note that we do not assume the adversary knows where the target example should go in the ranking.

The adversary's algorithm is to attempt to place the target example at each position in the ranking, and calculate the resulting AUCROC under the assumption that the example is positive and again assuming it is negative. The class that produces an answer closest to the released AUCROC for the full data set (or the most frequent class in the case of ties) is guessed as the class of the example. This setup is similar to the assumptions of differential privacy in terms of looking at the influence of a single example on AUCROC. However, it is not a worst case analysis and it is concerned with identifying an attribute value of the target example, not simply its presence in the original data.

Figure 5.1 shows the ability of the attacker to guess the class of the target example given a sample of data from the UCI `adult` data set (details of the data set are discussed in Section 5.6). One heuristic method that could be used to interfere with this attack is to round the released AUCROC to a smaller number of decimal places, and this is illustrated in Figure 5.1. When the AUCROC is given to a high number of decimal places, the adversary is able to recover the class value with high probability, though this ability decreases as the number of data points increases. Rounding the AUCROC value to fewer decimal places does reduce the adversary's success, but it comes at a cost to precision.

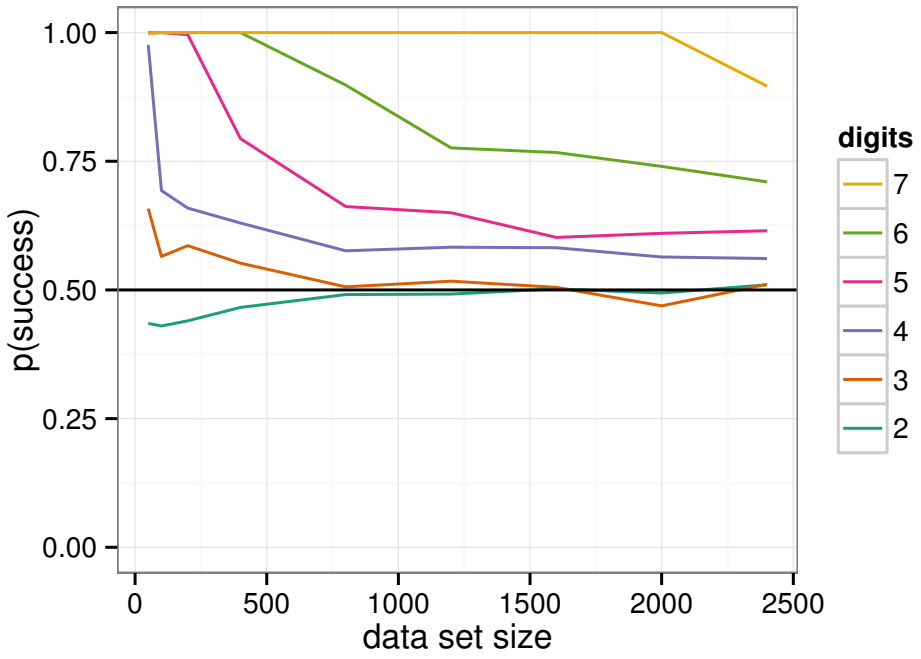


Figure 5.1: Adversary's success rate at identifying the class of the missing example given AUCROC of a data set containing half positives and half negatives with the specified significant digits. The horizontal black line at 0.5 denotes performance of randomly guessing the class.

5.3 Private Evaluation

Our discussion of differentially private evaluation will assume that a classification model is applied to a private database. The model could be hand-constructed by the submitter, trained on another private database in a differentially private way, or trained on a public database. Our goal is to ensure the evaluation output does not release too much information about any particular example in the private database by requiring a differentially private evaluation function.

We assume that the size of the database, $N = n + m$, is public information, but that the specific values of n and m are not publicly available. Though allowing n and m to be public would make our analysis for AUCROC and AP simpler and might achieve induced neighbors privacy (Kifer and Machanavajjhala, 2014), we believe that keeping the number of positive and negative examples private is a critical aspect of private model evaluation. If n and m were public information, the worst-case adversary for differential privacy who knows all but one row of the database would be able to trivially infer whether the last row is a positive or negative. Since the class label is often the most sensitive piece of information in a prediction task, releasing precise counts of positives and negatives would greatly weaken the security provided by a privacy framework. Instead, we assume that only the size of the database, N, is public information. As discussed in Section 2.4, we are using bounded differential privacy, so neighboring databases always have the same number of examples or rows. However, neighboring databases may differ (by a maximum of 1) in the number of positives and number of negatives. To illustrate the difference in neighboring databases in ROC analysis, ROC curves for two neighboring databases are shown in Figure 5.2.

What types of evaluation metrics can be released privately under this framework? Any metric based on a single confusion matrix can be made private by applying the standard methods, such as Laplace noise, for differentially private counts or marginals (Dwork, 2006). Therefore, differentially private accuracy, recall, specificity, precision, etc. can be obtained. We focus on more complex metrics, such as AUCROC, that are both more useful for classifier evaluation (Provost et al., 1998) and more challenging to implement privately.

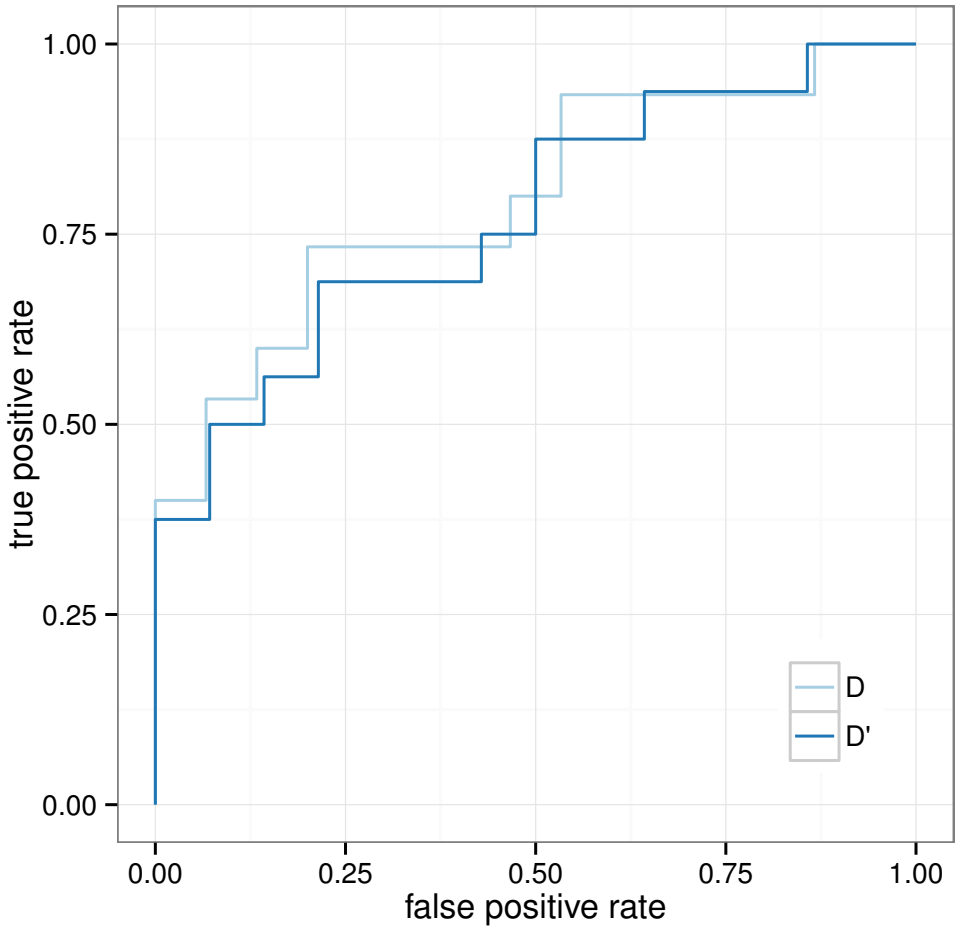


Figure 5.2: ROC curves for two neighboring databases where the difference between D and D′ is that a negative was changed to a positive and given a new score. D contains 15 positives and 15 negatives and D′ contains 16 positives and 14 negatives. AUCROCs for D and D′ are 0.796 and 0.772, respectively.

5.4 Private AUCROC

To create a private AUCROC algorithm, we first need to find the sensitivity of AUCROC. In particular, we will start with the local sensitivity (Definition 2.7) that provides database-specific bounds on the change in AUCROC.

We repeat the formula for AUCROC from Equation (2.1) for easy reference:

$$\text{AUCROC} = \frac{1}{nm} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{1}[x_i < y_j] \tag{5.1}$$

Looking at Equation (5.1) to calculate sensitivity of AUCROC, each example can contribute to the sum multiple times. The sensitivity of AUCROC is further complicated because the factor $\frac{1}{nm}$ differs between neighboring data sets when a positive example changes to a negative or vice versa. Fortunately, we can bound the maximum change in AUCROC between neighboring data sets to find the local sensitivity in Theorem 5.1. Note that we assume no ties in the scores assigned to positive and negative examples to simplify the proofs, i.e., we assume a total ordering for the ranking of examples from most to least likely to be positive. In case of ties, a complete ordering can be created where an arbitrary order is chosen within the tied negatives and tied positives and among scores with both negatives and positives, the negative examples are placed before the positive examples to avoid overestimation of the curves and areas.

Theorem 5.1. *Local sensitivity of the area under the ROC curve (AUCROC) is*

$$\text{LS}_{AUCROC}(n, m) = \begin{cases} \frac{1}{\min(n,m)} & \text{if } n > 1 \text{ and } m > 1 \\ 1 & \text{otherwise} \end{cases} \tag{5.2}$$

where n and m are the number of positive and negative examples in the test set, respectively.

Proof. Let D and D' be two neighboring databases that differ by exactly one row. Let n and m be the number of positive and negative examples in D.

We consider the four cases of moving a negative in the ranking, moving a positive, changing a positive to a negative (and moving it), and changing a negative to a positive. Our analysis of these four cases requires $n > 1$ and $m > 1$, so for completeness, we say the local sensitivity of AUCROC is 1 if either $n \leqslant 1$ or $m \leqslant 1$. Because the range of AUCROC is $[0, 1]$, the maximum change from neighboring databases is 1.

Case 1) Move negative: D' has the same x_i and y_j as D except for some x_k that is now x^* in D'. The only changes in Equation (5.1) occur when x_k is compared in the indicator functions. x_k appears n times and each time the indicator function can change by at most 1, so in this case, sensitivity is

$$\frac{n}{nm} = \frac{1}{m} \tag{5.3}$$

Case 2) Move positive: Similar to Case 1, D' is the same as D except for some y_k that changes to y^*. This y_k appears in Equation (5.1) m times, so the sensitivity is

$$\frac{m}{nm} = \frac{1}{n} \tag{5.4}$$

Case 3) Change negative to positive: Here, D' has $n + 1$ positive and $m - 1$ negative examples with the same x_i and y_j except for some x_k that has been removed and a new positive example with score y^* has been added. Note that we are only concerned with $m \geqslant 2$, so D' has at least 1 negative example. Without loss of generality, assume that $k = m$. Let C be the result of the sum for the indicator functions that remain the same between D and D'. Note that $0 \leqslant C \leqslant (m-1)n$. Using C to collect the identical

terms, we have

$$\text{AUCROC}(D) = \frac{1}{nm}(C + \sum_{j=1}^{n} \mathbb{1}[x_m < y_j]) \tag{5.5}$$

$$\text{AUCROC}(D') = \frac{1}{(n+1)(m-1)}(C + \sum_{i=1}^{m-1} \mathbb{1}[x_i < y^*]). \tag{5.6}$$

We need to bound the difference,

$$\begin{aligned} \text{AUCROC}(D) - \text{AUCROC}(D') &= (\frac{1}{nm} - \frac{1}{(n+1)(m-1)})C \\ &+ \frac{1}{nm}\sum_{j=1}^{n} \mathbb{1}[x_k < y_j] - \frac{1}{(n+1)(m-1)}\sum_{i=1}^{m-1} \mathbb{1}[x_i < y^*] \end{aligned} \tag{5.7}$$

$$\begin{aligned} &= \frac{m-n-1}{nm(n+1)(m-1)}C + \frac{1}{nm}\sum_{j=1}^{n} \mathbb{1}[x_m < y_j] \\ &- \frac{1}{(n+1)(m-1)}\sum_{i=1}^{m-1} \mathbb{1}[x_i < y^*] \end{aligned} \tag{5.8}$$

Equation (5.8) is maximized when each of the three terms is maximized. The first term is maximized when $m > n$ and $C = (m-1)n$,

$$\frac{m-n-1}{nm(n+1)(m-1)}C \leqslant \frac{m-n-1}{m(n+1)}. \tag{5.9}$$

The second and third terms are bounded above by $\frac{n}{nm}$ and 0, respectively. Putting it all together we have an upper bound of

$$\frac{m-n-1}{m(n+1)} + \frac{n}{nm} \leqslant \frac{m-n-1}{nm} + \frac{n}{nm} = \frac{m-1}{nm} \leqslant \frac{m}{nm} = \frac{1}{n}. \tag{5.10}$$

Similarly, the lower bound for Equation (5.8) occurs when $n > m$ and is

$$\frac{m-n-1}{m(n+1)} - \frac{m-1}{(n+1)(m-1)} = \frac{(m-1)(m-n-1) - m(m-1)}{(n+1)m(m-1)} \tag{5.11}$$

$$= \frac{-n-1}{m(n+1)} = -\frac{1}{m}. \tag{5.12}$$

Thus, for the case of changing a negative to a positive example, we have

$$LS_{AUCROC} \leqslant \max\left(\frac{1}{n}, \frac{1}{m}\right) = \frac{1}{\min(n,m)} \tag{5.13}$$

Case 4) Change positive to negative: Symmetric with Case 3, the result is the same as Equation (5.13).

Taking the maximum among all four cases we have

$$LS_{AUCROC} = \frac{1}{\min(n,m)} \tag{5.14}$$

as the local sensitivity for area under the ROC curve. □

Local sensitivity itself is not suitable for creating differentially private algorithms because adding different amounts of noise for adjacent databases can leak information (Nissim et al., 2007). Instead, we use β-smooth sensitivity which ensures that the scale of noise for adjacent databases is within a factor of e^{β}.

Theorem 5.2. *β-smooth sensitivity of the area under the ROC curve (AUCROC) is*

$$S^*_{AUCROC,\beta}(D) = \max_{0 \leqslant i \leqslant n+m} LS_{AUCROC}(i, n+m-i)e^{-\beta|i-n|} \tag{5.15}$$

where n and m are the number of positive and negative examples in D.

Proof. The proof is a straightforward application of the definition of β-smooth sensitivity. Let n' and m' be the number of positive and negative

examples in database D'. From Definition 2.2 of Nissim et al. (2007),

$$S^*_{\text{AUCROC},\beta}(D) = \max_{D' \in \mathbb{D}} \text{LS}_{\text{AUCROC}}(D') e^{-\beta d(D,D')}. \tag{5.16}$$

The smallest row difference between D and D' occurs if we just need to change the positive or negative labels on the minimal number of examples to ensure the n and m counts are correct, hence $d(D, D') \geqslant |n - n'|$. Then we have,

$$S^*_{\text{AUCROC},\beta}(D) = \max_{D' \in \mathbb{D}} \text{LS}_{\text{AUCROC}}(n', m') e^{-\beta|n-n'|} \tag{5.17}$$

$$= \max_{0 \leqslant i \leqslant n+m} \text{LS}_{\text{AUCROC}}(i, n+m-i) e^{-\beta|n-i|} \tag{5.18}$$

because there always exists some D' for which $d(D, D') = |n - n'|$. □

Figure 5.3 shows the smooth sensitivity given by Equation (5.15) for several database sizes and values of β. The advantages of small smooth sensitivity are only available with large β, large database size, and when neither positive nor negative examples are extremely rare.

With the β-smooth sensitivity of AUCROC, appropriately scaled Cauchy noise can be used to obtain ϵ-differential privacy or Laplace noise can be used to obtain (ϵ, δ)-differential privacy as described in Theorem 2.9. Because the range of AUCROC is $[0, 1]$, we truncate the output to be in that range. The truncation does not violate differential privacy because an adversary also knows the range of the true function (Ghosh et al., 2009).

5.5 Private Average Precision

Among the AUCPR estimators discussed in Chapter 4, average precision is one of the recommended estimators. AP is somewhat similar to AUCROC since it also uses sums and indicator functions for counting. This suggests we may be able to bound the change in AP between neighboring databases

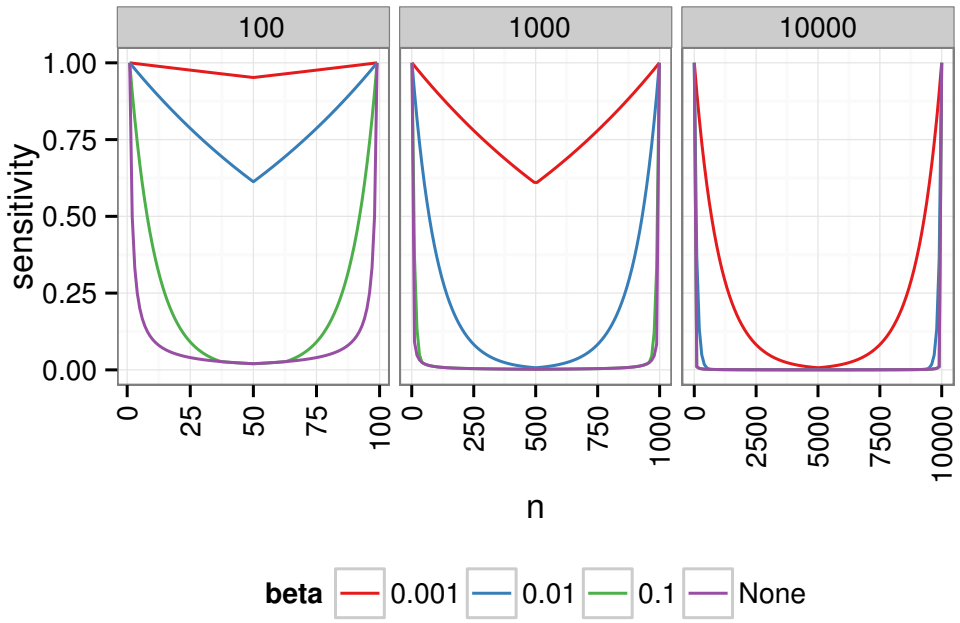


Figure 5.3: β-smooth sensitivity for AUCROC versus n, the number of positive examples in the database, for database sizes 100, 1000, and 10000. β of none indicates the original, non-smoothed local sensitivity.

as we did for AUCROC in Theorem 5.1. We use the following formulation for AP:

$$\mathrm{AP} = \frac{1}{n}\sum_{j=1}^{n}\frac{j}{j+\sum_{i=1}^{m}\mathbb{1}[x_i > y_j]} \tag{5.19}$$

where x_i for $1 \leqslant i \leqslant m$ are the scores on the negative examples in the test set and y_j for $1 \leqslant j \leqslant n$ are the scores on the positive examples. Additionally, we assume that the y_j's (but not the x_i's) are sorted, i.e., $y_1 > y_2 > \ldots > y_n$.

Precision at low recall has high variance because changing just a single row for neighboring data sets can cause precision to go from 1 to $\frac{1}{2}$ simply by adding a high-scoring negative example. Though precision at low

recalls can vary substantially between neighboring data sets, the impact on average precision is mitigated by the $\frac{1}{n}$ coefficient in Equation (5.19) and the sensitivity is bounded in the following theorem.

Theorem 5.3. *Local sensitivity of average precision (AP) is*

$$\mathrm{LS}_{AP} = \begin{cases} \max\left(\frac{\log(n+1)}{n}, \frac{9+\log(n-1)}{4(n-1)}\right) + \max\left(\frac{\log(n+1)}{n}, \frac{9+\log n}{4n}\right) & \text{if } n > 1 \\ 1 & \text{if } n \leqslant 1 \end{cases} \tag{5.20}$$

where n is the number of positive examples in the test set.

Proof. Let $x_1, x_2, ..., x_m$ and $y_1, y_2, ..., y_n$ be the classifier scores on the m negative and n positive examples for a data set D. To bound the maximum change in AP between D and a neighboring database, we consider the four cases of adding or removing a positive example and adding or removing a negative example. Once we have characterized adding and removing each type of example, we consider the combination of adding and removing in sequence to find the local sensitivity when the size of the database remains the same.

The rest of this proof will assume $n > 1$, so for $n \leqslant 1$ we default to a local sensitivity of 1, which encompasses the maximum range from 0 to 1 of AP [1].

Case 1) Remove positive: Assume WLOG that $y_1 > y_2 > ... > y_n$. Consider making D' by removing a positive example y_z. Separating out the different parts of the AP sum to facilitate comparison between D and D', we have

$$\mathrm{AP}(D) = \frac{1}{n}\left[\sum_{i=1}^{z-1} \frac{i}{i+s_i} + \frac{z}{z+s_z} + \sum_{i=z+1}^{n} \frac{i}{i+s_i}\right] \tag{5.21}$$

[1] Though there is a non-zero minimum AP for any particular choice of n and m, the minimum AP approaches 0 as $\frac{n}{m} \to 0$.

where $s_i = \sum_{j=1}^{m} \mathbb{1}[x_j > y_i]$. Removing the y_z example for D' yields

$$\mathrm{AP}(D') = \frac{1}{n-1}\left[\sum_{i=1}^{z-1}\frac{i}{i+s_i} + \sum_{i=z+1}^{n}\frac{i-1}{i-1+s_i}\right]. \tag{5.22}$$

We need to bound $|\mathrm{AP}(D) - \mathrm{AP}(D')|$, so we start by aligning like terms from Equations (5.21) and (5.22).

$$\begin{aligned}\mathrm{AP}(D) - \mathrm{AP}(D') = \frac{1}{n(n-1)}\Bigg[&\sum_{i=1}^{z-1}\left(\frac{(n-1)i}{i+s_i} - \frac{ni}{i+s_i}\right) + \frac{(n-1)z}{z+s_z} \\ &+ \sum_{i=z+1}^{n}\frac{(n-1)i}{i+s_i} - \frac{n(i-1)}{i-1+s_i}\Bigg]\end{aligned} \tag{5.23}$$

$$\begin{aligned}= \frac{1}{n(n-1)}\Bigg[&\sum_{i=1}^{z-1}\frac{-i}{i+s_i} + \frac{(n-1)z}{z+s_z} \\ &+ \sum_{i=z+1}^{n}\frac{(n-1)i(i-1+s_i) - n(i-1)(i+s_i)}{(i+s_i)(i-1+s_i)}\Bigg]\end{aligned} \tag{5.24}$$

$$\begin{aligned}= \frac{1}{n(n-1)}\Bigg[&\sum_{i=1}^{z-1}\frac{-i}{i+s_i} + \frac{(n-1)z}{z+s_z} \\ &+ \sum_{i=z+1}^{n}\frac{i - i^2 - is_i + ns_i}{(i+s_i)(i-1+s_i)}\Bigg]\end{aligned} \tag{5.25}$$

$$\begin{aligned}= \frac{1}{n(n-1)}\Bigg[&\sum_{i=1}^{z-1}\frac{-i}{i+s_i} + \frac{(n-1)z}{z+s_z} \\ &+ \sum_{i=z+1}^{n}\frac{-i}{i+s_i} + \sum_{i=z+1}^{n}\frac{ns_i}{(i+s_i)(i-1+s_i)}\Bigg]\end{aligned} \tag{5.26}$$

The two sums of $\frac{-i}{i+s_i}$ in Equation (5.26) include all i's except $i = z$. So we

can add and subtract $\frac{z}{z+s_z}$ to get,

$$\begin{aligned} AP(D) - AP(D') = \frac{1}{n(n-1)} \Bigg[\frac{nz}{z+s_z} + \sum_{i=1}^{n} \frac{-i}{i+s_i} \\ + \sum_{i=z+1}^{n} \frac{ns_i}{(i+s_i)(i-1+s_i)} \Bigg]. \end{aligned} \tag{5.27}$$

To find $|AP(D) - AP(D')|$, we maximize the absolute value of each term in Equation (5.27) separately. The first term is maximized when $s_z = 0$, so

$$\left| \frac{z}{(n-1)(z+s_z)} \right| \leqslant \frac{1}{n-1}. \tag{5.28}$$

The second term is maximized when $s_i = 0 \; \forall \; i$,

$$\left| \frac{1}{n(n-1)} \sum_{i=1}^{n} \frac{-i}{i+s_i} \right| \leqslant \frac{n}{n(n-1)} = \frac{1}{n-1}. \tag{5.29}$$

For the third term, the values of s_i that maximize the sum depend on i,

$$\left| \frac{1}{n-1} \sum_{i=z+1}^{n} \frac{s_i}{(i+s_i)(i-1+s_i)} \right| \leqslant \left| \frac{1}{n-1} \sum_{i=z+1}^{n} \frac{s_i}{(i-1+s_i)^2} \right|. \tag{5.30}$$

For a simpler analysis, we use the relaxation in Equation (5.30). We need to maximize $\frac{s_i}{(i-1+s_i)^2}$ for each i where s_i is free to take any (integer) value between 0 and m. Taking the derivative of $f(x) = \frac{x}{(i-1+x)^2}$, setting it to 0, and then solving for x, we find that $f(x)$ is maximized when $x = i-1$. Since i is an integer and $i > 1$, this means that the maximizer $s_i = i-1$ is always a valid choice for s_i, which gives an upper bound of

$$\frac{1}{n-1} \sum_{i=z+1}^{n} \frac{i-1}{(i-1+i-1)^2} = \frac{1}{4(n-1)} \sum_{i=z+1}^{n} \frac{1}{i-1}. \tag{5.31}$$

Since all terms of the sum in Equation (5.31) are positive ($z \geqslant 1$ so $i \geqslant 2$), it is maximized when there are as many terms as possible, i.e., when $z = 1$:

$$\frac{1}{4(n-1)} \sum_{i=z+1}^{n} \frac{1}{i-1} \leqslant \frac{1}{4(n-1)} \sum_{i=2}^{n} \frac{1}{i-1} = \frac{1}{4(n-1)} \sum_{j=1}^{n-1} \frac{1}{j}. \tag{5.32}$$

The final sum in Equation (5.32) is simply the $(n-1)$st harmonic number, H_{n-1}. Therefore, an upper bound for the third term in Equation (5.27) is

$$\frac{H_{n-1}}{4(n-1)}. \tag{5.33}$$

Combining the three terms from Equations (5.28), (5.29) and (5.33) to bound Equation (5.27), we have

$$LS_{AP} = \frac{2}{n-1} + \frac{H_{n-1}}{4(n-1)} = \frac{8 + H_{n-1}}{4(n-1)} \tag{5.34}$$

Case 2) Add positive: Equivalent to Case 1, but if D has n positive examples, then D' has $n+1$, so the sensitivity is

$$LS_{AP} = \frac{8 + H_n}{4n}. \tag{5.35}$$

Case 3) Remove negative: Consider making D' by removing a negative example x_k.

$$AP(D) = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{i + s_i} \tag{5.36}$$

$$AP(D') = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{i + s_i + \delta_i} \tag{5.37}$$

where $s_i = \sum_{j=1}^{m} \mathbb{1}[x_j > y_i]$ and $\delta_i = -\mathbb{1}[x_k > y_i]$ is the change in false

positive counts between D and D′. The difference in AP is

$$\mathrm{AP}(D) - \mathrm{AP}(D') = \frac{1}{n}\sum_{i=1}^{n} \frac{i}{i+s_i} - \frac{i}{i+s_i+\delta_i} \tag{5.38}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{i(i+s_i+\delta_i) - i(i+s_i)}{(i+s_i)(i+s_i+\delta_i)} \tag{5.39}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{i\delta_i}{(i+s_i)(i+s_i+\delta_i)}. \tag{5.40}$$

$\delta_i \in \{0, -1\}$, so the absolute value of Equation (5.40) is maximized when $\delta_i = -1$ and $s_i = 1 \; \forall \; i$ ($s_i = 1$ and not 0 because there must be an existing false positive to remove).

$$|\mathrm{AP}(D) - \mathrm{AP}(D')| \leqslant \frac{1}{n}\sum_{i=1}^{n} \frac{i}{(i+1)i} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{i+1} \tag{5.41}$$

This is again a harmonic sum (minus the first term), so

$$\mathrm{LS}_{\mathrm{AP}} = \frac{H_{n+1} - 1}{n}. \tag{5.42}$$

Case 4) Add negative: If we add a negative example instead of removing it, we again get to Equation (5.40), but now $\delta_i \in \{0, 1\}$, and the absolute value is maximized when $\delta_i = 1$ and $s_i = 0 \; \forall \; i$.

$$|\mathrm{AP}(D) - \mathrm{AP}(D')| \leqslant \frac{1}{n}\sum_{i=1}^{n} \frac{i}{i(i+1)} = \frac{H_{n+1} - 1}{n}. \tag{5.43}$$

Therefore the sensitivity for adding a negative example is the same as for removing a negative.

With bounds for each of the four cases, we can find the sensitivity of AP for changing a single row in a database. Changing a row is equivalent

to adding and removing a row (or vice versa), so the sensitivity is bounded by the sum of sensitivities of adding and removing an example. Thus,

$$\mathrm{LS}_{\mathrm{AP}} = \max\left(\frac{H_{n+1}-1}{n}, \frac{8+H_{n-1}}{4(n-1)}\right) + \max\left(\frac{H_{n+1}-1}{n}, \frac{8+H_n}{4n}\right) \tag{5.44}$$

is our tightest bound on the sensitivity. We can remove the dependence on the harmonic numbers by using the fact that $H_n < 1 + \log n$:

$$\begin{aligned}\mathrm{LS}_{\mathrm{AP}} = \max&\left(\frac{\log(n+1)}{n}, \frac{9+\log(n-1)}{4(n-1)}\right) \\ &+ \max\left(\frac{\log(n+1)}{n}, \frac{9+\log n}{4n}\right)\end{aligned} \tag{5.45}$$

Tighter bounds exist for the harmonic numbers (Guo and Qi, 2011; Qi and Guo, 2009), but we use this approximation for its simplicity. □

Note that the local sensitivity of AP depends only on the number of positive examples, n, and not the number of negative examples. This aligns with the notion that AP (and PR curves) focuses on the positives and does not give credit for true negatives.

Theorem 5.4. *β-smooth sensitivity of average precision (AP) is*

$$S^*_{AP,\beta} = \max_{0 \leqslant i \leqslant n+m} \mathrm{LS}_{AP}(i) e^{-\beta|i-n|} \tag{5.46}$$

Proof is virtually identical to that of Theorem 5.2.

As in AUCROC, we can use Cauchy or Laplace noise to produce ϵ- or (ϵ, δ)-differentially private outputs. As discussed in Chapter 3, the range of AP is not [0,1] because the minimum AP for any particular n and m is strictly greater than zero. Though the minimum AP can be sizable (about 0.3 when $n = m$), it depends on the non-public n and m, so we cannot

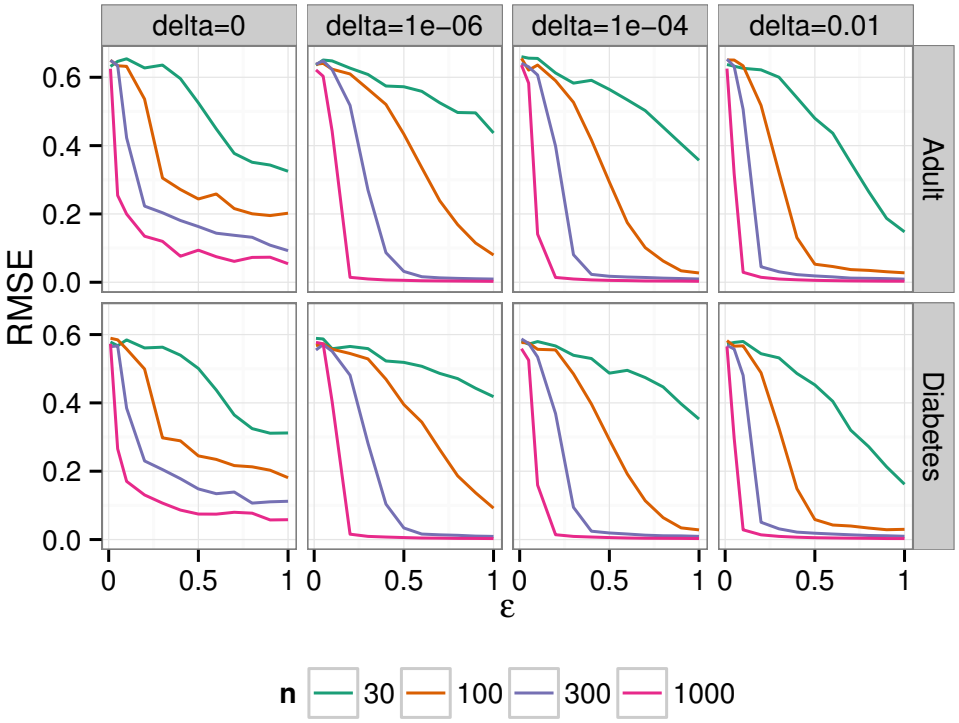


Figure 5.4: Root-mean-square error (RMSE) of (ϵ, δ)-differentially private AUCROC versus ϵ for several different data set sizes (n=m) using β-smooth sensitivity. The far left subplots use Cauchy noise, such that $\delta = 0$, and are ϵ-differentially private. The other subplots use Laplace noise with varying values of δ.

truncate to the database specific minimum AP and instead just truncate to the overall range of [0,1].

5.6 Experiments

In this section, we apply the algorithms from Sections 5.4 and 5.5 to two data sets. Since our mechanisms operate on the outputs of a classification

model and the true labels, they should not be influenced by the number of features in the original data set. The first data set is `adult` from the UCI repository (Bache and Lichman, 2013). It contains potentially private information in both the class label (yearly income greater or less than $50,000) and other features (e.g., capital gain/loss and work status) that individuals might be hesitant to provide without privacy guarantees. The data set has 14 features and 48,842 examples. The second data set is `diabetes` – a medical data set from a Kaggle competition[2] to predict diabetes from anonymized electronic health records. We processed the data set to include age, gender, and binary features for the 50 most common drugs, diagnoses, and laboratory tests for a total of 152 features. The data set contains 9,948 patients. Again, many of these features could be considered private information.

We imagine a scenario where an organization collects this information from individuals with the promise that all query responses will be differentially private. In these experiments, we trained a model on part of each data set using logistic regression. We perform differentially private evaluation on subsets of the rest of the data set. These subsets are a surrogate for a private test database. We investigate the accuracy of the differentially private evaluations with root-mean-square error (RMSE) between the differentially private output and the true answer as calculated directly from the private data.

Figure 5.4 shows the error of private AUCROC for several data set sizes. When $\delta = 0$, Cauchy noise is used as described in Theorem 2.9. This provides stronger privacy guarantees, but the RMSE approaches zero error more slowly as ϵ and n increase. For $\delta > 0$, Laplace noise is used for the relaxed (ϵ, δ)-differential privacy. As data set size or ϵ increases, utility improves as RMSE approaches 0. With 1000 each of positive and negative examples in the data set, reasonable empirical accuracy of the

[2]http://www.kaggle.com/c/pf2012-diabetes

differentially private AUCROC is obtained for $\epsilon > 0.25$.

For AP, we begin with the same setup as Figure 5.4, with error versus ϵ for several data set sizes with $n = m$. The general trends for AP in Figure 5.5 are similar to those for AUCROC, but with much higher error and slower decay of error as ϵ increases. This is due to the additional $\log n$ factor in the local sensitivity of AP. Figure 5.6 shows the distribution of outputted private AP values for selected n and m. For larger data sets like the top histogram, the outputs are nicely clustered around the true AP. However, when n and m are small, most of the outputs are truncated to 0 or 1.

5.7 Chapter Summary

Differentially private models allow organizations with sensitive data to provide guarantees about the effect of model release on the privacy of database entries. But for these models to be effectively evaluated, they must be run on new data, which may have similar privacy concerns. We presented methods for providing the same differential privacy guarantees for model evaluation, irrespective of the training setting. We provided high-utility mechanisms for AUCROC and AP and discussed the straightforward application of Laplace noise for accuracy and similar metrics. Future work includes creating mechanisms for other evaluation methods, such as private ROC and PR curves, and investigating the effect of performing cross-validation on a private database. We hope the discussion of differential privacy for model evaluation motivates future work to enable differential privacy to be applied more broadly throughout machine learning.

This chapter on private evaluation concludes the main contributions of this dissertation. In the final chapter we discuss future work in more detail and conclude with a summary.

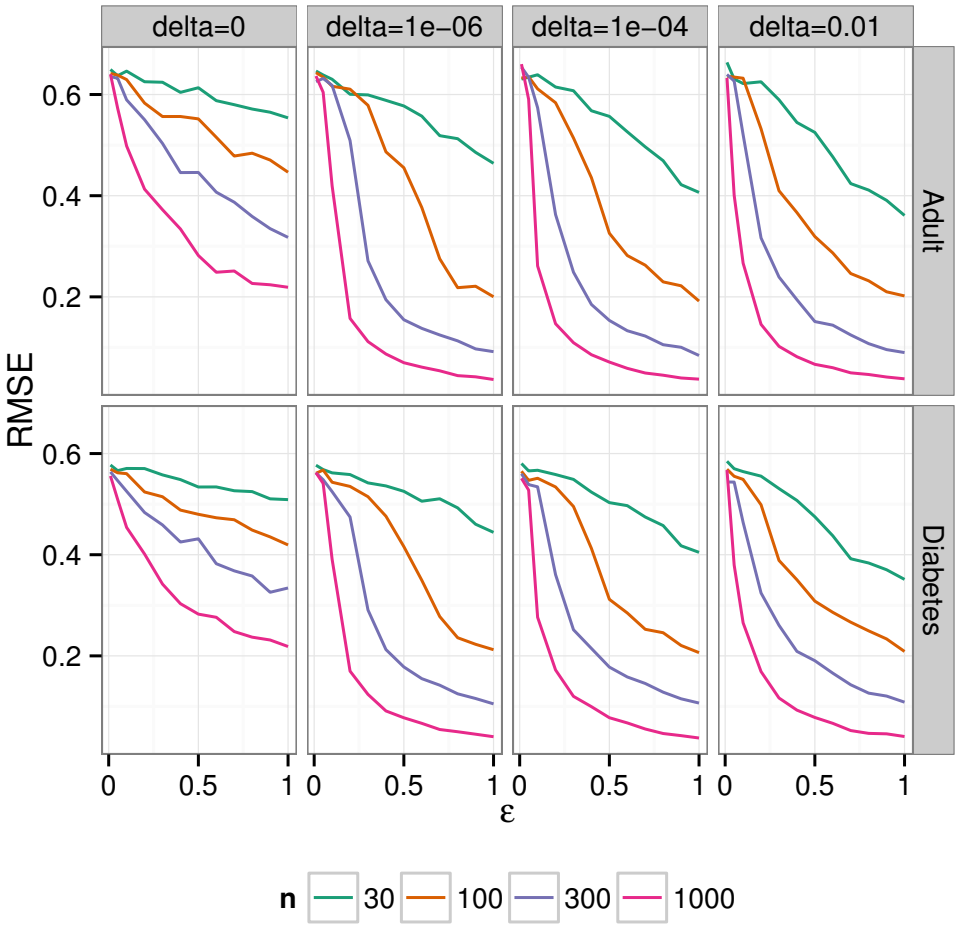


Figure 5.5: Root-mean-square error of (ϵ, δ)-differentially private AP versus ϵ for several different data set sizes $(n = m)$ using β-smooth sensitivity. The far left subplots use Cauchy noise, such that $\delta = 0$, and are ϵ-differentially private. The other subplots use Laplace noise.

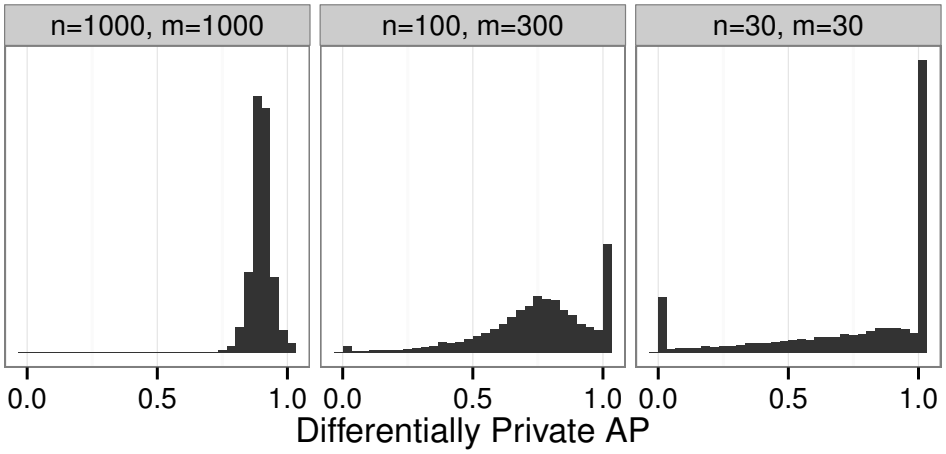


Figure 5.6: Histograms of (ϵ, δ)-differentially private AP output with varying data set sizes on the `adult` data set. $\epsilon = 1$ and $\delta = 0.01$.

6.1 Future Work

Before a final summary in Section 6.2, we briefly discuss several possibilities for future work building upon and inspired by this dissertation.

Aggregation with Different Skews

As mentioned in Section 3.4, the best way to aggregate PR curves from tasks with different skews is not known. The normalization used for AUCNPR suggests calculating the percentage of the achievable range of precision obtained at each recall by each of the curves. This can then be averaged across the tasks and translated back to a PR curve by choosing a representative skew. However, this leads to nonlinear transformations of PR space that can change the area under the curves in counterintuitive ways. An effective method for generating a summary PR curve that preserves measures of area in a satisfactory way and accounts for the unachievable region would be useful and is a promising area of future research.

AUCPR Estimators

While the *lower trapezoid*, *average precision*, and *interpolated median* estimators from Chapter 4 all converge to the correct answer in expectation as the test set size increases, they can have poor performance for small data sets. A high variance of the estimates is to be expected with few samples, but the fact that the recommended methods' estimates were substantially biased is troubling. Investigation of how these methods work on real outputs from trained classifiers and how well real outputs match the three scenarios would tell us how concerning this really is. Additionally, development of an improved estimator that is not as biased for small sample

sizes would be valuable. An improved estimator would be particularly useful for cross-validation since cross-validation's poor performance is mostly due to the biased estimators providing the wrong center for the interval.

AUCPR Confidence Intervals

In Chapter 4, we recommend using the *binomial* or *logit* methods to compute confidence intervals because they provide proper coverage. While the coverage is above the nominal $(1-\alpha)\%$, the intervals appear to be slightly too wide, as the coverage is consistently above $(1-\alpha)\%$ in Figure 4.7. The intervals being wider than necessary is also supported by Figure 4.8. This suggests that the sample size, n, used in calculating the standard error is not entirely correct. Perhaps the number of negative examples should partially contribute to an effective sample size that would produce tighter, but still valid, intervals. Thus, empirical and theoretical investigation of a more representative effective sample size for use in the parametric intervals is an intriguing area for future research.

Private Curves

Chapter 5 presented private methods for calculating AUCROC and AUCPR. However, the ROC and PR curves themselves are highly indicative of performance and provide a visual representation of the trade-offs at different operating points. Therefore, it would useful to have private methods for generating ROC and PR curves. Since confusion matrices are simply a collection of count queries, a simple approach is to add Laplace noise to each of the N confusion matrices from every decision threshold. Unfortunately, because the privacy budget must be split amongst all the confusion matrices, there is too much noise added at each point to be useful. This approach also does not improve with more data, as the privacy budget

must be spread across even more applications of Laplace noise with the larger test set.

Though adding noise to each point individually is not practical, it appears a private version of the curves might still be possible because curves from neighboring databases are quite similar, especially as N increases. Some potential approaches include choosing a small number of "important" points in the true curve to add noise to, restricting the outputted curves to a parametric family like binormal ROC curves, using the exponential mechanism to choose from some set of potential curves, or using the propose-test-release framework of Dwork and Lei (2009).

Cross-Validation and Privacy

Another important topic regarding privacy of test sets is how differential privacy applies to the commonly used evaluation technique of k-fold cross-validation. In cross-validation, a data set is divided into k folds and several iterations of training and testing are performed to obtain an estimate of future performance. Open questions regarding private applications of cross-validation include:

- When training and testing on the k folds, should the sequential composition of Theorem 2.5 be used? Or is the more generous parallel composition from McSherry (2009) applicable?
- How should the folds be chosen? Should the folds be randomized for every query or just once for each database?
- Can a non-private learning algorithm (in the form of code) be submitted and evaluated provided the only output is the result of a differentially private evaluation method?
- Are there additional query types or constructs that should be added to a private database system to facilitate cross-validation?

6.2 Summary

In this dissertation, we investigated novel properties of PR space, assessed methods for estimating AUCPR, and considered privacy preservation for test data in support of the thesis: *Not all methods of generating thresholdless metrics are created equal, and potential pitfalls and benefits accrue based on which methods are chosen.*

In Chapter 3, we demonstrated the existence of the unachievable region in PR space, proved theorems regarding its size and location, and discussed the implications for machine learning practitioners. In particular, great care must be taken when skew changes, because PR curves, AUCPR, AP, and F_β metrics intrinsically change with skew, regardless of the algorithm or model being evaluated. Next, an empirical evaluation of methods for estimating AUCPR and associated confidence intervals was performed in Chapter 4. We showed that different estimators do exhibit different behavior based on the score distributions and test set size. Therefore, choosing a good estimator and confidence interval method for any given problem is not simply a matter of blindly using a default choice. While we provide recommendations on good all-around methods, it is important to understand the properties of these methods and use those that best fit the prior knowledge of a particular task.

Switching to privacy, in Chapter 5 we raised the issue of test set privacy in addition to the privacy of training data. While a differentially private learning algorithm may provide a model that may be published with a small risk of disclosure, the same privacy concerns exist when evaluating that model. We provide algorithms to create private AUCROC and AP and prove their differential privacy, and we outline how to use standard differential privacy methods to privatize dichotomous model metrics.

In summary, evaluation methods are just as important as the learning algorithms we evaluate, and they should bear equal scrutiny and investigation. This dissertation investigated PR analysis and differentially

private evaluation to expand the knowledge of thresholdless evaluation and provided several avenues for future research in Section 6.1.

LIST OF NOTATION

D

a data set or database.

H_n

nth harmonic number.

$\text{Beta}(a, b)$

beta distribution parametrized by shape parameters a and b.

$\text{Binomial}(n, p)$

binomial distribution for n trials with probability p of success.

$\text{Uniform}(a, b)$

uniform distribution on the range $[a, b]$.

$\text{AUCPR}_{\text{MAX}}$

maximum AUCPR, $b - a$ when calculating AUCPR for recalls between a and b.

$\text{AUCPR}_{\text{MIN}}$

minimum AUCPR, equivalent to the size of the unachievable region.

fn

number of false negatives in test set.

fp

number of false positives in test set.

AP_{MIN}

minimum AP as required by the unachievable region.

θ

variable for AUCPR.

tn

number of true negatives in test set.

tp

number of true positives in test set.

c

decision threshold for labeling examples greater than c positive and the rest negative.

$d(D, D')$

number of rows that differ between D and D'.

$\mathbb{D}$

set of all data sets with N examples or databases with N rows.

$\mathbb{1}[A]$

indicator function for event A, 0 if A is false, 1 if A is true.

m

number of negative examples in test set.

N

total number of examples in test set.

n

number of positive examples in test set.

$\mathcal{N}(\mu, \sigma^2)$

: normal distribution with mean μ and variance σ^2.

p

: precision on test set.

π

: proportion of positive examples in test set, $\frac{n}{N}$.

r

: recall on test set.

X

: random variable for outputs on negative examples.

x_i

: score or model output on the ith negative test example.

Y

: random variable for outputs on positive examples.

y_j

: score or model output on the jth positive test example.

$y_{(i)}$

: ith order statistic of y, i.e., the ith largest value among the y_i's.

Z

: random variable for outputs (all examples).

LIST OF ACRONYMS

AP average precision.

AUCNPR normalized area under the PR curve.

AUCPR area under the PR curve.

AUCROC area under the ROC curve.

CI confidence interval.

FPR false positive rate.

IMDB internet movie database.

IR information retrieval.

LSM Learning using Structural Motifs.

MAP mean average precision.

PR precision-recall.

RMSE root-mean-square error.

ROC receiver operating characteristic.

SAYU Score-As-You-Use.

TPR true positive rate.

UW-CSE University of Washington Department of Computer Science and Engineering.

WLOG without loss of generality.

Abeel, Thomas, Yves Van de Peer, and Yvan Saeys. 2009. Toward a gold standard for promoter prediction evaluation. *Bioinformatics* 25(12):i313–i320.

Bache, K., and M. Lichman. 2013. UCI machine learning repository.

Bamber, Donald. 1975. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology* 12(4):387–415.

Boyd, Kendrick, Vítor Santos Costa, Jesse Davis, and David Page. 2012. Unachievable region in precision-recall space and its effect on empirical evaluation. In *Proceedings of the 29th international conference on machine learning*, ed. John Langford and Joelle Pineau, 639–646. ICML '12, New York, NY, USA: Omnipress.

Boyd, Kendrick, Kevin H. Eng, and C. David Page. 2013. Area under the precision-recall curve: Point estimates and confidence intervals. In *Machine learning and knowledge discovery in databases*, ed. Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, vol. 8190 of *Lecture Notes in Computer Science*, 451–466. Springer Berlin Heidelberg.

Brodersen, Kay Henning, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The binormal assumption on precision-recall curves. In *Pattern recognition (ICPR), 2010 20th international conference on*, 4263–4266. IEEE.

Bunescu, Razvan, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine* 33(2):139–155.

Carterette, Ben, and Ellen M. Voorhees. 2011. Overview of information retrieval evaluation. In *Current challenges in patent information retrieval*, ed. Mihai Lupu, Katja Mayer, John Tait, and Anthony J. Trippe, vol. 29 of *The Information Retrieval Series*, 69–85. Springer Berlin Heidelberg.

Chaudhuri, Kamalika, and Claire Monteleoni. 2008. Privacy-preserving logistic regression. In *Advances in neural information processing systems*, ed. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, vol. 8 of *NIPS 2008*, 289–296. Curran Associates, Inc.

Clémençon, Stéphan, and Nicolas Vayatis. 2009. Nonparametric estimation of the precision-recall curve. In *Proceedings of the 26th international conference on machine learning*, ed. Léon Bottou and Michael Littman, 185–192. ICML '09, Montreal: Omnipress.

Davis, Jesse, Elizabeth Burnside, Inês de Castro Dutra, David Page, and Vítor Santos Castro. 2005. An integrated approach to learning Bayesian networks of rules. In *Machine learning: ECML 2005*, ed. João Gama, Rui Camacho, Pavel Brazdil, Alípio Mário Jorge, and Luís Torgo, vol. 3720 of *Lecture Notes in Computer Science*, 84–95. Springer Berlin Heidelberg.

Davis, Jesse, and Mark Goadrich. 2006. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on machine learning*, 233–240. ICML '06, New York, NY, USA: ACM.

DeGroot, Morris H., and Mark J. Schervish. 2001. *Probability and statistics*. Addison-Wesley.

DeLong, Elizabeth R., David M. DeLong, and Daniel L. Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics* 44(3): 837–845.

Dietterich, Thomas G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10: 1895–1923.

Drummond, Chris, and Robert C. Holte. 2006. Cost curves: An improved method for visualizing classifier performance. *Machine Learning* 65(1): 95–130.

Dwork, Cynthia. 2006. Differential privacy. In *Automata, languages and programming*, 1–12. Springer.

Dwork, Cynthia, and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 45st annual ACM symposium on theory of computing*, 371–380. ACM.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, 265–285. Springer.

Efron, B. 1988. Bootstrap confidence intervals: Good or bad? *Psychological Bulletin* 104(2):293–296.

Efron, Bradley. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7(1):1–26.

Elkan, Charles. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, vol. 17, 973–978. Citeseer.

Fawcett, Tom. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27(8):861–874.

Fawcett, Tom, and Alexandru Niculescu-Mizil. 2007. PAV and the ROC convex hull. *Machine Learning* 68(1):97–106.

Ferri, Cèsar, José Hernández-Orallo, and Peter A Flach. 2011. Brier curves: a new cost-based visualisation of classifier performance. In *Proceedings of the 28th international conference on machine learning*, 585–592. ICML '11.

Flach, Peter A. 2003. The geometry of ROC space: understanding machine learning metrics through roc isometrics. In *Proceedings of the 20th international conference on machine learning*, 194–201. ICML '03.

Forman, George, and Martin Scholz. 2010. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter* 12(1):49–57.

Friedman, Arik, and Assaf Schuster. 2010. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, 493–502. ACM.

Ghosh, Arpita, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of ACM symposium on theory of computer science*. STOC 2009.

Giudici, Paolo. 2003. *Applied data mining*. John Wiley & Sons.

Goadrich, Mark, Louis Oliphant, and Jude Shavlik. 2006. Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves. *Machine Learning* 64(1-3):231–261.

Green, David Marvin, and John A. Swets. 1966. *Signal detection theory and psychophysics*. New York: John Wiley & Sons.

Guo, Bai-Ni, and Feng Qi. 2011. Sharp bounds for harmonic numbers. *Applied Mathematics and Computation* 218(3):991–995.

Hanley, J. A., and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143(1): 29–36.

Hernández-Orallo, José, Peter Flach, and César Ferri. 2013. ROC curves in cost space. *Machine Learning* 93(1):71–91.

Hu, Meiqun, Ee-Peng Lim, and Ramayya Krishnan. 2009. Predicting outcome for collaborative featured article nomination in wikipedia. In *Proceedings of the 3rd international AAAI conference on weblogs and social media*. ICWSM '09, AAAI Press.

Kifer, Daniel, and Ashwin Machanavajjhala. 2011. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD international conference on management of data*, 193–204. ACM.

Kifer, Daniel, and Ashwin Machanavajjhala. 2014. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions Database Systems (TODS)* 39(1):3:1–3:36.

Kok, Stanley, and Pedro Domingos. 2010. Learning Markov logic networks using structural motifs. In *Proceedings of the 27th international conference on machine learning*, 551–558. ICML '10.

Lehmann, Erich Leo, and George Casella. 1998. *Theory of point estimation*, vol. 31. Springer.

Ling, Xiao, and Daniel Weld. 2010. Temporal information extraction. In *AAAI conference on artificial intelligence*.

Liu, Yang, and Elizabeth Shriberg. 2007. Comparing evaluation metrics for sentence boundary detection. In *Acoustics, speech and signal processing, 2007. ICASSP 2007. IEEE international conference on*, vol. 4, IV–185. IEEE.

Lobo, Jorge M., Alberto JimÃ©nez-Valverde, and Raimundo Real. 2008. AUC: A misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography* 17(2):145–151.

Manning, Christopher D, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. New York, NY, USA: Cambridge University Press.

Manning, Christopher D, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. Cambridge, MA: MIT press.

Matthews, Gregory J., and Ofer Harel. 2013. An examination of data confidentiality and disclosure issues related to publication of empirical ROC curves. *Academic Radiology* 20(7):889 – 896.

McSherry, Frank, and Kunal Talwar. 2007. Mechanism design via differential privacy. In *Foundations of computer science, 2007. FOCS'07. 48th annual IEEE symposium on*, 94–103. IEEE.

McSherry, Frank D. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD international conference on management of data*, 19–30. ACM.

Metz, Charles E, and Helen B Kronman. 1980. Statistical significance tests for binormal ROC curves. *Journal of Mathematical Psychology* 22(3):218 – 243.

Mihalkova, Lilyana, and Raymond J. Mooney. 2007. Bottom-up learning of Markov logic network structure. In *Proceedings of the 24th international conference on machine learning*, ed. Zoubin Ghahramani, 625–632. ICML '07, Omnipress.

Natarajan, Sriraam, Tushar Khot, Kristian Kersting, Bernd Gutmann, and Jude Shavlik. 2012. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning* 86(1):25–56.

Nissim, Kobbi, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the*

39th annual ACM symposium on theory of computing, 75. STOC '07, New York, New York, USA: ACM Press.

Pepe, Margaret Sullivan. 2000. Receiver operating characteristic methodology. *Journal of the American Statistical Association* 95(449):308–311.

Pepe, Margaret Sullivan. 2004. *The statistical evaluation of medical tests for classification and prediction*. Oxford University Press, USA.

Piatetsky-Shapiro, Gregory, and Brij Masand. 1999. Estimating campaign benefits and modeling lift. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining*, 185–193. ACM.

Provost, Foster, and Tom Fawcett. 2001. Robust classification for imprecise environments. *Machine Learning* 42(3):203–231.

Provost, Foster J., Tom Fawcett, and Ron Kohavi. 1998. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th international conference on machine learning*, ed. Jude W. Shavlik, 445–453. ICML '98, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Provost, Foster J, Tom Fawcett, et al. 1997. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *In proceedings of the 3rd international conference on knowledge discovery and data mining*, 43–48. AAAI Press.

Qi, Feng, and Bai-Ni Guo. 2009. Sharp inequalities for the psi function and harmonic numbers. *arXiv preprint arXiv:0902.2524*.

Raghavan, Vijay, Peter Bollmann, and Gwang S Jung. 1989. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)* 7(3):205–229.

Richardson, Matthew, and Pedro Domingos. 2006. Markov logic networks. *Machine learning* 62(1-2):107–136.

Shao, Jun. 2003. *Mathematical statistics*. 2nd ed. Springer Verlag.

Sonnenburg, Sören, Alexander Zien, and Gunnar Rätsch. 2006. ARTS: accurate recognition of transcription starts in human. *Bioinformatics* 22(14): e472–e480.

Sutskever, Ilya, Ruslan Salakhutdinov, and Joshua Tenenbaum. 2009. Modelling relational data using Bayesian clustered tensor factorization. In *Advances in neural information processing systems*, 1821–1828. NIPS 2009.

Swets, John A., and Ronald M. Pickett. 1982. *Evaluation of diagnostic systems: Methods from signal detection theory*. New York: Academic Press.

Tang, Yuchun, Yan-Qing Zhang, Nitesh V. Chawla, and Sven Krasser. 2009. SVMs modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 39(1):281–288.

Wasserman, Larry. 2004. *All of statistics: A concise course in statistical inference*. Springer Verlag.

Yue, Yisong, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, 271–278. New York, NY, USA: ACM.

Zhang, Jun, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment* 5(11):1364–1375.