Statistical Learning along with Clustering

by

Xiaomin Zhang

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

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2021

Date of final oral examination: 11/10/2021

The dissertation is approved by the following members of the Final Oral Committee:
Po-Ling Loh, Lecturer, University of Cambridge
Yingyu Liang, Assistant Professor, Computer Sciences
Xiaojin Zhu, Professor, Computer Sciences
William Sethares, Professor, Electrical and Computer Engineering

© Copyright by Xiaomin Zhang 2021
All Rights Reserved

To my parents.

ACKNOWLEDGMENTS

First of all, I would like to deeply thank my advisor, Po-Ling Loh, for her patience, guidance and support. Research is a long journey. Hence, I appreciate that Po-Ling developed my independence and thinking on how to do research. I still remember the time in my first year working with her, at when my work draft was like a wall filled with only math formulations and I did not realize the importance of good writing skills. She explained me a lot and helped me to improve my writings. There were also lots of great memories with Po-Ling in Europe and I learnt from her on not only how to do research but also how to become a better human-being.

I would like to thank my committee members, Jerry Zhu, Yingyu Liang and William Sethares for their time and suggestions to my works. I would greatly thank Yingyu for his kind support and great insights. I got many inspirations from him in research and career life, which I believe that I will get benefits from it in the future as well. I would also greatly thank Jerry's critical questions in our meetings to encourage me to push our work to the best state. And I thank William for all his great questions in my defense.

Besides, I enjoyed the summer internships and would like to thank all the friends in San Diego and Seattle. Thank Xun Tang, Maolong Tang and Jisheng Liang to give me a few suggestions to start a new career.

I also hope to thank many academic professors and friends who kindly helped and supported me: Robert Nowak, Michael Ferris, Stephen Wright, Laurent Lessard, Nigel Boston, Mohit Gupta, Xucheng Zhang, Yifu Chen, Jinman Zhao.

There are many lovely friends in Madison who company me during the graduate life. We share joy and pain, success and failures, and we grow together. Big thanks to all of them!

Last but not least, I am endlessly grateful to my parents, Shichao Zhang and Lamei Ji. I am strong, when I am on your shoulders. You raise me up

to more than I can be.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

In this thesis, we build learning models as cluster the data points. Our motivation comes from the observations of learning models that the prediction performance stays in a bottleneck and cannot be improved via however we tune the model parameters. The phenomenon can be traced back to that the model does not fit to all data points. Thus, we hope each data point to get a better fit by introducing clustering along with learning. In particular, we consider the following two settings. One is with a large fraction of clean data points which satisfy the model specification and with a small portion outliers which do not satisfy the model specification. The outliers could be arbitrary, aiming to bias the estimation. The other is with two mixtures of data points, which satisfy two specified models respectively. Given the data diversity, our approach is to introduce some new variables to cluster the data points such that we do estimation and clustering at the same time.

In the first half of this thesis, we study a linear regression model with outliers/bugs in the training dataset that could adversely affect the estimation or prediction results. In the last decades, lots of researchers focus on building a robust model by changing the loss function. For example, instead of least squares, Huber loss became popular as a robust model. If the data point is clean, then Huber loss expects it to keep the square of the error. Otherwise, Huber loss takes the linear rate of the error. Such careful design could weaken the influence of the outliers so that a buggy point would not have infinite influence on the estimation. Our idea is to introduce each data point a new variable to determine whether it is buggy or not. In this new manner, we classify the points into clean ones and buggy ones and at the same time we do the estimation for the linear regression. If the data point is buggy, the new variable corresponding to the data point will absorb the bad influence into itself rather than push it to the estimator. Therefore, we can also decide whether a point is clean via the optimization results through the new variables. An interesting analysis shows that this new model is equivalent to using Huber loss, which suggests the robustness of our proposed model. While studying the model, we found a number of interesting problems to solve. We will give more details in Chapter 2.

In the second half of this thesis, we study the mixtures of ranking problem. In the past, it is popular to assume that there is one potential ranking over all items. When we explain a contradiction like votes of $i \succ j$ and $j \succ i$ ($\succ$ means the former item is preferred than the later one), we say it is caused by the noise and we use probabilistic model to absorb such noise. However, it is more likely in the real world that more potential rankings exist since people recognize things differently. Following modern views and ways to build a statistical model, we propose to create more clusters and estimate the rankings for each cluster. In practice, such a problem can be solved by EM algorithm, which alternatively clusters the data points and executes the estimation. In the study, we found that even a fundamental issue, identifiability, hasn't been solved so far. Therefore, our focus in this work would be giving a general framework for affirming the identifiability for mixtures of ranking models. More details will be discussed in Chapter 3.

2 PROVABLE TRAINING SET DEBUGGING FOR LINEAR REGRESSION

In this chapter, we investigate problems in penalized M-estimation, inspired by applications in machine learning debugging. Data are collected from two pools, one containing data with possibly contaminated labels, and the other which is known to contain only cleanly labeled points. We first formulate a general statistical algorithm for identifying buggy points and provide rigorous theoretical guarantees when the data follow a linear model. We then propose an algorithm for tuning parameter selection of our Lasso-based algorithm with theoretical guarantees. Finally, we consider a two-person "game" played between a bug generator and a debugger, where the debugger can augment the contaminated data set with cleanly labeled versions of points in the original data pool. We develop and analyze a debugging strategy in terms of a Mixed Integer Linear Programming (MILP). Finally, we provide empirical results to verify our theoretical results and the utility of the MILP strategy.

2.1 Introduction

Modern machine learning systems are extremely sensitive to training set contamination. Since sources of error and noise are unavoidable in real-world data (e.g., due to Mechanical Turkers, selection bias, or adversarial attacks), an urgent need has arisen to perform automatic debugging of large data sets. Cadamuro et al. (2016) and Zhang et al. (2018) proposed a method called "machine learning debugging" to identify training set errors by introducing new clean data. Consider the following real-world scenario: Company A collects movie ratings for users on a media platform, from which it learns relationships between features of movies and ratings in order to perform future recommendations. A competing company B knows

A's learning method and hires some users to provide malicious ratings. Company A could employ a robust method for learning contaminated data—but in the long run, it would be more effective for company A to *identify* the adversarial users and prevent them from submitting additional buggy ratings in the future. This distinguishes debugging from classical learning. The debugging problem also assumes that company A can hire an expert to help rate movies, from which it obtains a second trusted data set which is generally smaller than the original data set due to budget limitations. In this paper, we will study a theoretical framework for the machine learning debugging problem in a linear regression setting, where the main goal is to identify bugs in the data. We will also discuss theory and algorithms for selecting the trusted data set.

Our *first contribution* is to provide a rigorous theoretical framework explaining how to identify errors in the "buggy" data pool. Specifically, we embed a squared loss term applied to the trusted data pool into the extended Lasso algorithm proposed by Nguyen and Tran (2013), and reformulate the objective to better service the debugging task. Borrowing techniques from robust statistics (Huber and Ronchetti, 2011; She and Owen, 2011; Nguyen and Tran, 2013; Foygel and Mackey, 2014; Slawski and Ben-David, 2017) and leveraging results on support recovery analysis (Wainwright, 2009; Meinshausen and Yu, 2009), we provide sufficient conditions for successful debugging in linear regression. We emphasize that our setting, involving data coming from multiple pools, has not been studied in any of the earlier papers.

The work of Nguyen and Tran (2013) and Foygel and Mackey (2014) (and more recently, Sasai and Fujisawa (2020)) provided results for the extended Lasso with a theoretically optimal choice of tuning parameter, which depends on the unknown noise variance in the linear model. Our *second contribution* is to discuss a rigorous procedure for tuning parameter selection which does not require such an assumption. Specifically, our algorithm starts from a sufficiently large initial tuning parameter that produces the all-zeros vector as an estimator. Assuming the sufficient conditions for successful support recovery are met, this tuning parameter selection algorithm is guaranteed to terminate with a correct choice of tuning parameter after a logarithmic number of steps. Note that when outliers exist in the training data set, it is improper to use cross-validation to select the tuning parameter due to possible outliers in the validation data set.

Our *third contribution* considers how to design a second clean data pool, which is an important but previously unstudied problem in machine learning debugging. We consider a two-player “game" between a bug generator and debugger, where the bug generator performs adversarial attacks (Chakraborty et al., 2018), and the debugger applies Lasso-based linear regression to the augmented data set. On the theoretical side, we establish a sufficient condition under which the debugger can always beat the bug generator, and show how to translate this condition into a debugging strategy based on mixed integer linear programming. Our theory is only derived in the “noiseless” setting; nonetheless, empirical simulations show that our debugging strategy also performs well in the noisy setting. We experimentally compare our method to two other algorithms motivated by the machine learning literature, which involve designing two neural networks, one to correct labels and one to fit cleaned data (Veit et al., 2017); and a method based on semi-supervised learning that weights the noisy and clean datasets differently and employs a similarity matrix based on the graph Laplacian (Fergus et al., 2009).

The remainder of the paper is organized as follows: Section 2.2 introduces our novel framework for machine learning debugging using weighted M-estimators. Section 2.3 provides theoretical guarantees for recovery of buggy data points. Section 2.4 presents our algorithm for tuning parameter selection and corresponding theoretical guarantees. Section 2.5

discusses strategies for designing the second pool. Section 2.6 provides experimental results. Section 2.7 concludes the paper.

Notation: We write $\Lambda_{\min}(A)$ and $\Lambda_{\max}(A)$ to denote the minimum and maximum eigenvalues, respectively, of a matrix A. We use $\mathrm{Null}(A)$ to denote the nullspace of A. For subsets of row and column indices S and T, we write $A_{S,T}$ to denote the corresponding submatrix of A. We write $\|A\|_{\max}$ to denote the elementwise ℓ_∞-norm, $\|A\|_2$ to denote the spectral norm, and $\|A\|_\infty$ to denote the ℓ_∞-operator norm. For a vector $v \in \mathbb{R}^n$, we write $\mathrm{supp}(v) \subseteq \{1, \ldots, n\}$ to denote the support of v, and $\|v\|_\infty = \max |v_i|$ to denote the maximum absolute entry. We write $\|v\|_p$ to denote the ℓ_p-norm, for $p \geqslant 1$. We write $\mathrm{diag}(v)$ to denote the $n \times n$ diagonal matrix with entries equal to the components of v. For $S \subseteq \{1, \ldots, n\}$, we write v_S to denote the $|S|$-dimensional vector obtained by restricting v to S. We write $[n]$ as shorthand for $\{1, \ldots, n\}$.

2.2 Problem Formulation

We first formalize the data-generating models analyzed in this paper. Suppose we have observation pairs $\{(x_i, y_i)\}_{i=1}^n$ from the contaminated linear model

$$y_i = x_i^\top \beta^* + \gamma_i^* + \epsilon_i, \qquad 1 \leqslant i \leqslant n, \tag{2.1}$$

where $\beta^* \in \mathbb{R}^p$ is the unknown regression vector, $\gamma^* \in \mathbb{R}^n$ represents possible contamination in the labels, and the ϵ_i's are i.i.d. sub-Gaussian noise variables with variance parameter σ^2. We also assume the x_i's are i.i.d. and $x_i \perp\!\!\!\perp \epsilon_i$. This constitutes the "first pool." Note that the vector γ^* is unknown and may be generated by some adversary. If $\gamma_i^* = 0$, the i^{th} point is uncontaminated and follows the usual linear model; if $\gamma_i^* \neq 0$, the i^{th} point is contaminated/buggy. Let $T := \mathrm{supp}(\gamma^*)$ denote the indices of the buggy points, and let $t := |T|$ denote the number of bugs.

We also assume we have a clean data set which we call the "second pool." We observe $\{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^m$ satisfying

$$\widetilde{y}_i = \widetilde{x}_i^\top \beta^* + \widetilde{\epsilon}_i, \qquad 1 \leqslant i \leqslant m, \tag{2.2}$$

where the $\widetilde{\epsilon}_i$'s are i.i.d. sub-Gaussian noise variables with parameter $\widetilde{\sigma}^2$. Let $L := \frac{\sigma}{\widetilde{\sigma}}$, and suppose $L \geqslant 1$. Unlike the first pool, the data points in the second pool are all known to be uncontaminated.

For notational convenience, we also use $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$, and $\epsilon \in \mathbb{R}^m$ to denote the matrix/vectors containing the x_i's, y_i's, and ϵ_i's, respectively. Similarly, we define the matrices $\widetilde{X} \in \mathbb{R}^{m \times p}$, $\widetilde{y} \in \mathbb{R}^m$, and $\widetilde{\epsilon} \in \mathbb{R}^m$. Note that β^*, γ^*, T, t, and the noise parameters σ and $\widetilde{\sigma}$ are all assumed to be unknown to the debugger. In this paper, we will work in settings where $X^\top X$ is invertible.

Goal: Upon observing $\{(x_i, y_i)\}_{i=1}^n$, the debugger is allowed to design m points $\widetilde{X}$ in a stochastic or deterministic manner and query their corresponding labels $\widetilde{y}$, with the goal of recovering the support of γ^*. We have the following definitions:

Definition 2.1. *An estimator $\widehat{\gamma}$ satisfies **subset support recovery** if* $\text{supp}(\widehat{\gamma}) \subseteq \text{supp}(\gamma^*)$*. It satisfies **exact support recovery** if* $\text{supp}(\widehat{\gamma}) = \text{supp}(\gamma^*)$*.*

In words, when $\widehat{\gamma}$ satisfies subset support recovery, all estimated bugs are true bugs. When $\widehat{\gamma}$ satisfies exact support recovery, the debugger correctly flags *all* bugs. We are primarily interested in exact support recovery.

Weighted M-estimation Algorithm: We propose to optimize the joint objective

$$(\widehat{\beta}, \widehat{\gamma}) \in \arg \min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^n} \left\{ \frac{1}{2n} \|y - X\beta - \gamma\|_2^2 + \frac{\eta}{2m} \|\widetilde{y} - \widetilde{X}\beta\|_2^2 + \lambda \|\gamma\|_1 \right\}, \tag{2.3}$$

where the weight parameter $\eta > 0$ determines the relative importance of the two data pools. The objective function applies the usual squared loss to the points in the second pool and introduces the additional variable γ to help identify bugs in the first pool. Furthermore, the ℓ_1-penalty encourages $\widehat{\gamma}$ to be sparse, since we are working in settings where the number of outliers is relatively small compared to the total number of data points. Note that the objective function (2.3) may equivalently be formulated as a weighted sum of M-estimators applied to the first and second pools, where the loss for the first pool is the robust Huber loss and the loss for the second pool is the squared loss (cf. Proposition A.2 in Appendix A.1.1).

Lasso Reformulation: Recall that our main goal is to estimate (the support of) γ^* rather than β^*. Thus, we will restrict our attention to γ^* by reformulating the objectives appropriately. We first introduce some notation: Define the stacked vectors/matrices

$$X' = \begin{pmatrix} X \\ \sqrt{\frac{\eta n}{m}}\widetilde{X} \end{pmatrix}, y' = \begin{pmatrix} y \\ \sqrt{\frac{\eta n}{m}}\widetilde{y} \end{pmatrix}, \epsilon' = \begin{pmatrix} \epsilon \\ \sqrt{\frac{\eta n}{m}}\tilde{\epsilon} \end{pmatrix}, \qquad (2.4)$$

where $X' \in \mathbb{R}^{(m+n)\times p}$ and $y', \epsilon' \in \mathbb{R}^{m+n}$. For a matrix A, let $P_A = A(A^\top A)^{-1}A^\top$ and $P_A^\perp = I - A(A^\top A)^{-1}A^\top$ denote projection matrices onto the range of the column space of A and its orthogonal complement, respectively. For a matrix $S \subseteq [n]$, let M_S denote the $(n+m)\times|S|$ matrix with i^{th} column equal to the canonical vector $e_{S(i)}$. Thus, right-multiplying by M_S truncates a matrix to only include columns indexed by S. We have the following useful result:

Proposition 2.2. *The objective function*

$$\widehat{\gamma} \in \arg\min_{\gamma\in\mathbb{R}^n} \left\{ \frac{1}{2n} \|P_{X'}^\perp y' - P_{X'}^\perp M_{[n]}\gamma\|_2^2 + \lambda\|\gamma\|_1 \right\} \qquad (2.5)$$

shares the same solution for $\widehat{\gamma}$ with the objective function (2.3).

Proposition 2.2, proved in Appendix A.1.2, translates the joint optimization problem (2.3) into an optimization problem only involving the parameter of interest γ. We provide a discussion regarding the corresponding solution $\widehat{\beta}$ in Appendix A.1.1 for the interested reader. Note that the optimization problem (2.5) corresponds to linear regression of the vector/matrix pairs $(P_{X'}^{\perp}y', P_{X'}^{\perp}M_{[n]})$ with a Lasso penalty, inspiring us to borrow techniques from high-dimensional statistics.

2.3 Support Recovery

The reformulation (2.5) allows us to analyze the machine learning debugging framework through the lens of Lasso support recovery. The three key conditions we impose to ensure support recovery are provided below. Recall that we use M_T to represent the truncation matrix indexed by T.

Assumption 2.3 (Minimum Eigenvalue). *Assume that there is a positive number* $b'_{\min}$ *such that*

$$\Lambda_{\min}\left(M_T^\top P_{X'}^{\perp} M_T\right) \geqslant b'_{\min}. \tag{2.6}$$

Assumption 2.4 (Mutual Incoherence). *Assume that there is a number* $\alpha' \in [0, 1)$ *such that*

$$\|M_{T^c}^\top P_{X'}^{\perp} M_T (M_T^\top P_{X'}^{\perp} M_T)^{-1}\|_\infty \leqslant \alpha'. \tag{2.7}$$

Assumption 2.5 (Gamma-Min). *Assume that*

$$\min_{i \in T} |\gamma_i^*| > G' := \|(M_T^\top P_{X'}^{\perp} M_T)^{-1} M_T^\top P_{X'}^{\perp} \epsilon'\|_\infty + n\lambda \left\|(M_T^\top P_{X'}^{\perp} M_T)^{-1}\right\|_\infty. \tag{2.8}$$

Assumption 2.3 comes from a primal-dual witness argument Wainwright (2009) to guarantee that the minimizer $\widehat{\gamma}$ is unique. Assumption 2.4 measures a relationship between the sets T^c and T, indicating that the large number of nonbuggy covariates (i.e., T^c) cannot exert an overly strong effect on the subset of buggy covariates Ravikumar et al. (2010). To aid intuition, consider an orthogonal design, where $X = \begin{bmatrix} cI_{[t],[p]} \\ c'I_{p\times p} \end{bmatrix}$ and $\widetilde{X} = c''I_{p\times p}$, for some $t < p$, and $c, c', c'' > 0$. We use the notation $I_{[t],[p]}$ to denote a submatrix of $I_{p\times p}$ with rows indexed by the set $[t]$. Suppose the first t points are bugs, and for simplicity, let $\eta = m/n$. Then the mutual incoherence condition requires $c < c' + \frac{(c'')^2}{c'}$, meaning that in every direction e_i, the component of buggy data cannot be too large compared to the nonbuggy data and the clean data. Assumption 2.5 lower-bounds the minimum absolute value of elements of γ. Note that λ is chosen based on ϵ', so the right-hand expression is a function of ϵ'. This assumption indeed captures the intuition that the signal-to-noise ratio, $\frac{\min_{i\in T}|\gamma_i^*|}{\sigma}$, needs to be sufficiently large.

We now provide two general theorems regarding subset support recovery and exact support recovery.

Theorem 2.6 (Subset support recovery)**.** *Suppose* $P_{X'}^{\perp}$ *satisfies Assumptions 2.3 and 2.4. If the tuning parameter satisfies*

$$\lambda \geqslant \frac{2}{1-\alpha'}\left\|M_{T^c}P_{X'}^{\perp}\left(I - P_{X'}^{\perp}M_T(M_T^\top P_{X'}^{\perp}M_T)^{-1}M_T^\top P_{X'}^{\perp}\right)\frac{\epsilon'}{n}\right\|_\infty, \tag{2.9}$$

then the objective (2.5) *has a unique optimal solution* $\widehat{\gamma}$*, satisfying* $\mathrm{supp}(\widehat{\gamma}) \subseteq \mathrm{supp}(\gamma^*)$ *and* $\|\widehat{\gamma} - \gamma^*\|_\infty \leqslant G'$.

Theorem 2.7 (Exact support recovery)**.** *In addition to the assumptions in Theorem 2.6, suppose Assumption 2.5 holds. Then we have a unique optimal solution* $\widehat{\gamma}$*, which satisfies exact support recovery.*

Note that we additionally need Assumption 2.5 to guarantee exact support recovery. This follows the aforementioned intuition regarding the assumption. In particular, recall that ϵ and $\widetilde{\epsilon}$ are sub-Gaussian vectors with parameters σ^2 and σ^2/L, respectively, where $L \geqslant 1$ (i.e., the clean data pool has smaller noise). The minimum signal strength $\min_{i \in T} |\gamma_i^*|$ needs to be at least $\Theta(\sigma\sqrt{\log n})$, since $\mathbb{E}\left[\max_{i\in[n]} |\epsilon_i|\right] \leqslant \sigma\sqrt{2\log(2n)}$. Intuitively, if $\min_{i\in T} |\gamma_i^*|$ is of constant order, it is difficult for the debugger to distinguish between random noise and intentional contamination.

We now present two special cases to illustrate the theoretical benefits of including a second data pool. Although Theorems 2.6 and 2.7 are stated in terms of *deterministic* design matrices and error vectors ϵ and $\widetilde{\epsilon}$, the assumptions can be shown to hold with high probability in the example. We provide formal statements of the associated results in Appendix A.1.3 and Appendix A.1.3.

Example 2.8 (Orthogonal design)**.** *Suppose* Q *is an orthogonal matrix with columns* $q_1, q_2, \dots, q_p$*, and consider the setting where* $X_T = RQ^\top \in \mathbb{R}^{t\times p}$ *and* $X_{T^c} = FQ^\top \in \mathbb{R}^{p\times p}$*, where* $R = \left[\operatorname{diag}(\{r_i\}_{i=1}^t) \mid \mathbf{0}_{t\times(p-t)}\right]$ *and* $F = \operatorname{diag}(\{f_i\}_{i=1}^p)$*. Thus, points in the contaminated first pool correspond to orthogonal vectors. Similarly, suppose the second pool consists of* (*rescaled*) *columns of* Q*, so* $\widetilde{X} = WQ^\top \in \mathbb{R}^{m\times p}$*, where* $W = \operatorname{diag}(\{w_i\}_{i=1}^p)$*.* (*To visualize this setting, one can consider* $Q = I$ *as a special case.*) *The mutual incoherence parameter is* $\alpha' = \max_{1\leqslant i\leqslant t}\left|\frac{r_i f_i}{f_i^2+\eta\frac{n}{m}w_i^2}\right|$*. Hence,* $\alpha' < 1$ *if the weight of a contaminated point dominates the weight of a clean point in any direction, e.g., when* $|r_i| > |f_i|$ *and* $w_i = 0$*; in contrast, if the second pool includes clean points* $w_i q_i$ *with sufficiently large* $|w_i|$*, we can guarantee that* $\alpha' < 1$*. Furthermore,*

$$G' \approx \sigma\left(\sqrt{2\log t} + c\right)\sqrt{1 + \max_{1\leqslant i\leqslant t}\frac{r_i^2(Lf_i^2 + \frac{\eta n}{m}w_i^2)}{L(f_i^2 + \frac{\eta n}{m}w_i^2)^2}} + \frac{2\sigma}{1-\alpha'}\left(\sqrt{\log 2(n-t)} + C\right)\left(1 + \max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2 + \frac{\eta n}{m}w_i^2}\right)$$

for some constant C*. It is not hard to verify that* G' *decreases by adding a second pool. Further note that the behavior of the non-buggy subspace,* $\operatorname{span}\{q_{t+1}, \dots, q_p\}$*,* *is not involved in any conditions or conclusions. Thus, our key observation is that the theoretical results for support recovery consistency only rely on the addition of second-pool points in buggy directions.*

Example 2.9 (Random design). *Consider a random design setting where the rows of* X *and* $\widetilde{X}$ *are drawn from a common sub-Gaussian distribution with covariance* Σ*. The conditions in Assumptions 2.3–2.5 are relaxed in the presence of a second data pool when* n *and* m *are large compared to* p*: First,* $b'_{\min}$ *increases by adding a second pool. Second,* $\alpha' \approx \frac{\|X_{T^c}\Sigma^{-1}X_T\|_\infty}{n-t+\eta n}$*, so the mutual incoherence parameter also decreases by adding a second pool. Third,*

$$G' \approx \frac{2\sigma\sqrt{\log t}}{b'_{\min}} + \frac{2\sigma}{1-\alpha'} \max\left\{1, \sqrt{\frac{\eta n}{mL}}\right\} \left\|(I_{t\times t} - \frac{X_T\Sigma^{-1}X_T^\top}{n+\eta n})^{-1}\right\|_\infty,$$

where X_T *and* X_{T^c} *represent the submatrices of* X *with rows indexed by* T *and* T^c*, respectively. Note that the one-pool case corresponds to* $\eta = 0$ *and*

$$\left\|(I_{t\times t} - \frac{X_T\Sigma^{-1}X_T^\top}{n+\eta n})^{-1}\right\|_\infty < \left\|(I_{t\times t} - \frac{X_T\Sigma^{-1}X_T^\top}{n})^{-1}\right\|_\infty,$$

so if we choose $\eta \leqslant \frac{mL}{n}$*, then* G' *decreases by adding a second pool. Therefore, all three assumptions are relaxed by having a second pool, making it easier to achieve exact support recovery.*

We also briefly discuss the three assumptions with respect to the weight parameter η: Increasing η always relaxes the eigenvalue and mutual incoherence conditions, so placing more weight on the second pool generally helps with subset support recovery. However, the same trend does not necessarily hold for exact recovery. This is because a larger value of η causes the lower bound (2.9) on λ to increase, resulting in a stricter gamma-min condition. Therefore, there is a tradeoff for selecting η.

2.4 Tuning Parameter Selection

A drawback of the results in the previous section is that the proper choice of tuning parameter depends on a lower bound (2.9) which cannot be calculated without knowledge of the unknown parameters (T, α', ϵ'). The tuning parameter λ determines how many outliers a debugger detects; if λ is large, then $\widehat{\gamma}$ contains more zeros and the algorithm detects fewer bugs. A natural question arises: *In settings where the conditions for exact support recovery hold, can we select a data-dependent tuning parameter that correctly identifies all bugs?* In this section, we propose an algorithm which answers this question in the affirmative.

2.4.1 Algorithm and Theoretical Guarantees

Our tuning parameter selection algorithm is summarized in Algorithm 1, which searches through a range of parameter values for λ, starting from a large value λ_u and then halving the parameter on each successive step until a stopping criterion is met. The intuition is as follows: First, let λ^* be the right-hand expression of inequality (2.9). Suppose that for any value in $I = [\lambda^*, 2\lambda^*]$, support recovery holds. Then given $\lambda_u > \lambda^*$, the geometric series $\Lambda = \left\{\lambda_u, \frac{\lambda_u}{2}, \frac{\lambda_u}{4}, \dots\right\}$ must contain at least one correct parameter for exact support recovery since $\Lambda \cap I \neq \emptyset$, guaranteeing that the algorithm stops. As for the stopping criterion, let X_S denote the submatrix of X with rows indexed by S for $T^c \subseteq S \subseteq [n]$. We have $P^{\perp}_{X_S} \overset{|S|\to\infty}{\longrightarrow} \left(1 - \frac{p}{|S|)}\right) I$ under some mild assumptions on X, in which case $P^{\perp}_{X_S} y_S \to \left(1 - \frac{p}{|S|}\right)(\gamma^*_S + \epsilon_S)$. When λ is large and the conditions hold for subset support recovery but not exact recovery, we have $S \cap T \neq \emptyset$, so

$$\min |P^{\perp}_{X_S} y_S| \geqslant \left(1 - \frac{p}{|S|}\right)\left(\min |\gamma^*_T| - \max_{i \in [n]} |\epsilon_i|\right).$$

In contrast, when $S = T^c$, we have

$$\min |P^{\perp}_{X_S} y_S| \leqslant \left(1 - \frac{p}{|S|}\right) \max_{i \in [n]} |\epsilon_i|.$$

When $\min |\gamma^*_T|$ is large enough, the task then reduces to choosing a proper threshold to distinguish the error $|\epsilon_{T^c}|$ from the bug signal $|\gamma^*_T|$, which occurs when the threshold is chosen between $\max_i |\epsilon_i|$ and $\min_{i \in T} |\gamma^*_i| - \max_i |\epsilon_i|$.

Algorithm 1 Regularizer selection

Input: $\lambda_u, \bar{c}$
Output: $\hat{\lambda}^k$

$C = 1, k = 1, \hat{\lambda}^k = \lambda_u$.
while $C = 1$ **do**
  $\hat{\gamma}^k \in \arg\min_{\gamma \in \mathbb{R}^n} \left\{ \frac{1}{2n} \|P^{\perp}_{X'} y' - P^{\perp}_{X'} M_{[n]} \gamma\|_2^2 + \hat{\lambda}^k \|\gamma\|_1 \right\}$.
  Let $X^{(k)}, y^{(k)}$ consist of x_i, y_i such that $i \notin \operatorname{supp}(\hat{\gamma}^k)$. Let $l^{(k)}$ be the length of $y^{(k)}$.
  $\hat{\sigma} = \frac{l^{(k)}}{l^{(k)} - p} \cdot \operatorname{median}\left(\left|P^{\perp}_{X^{(k)}} y^{(k)}\right|\right)$.
  $C = 0$ if $\|P^{\perp}_{X^{(k)}} y^{(k)}\|_\infty \leqslant \frac{5}{2} \bar{c}^{-1} \sqrt{\log 2n}\, \hat{\sigma}$.
  $k = k + 1, \hat{\lambda}^k = \hat{\lambda}^{k-1}/2$.
end while

With the above intuition, we now state our main result concerning exact recovery guarantees for our algorithm. Recall that the ϵ_i's are sub-Gaussian with parameter σ^2.

Let $c_t := \frac{t}{n} < \frac{1}{2}$ denote the fraction of outliers. We assume knowledge of a constant $\bar{c}$ that satisfies $c_t + \mathbb{P}[|\epsilon_i| \leqslant \bar{c}\sigma] < \frac{1}{2}$. Note that a priori knowledge of $\bar{c}$ is a less stringent assumption than knowing σ, since we can always choose $\bar{c}$ to be close to zero. For instance, if we know the ϵ_i's are Gaussian, we can choose $\bar{c} < \operatorname{erf}^{-1}(\frac{1}{2} - c_t)$; in practice, we can usually estimate c_t to be less than $\frac{1}{3}$, so we can take $\bar{c} = \operatorname{erf}^{-1}(\frac{1}{6})$. As shown later, the tradeoff is that having a larger value of $\bar{c}$ provides the desired guarantees under weaker requirements on the lower bound of $\min_{i \in T} |\gamma^*_i|$.

Hence, if we know more about the shape of the error distribution, we can be guaranteed to detect bugs of smaller magnitudes. We will make the following assumption on the design matrix:

Assumption 2.10. *There exists a* $p \times p$ *positive definite matrix* Σ*, with bounded minimum and maximum eigenvalues, such that for all* $X^{(k)}$ *appearing in the while loop of Algorithm 1, we have*

$$\left\| \frac{X^{(k)}\Sigma^{-1}X^{(k)\top}}{p} - I \right\|_{\max} \leqslant c \max\left\{ \sqrt{\frac{\log l^{(k)}}{p}}, \frac{\log l^{(k)}}{p} \right\},$$
$$\left\| \frac{X^{(k)\top}X^{(k)}}{l^{(k)}} - \Sigma \right\|_2 \leqslant \frac{\lambda_{\min}(\Sigma)}{2}, \tag{2.10}$$

where $l^{(k)}$ *is the number of rows of the matrix* $X^{(k)}$ *and* c *is a universal constant.*

This assumption is a type of concentration result, which we will show holds w.h.p. in some random design settings in the following proposition:

Proposition 2.11. *Suppose the* x_i*'s are i.i.d. and satisfy any of the following additional conditions:*

(*a*) *the* x_i*'s are Gaussian and the spectral norm of the covariance matrix is bounded;*

(*b*) *the* x_i*'s are sub-Gaussian with mean zero and independent coordinates, and the spectral norm of the covariance matrix is bounded; or*

(*c*) *the* x_i*'s satisfy the convex concentration property.*

Then Assumption 2.10 holds with probability at least $1 - O(n^{-1})$*.*

The Σ matrix can be chosen as the covariance of X. In fact, Assumption 2.10 shows that $P^{\perp}_{X^{(k)}}$ is approximately a scalar matrix. We now introduce some additional notation: For $\nu > 0$, define c_ν and C_ν such that $\nu = \mathbb{P}[|\epsilon_i| \leqslant c_\nu \sigma]$ and $\nu = \mathbb{P}[|\epsilon_i| \geqslant C_\nu \sigma]$. We write $G'(\lambda)$ to denote the function of λ in the right-hand expression of inequality (2.8). Proofs of the theoretical results in this section are provided in Appendix A.1.4.

Theorem 2.12. *Assume ν is a constant satisfying $\nu + c_t < \frac{1}{2}$. Suppose Assumption 2.10, the minimum eigenvalue condition, and the mutual incoherence condition hold. If*

$$n \geqslant \max\left\{\left[\frac{24}{c_\nu}\right]^{\frac{1}{c_n}}, \left[\frac{C\log 2n}{1-c_t}(p^2+\log^2 n)\right]^{\frac{1}{1-2c_n}}\right\}, \tag{2.11}$$

where C is an absolute constant, and

$$\begin{aligned} \min_{i\in T}|\gamma_i^*| &> \max\left\{G'(2\lambda^*), 4\sqrt{\log(2n)}\sigma, \frac{5}{4}\sqrt{\log(2n)}\frac{c_\nu+5C_\nu}{\bar{c}}\sigma\right\}, \\ \|\gamma^*\|_\infty &\leqslant \frac{\sqrt{C}c_\nu}{16\sqrt{2}}\sqrt{1-c_t}\sqrt{\log 2n}\,\frac{n^{1/2+c_n}}{t}\sigma, \end{aligned} \tag{2.12}$$

for some $c_n \in (0, \frac{1}{2})$, then Algorithm 1 with inputs $\bar{c} < c_\nu$ and $\lambda_u \geqslant \lambda^*$ will return a feasible $\widehat{\lambda}$ in at most $\log_2\left(\frac{\lambda_u}{\lambda^*}\right)$ iterations such that the Lasso estimator $\widehat{\gamma}$ based on $\widehat{\lambda}$ satisfies $\operatorname{supp}(\widehat{\gamma}) = \operatorname{supp}(\gamma^*)$, with probability at least

$$1 - \frac{3\log_2\left(\frac{\lambda_u}{\lambda^*}\right)}{n-t} - 2\log_2\left(\frac{\lambda_u}{\lambda^*}\right)\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right).$$

Theorem 2.12 guarantees exact support recovery for the output of Algorithm 1 without knowing σ. Note that compared to the gamma-min condition (2.8) with $\lambda = \lambda^*$, the required lower bound (2.12) only differs by a constant factor. In fact, the constant 2 inside $G'(2\lambda^*)$ can be replaced by any constant $c > 1$, but Algorithm 1 will then update $\hat{\lambda}^k = \hat{\lambda}^{k-1}/c$ and require $\log_c\left(\frac{\lambda_u}{\lambda^*}\right)$ iterations. Further note that larger values of c_t translate into a larger sample size requirement, as $n = \Omega\left(\frac{1}{1-c_t}\right)$ for c_n being close to 0. A limitation of the theorem is the upper bound on $\|\gamma^*\|_\infty$, where t needs to be smaller than n in a nonlinear relationship. Also, n is required to be $\Omega(p^2)$. These two conditions are imposed in our analysis in order to guarantee that $P^{\perp}_{X_S} y_S \to \left(1 - \frac{p}{|S|}\right)(\gamma^*_S + \epsilon_S)$. We now present a result indicating a practical choice of λ_u:

Corollary 2.13. *Define*

$$\lambda(\sigma) := \frac{8 \max\{1, \sqrt{\frac{\eta n}{Lm}}\}}{1-\alpha'} \sqrt{\log 2(n-t)} \frac{\|P^{\perp}_{X,T^c}\|_2}{n} \cdot c\sigma.$$

Suppose Assumption 2.10, the minimum eigenvalue condition, and the mutual incoherence condition hold. Also assume conditions (2.11) *and* (2.12) *hold when replacing* λ^* *by* $\lambda(\sigma)$*. Taking the input* $\lambda_u = \frac{2\|M_{[n]} P^{\perp}_{X'} y'\|_\infty}{n}$*, Algorithm 1 outputs a parameter* $\widehat{\lambda}$ *in* $O(\log n)$ *iterations which provides exact support recovery, with probability at least*

$$1 - \frac{4\left(c' \log_2 n + \max\left\{0, \frac{1}{2}\log_2 \frac{\eta n}{mL}\right\}\right)}{n-t} - 2\left(c' \log_2 n + \frac{1}{2}\max\left\{0, \log_2 \frac{\eta n}{mL}\right\}\right) e^{-2\left(\frac{1}{2} - c_t - \nu\right)^2 n}.$$

Note that λ_u can be calculated using the observed data set. Further note that the algorithm is guaranteed to stop after $O(\log n)$ iterations, meaning it is sufficient to test a relatively small number of candidate parameters in order to achieve exact recovery.

2.5 Strategy for Second Pool Design

We now turn to the problem of designing a clean data pool. In the preceding sections, we have discussed how a second data pool can aid exact recovery under sub-Gaussian designs. In practice, however, it is often unreasonable to assume that new points can be drawn from an entirely different distribution. Specifically, recall the movie rating example discussed in Section 2.1: The expert can only rate movies in the movie pool, say $\{x_i\}_{i=1}^n$, whereas an arbitrarily designed $\widetilde{x}$, e.g., $\widetilde{x} = x_1/2$, is unlikely to correspond to an existing movie. Thus, we will focus on devising a debugging strategy where the debugger is allowed to choose points for the second pool which have the same covariates as points in the first pool.

In particular, we consider this problem in the "worst" case: suppose a bug generator can generate any $\gamma^* \in \Gamma := \{\gamma \in \mathbb{R}^n : \text{supp}(\gamma)| \leqslant t\}$ and add it to the correct labels $X\beta^*$. We will also suppose the bug generator knows the debugger's strategy. The debugger attempts to add a second data pool which will ensure that all bugs are detected regardless of the choice of γ^*. Our theory is limited to the noiseless case, where $y = X\beta^* + \gamma^*$ and $\widetilde{y} = \widetilde{X}\beta^*$; the noisy case is studied empirically in Section 2.6.3.

2.5.1 Preliminary Analysis

We denote the debugger's choice by $\widetilde{x}_i = X^\top e_{\nu(i)}$, for $i \in [m]$, where $e_{\nu(i)} \in \mathbb{R}^n$ is a canonical vector and $\nu : [m] \to [n]$ is injective. In matrix form, we write $\widetilde{X} = X_D$, where $D \subseteq [n]$ represents the indices selected by the debugger. Assume $m < p$, so the debugger cannot simply use the clean pool to obtain a good estimate of β. In the noiseless case, we can write the debugging algorithm as follows:

$$\begin{aligned} &\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^n} \|\gamma\|_1 \\ &\text{subject to } y = X\beta + \gamma,\ \widetilde{y} = \widetilde{X}\beta. \end{aligned} \tag{2.13}$$

Similar to Proposition 2.2, given a γ, we can pick β to satisfy the constraints, specifically $\beta = \left(X^\top X + \widetilde{X}^\top \widetilde{X}\right)^{-1} \left(X^\top (y - \gamma) + \widetilde{X}^\top \widetilde{y}\right)$. Eliminating β, we obtain the optimization problem

$$\begin{aligned}&\min_{\gamma\in\mathbb{R}^n} \|\gamma\|_1\\ &\text{subject to } \begin{bmatrix} y \\ \widetilde{y} \end{bmatrix} = \begin{bmatrix} X \\ \widetilde{X} \end{bmatrix} \left(X^\top X + \widetilde{X}^\top \widetilde{X}\right)^{-1} \left(X^\top (y-\gamma) + \widetilde{X}^\top \widetilde{y}\right) + \begin{bmatrix} \gamma \\ \vec{0} \end{bmatrix}.\end{aligned} \tag{2.14}$$

Before presenting our results for support recovery, we introduce some definitions. Define the cone set $\mathbb{C}(K)$ for some subset $K \subseteq [n]$ and $|K| = t$:

$$\mathbb{C}(K) := \{\Delta \in \mathbb{R}^n : \|\Delta_{K^c}\|_1 \leqslant \|\Delta_K\|_1\}. \tag{2.15}$$

Further let $\mathbb{C}^A = \cup_{K\subseteq[n],|K|=t}\mathbb{C}(K)$, and define

$$\bar{P}(D) = \begin{bmatrix} I - X\left(X^\top X + X_D^\top X_D\right)^{-1} X^\top \\ X_D\left(X^\top X + X_D^\top X_D\right)^{-1} X^\top \end{bmatrix}.$$

Theorem 2.14. *Suppose*

$$\mathrm{Null}(\bar{P}(D)) \cap \mathbb{C}^A = \{\vec{0}\}. \tag{2.16}$$

Then a debugger who queries the points indexed by D *cannot be beaten by any bug generator who introduces at most* t *bugs.*

Theorem 2.14 suggests that equation (2.16) is a sufficient condition for support recovery for an omnipotent bug generator who knows the subset D. As a debugger, the consequent goal is to find such a subset D which makes equation (2.16) true. Whether such a D exists and how to find it will be discussed in Section 2.5.2.

Remark 2.15. *When* $m = n$*, we can verify that* $\mathrm{Null}(\bar{P}(D)) = \{\vec{0}\}$*, which implies that equation* (2.16) *always holds. Indeed, in this case, we can simply take* $\widetilde{X} = X$ *and solve for* β^* *explicitly to recover* γ^**.*

Remark 2.16. *As stated in Theorem 2.14, equation* (2.16) *is a sufficient condition for support recovery. In fact, it is an if-and-only-if condition for signed support recovery: When equation* (2.16) *holds,* $\text{sign}(\widehat{\gamma}) = \text{sign}(\gamma^*)$*; and when it does not hold, the bug generator can find a* γ^* *with* $\text{supp}(\gamma^*) \leqslant t$ *such that* $\text{sign}(\widehat{\gamma}) \neq \text{sign}(\gamma^*)$.

Remark 2.17. *We can also write* $\text{Null}(\bar{P}(D))$ *as*

$$\{u \in \mathbb{R}^n \mid \exists v \in \mathbb{R}^p, \textit{ s.t. } u = Xv, X_D v = 0\}.$$

Let $\widehat{\beta} = \beta^* + \nu$ *for some vector* $\nu \in \mathbb{R}^p$. *From the constraint-based algorithm, we obtain*

$$\begin{aligned} y_T &= X_T(\beta^* + \nu) + \widehat{\gamma}_T, \\ y_{T^c} &= X_{T^c}(\beta^* + \nu) + \widehat{\gamma}_{T^c}, \\ y_D &= X_D(\beta^* + \nu), \end{aligned}$$

which implies that $\widehat{\gamma}_T = \gamma^*_T - X_T \nu$ *and* $\widehat{\gamma}_{T^c} = -X_{T^c}\nu$, $X_D \nu = 0$. *Let* $u = X\nu$. *Then we obtain* $\widehat{\gamma} = \gamma^* - u$. *As can be seen, equation* (2.16) *requires that* $u = \vec{0}$, *which essentially implies* $\widehat{\gamma} = \gamma^*$, *and thus* $\text{supp}(\widehat{\gamma}) = \text{supp}(\gamma^*)$.

2.5.2 Optimal Debugger via MILP

The above analysis is also useful in practice for providing a method for designing $\widetilde{X}$. Consider the following optimization problem:

$$\max_{K \subseteq [n], |K| \leqslant t, u \in \mathbb{R}^n, \nu \in \mathbb{R}^d} \|u_K\|_1 - \|u_{K^c}\|_1, \tag{2.17a}$$

$$\text{subject to } u = X\nu, X_D \nu = 0, \|u\|_\infty \leqslant 1. \tag{2.17b}$$

By Theorem 2.14 and Remark 2.17, we immediately conclude that if the problem (2.17) has the unique solution $(u, \nu) = (\vec{0}, \vec{0})$, then a debugger who queries the points indexed by D cannot be beaten by a bug generator who introduces at most t bugs.

Based on this argument, we can construct a bilevel optimization problem for the debugger to solve by further minimizing the objective (2.17a) with respect to $D \subseteq [n]$ such that $|D| \leqslant m$. The optimization problem can then be transformed into a minimax MILP:

$$
\begin{aligned}
&\min_{\xi \in \{0,1\}^n} \max_{\substack{a, a^+, a^- \in \mathbb{R}^n, \\ u, u^+, u^- \in \mathbb{R}^n, v \in \mathbb{R}^d, \\ z, w \in \{0,1\}^n}} \sum_{j=1}^{n} a_j^+ - a_j^-, \\
&\text{subject to } \Big\{ u = Xv, u = u^+ - u^-, u^+, u^- \geqslant 0, \\
&\qquad a = u^+ + u^-, u^+ \leqslant z,\ u^- \leqslant (\mathbb{1}_n - z), \\
&\qquad \sum_{i=1}^{n} w_i \leqslant t, a^+ \leqslant Mw,\ a^- \leqslant M(\mathbb{1}_n - w), \\
&\qquad a = a^+ + a^-, a^+ \geqslant 0, a^- \geqslant 0, \\
&\qquad \sum_{i=1}^{n} \xi_i \leqslant m, u \leqslant (\mathbb{1}_n - \xi), u \geqslant -(\mathbb{1}_n - \xi). \Big\}
\end{aligned}
\tag{2.18}
$$

Theorem 2.18 (MILP for debugging)**.** *If the optimization problem* (2.18) *has the unique solution* $(u, v) = (\vec{0}, \vec{0})$*, then the debugger can add* m *points indexed by* $D = \text{supp}(\xi)$ *to achieve support recovery.*

Remark 2.19. *For more information on efficient algorithms for optimizing minimax MILPs, we refer the reader to the references Tang et al.* (*2016*)*, Xu and Wang* (*2014*)*, and Zeng and An* (*2014*)*.*

2.6 Experiments

In this section, we empirically validate our Lasso-based debugging method for support recovery. The section is organized as follows:

- Subsection 2.6.1, corresponding to Section 2.3, contains a number of experiments which investigate the performance of our proposed debugging formulation.
- Subsection 2.6.2, corresponding to Section 2.4, studies the proposed tuning parameter selection procedure.
- Subsection 2.6.3 studies the Lasso-based debugging method with a clean data pool, including the proposed MILP algorithm from Section 2.5.

We also compare our proposed method to alternative methods motivated by existing literature.

We begin with an outline of the experimental settings used in most of our experiments:

S1 Generate the feature design matrix $X \in \mathbb{R}^{n \times p}$ by sampling each row i.i.d. from $\mathcal{N}(\vec{0}_p, I_{p \times p})$.

S2 Generate $\beta^* \in \mathbb{R}^p$, where each entry β_i^* is drawn i.i.d. from $\mathtt{Unif}(-1, 1)$.

S3 Generate $\epsilon \in \mathbb{R}^n$, where each entry ϵ_i is drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$.

S4 Generate the bug vector $\gamma^* \in \mathbb{R}^n$, where we draw $\gamma_i^* = (10\sqrt{\log(2n)}\sigma + \mathtt{Unif}(0, 10)) \cdot \mathtt{Bernoulli}(\pm 1, 0.5)$ for $i \in [t]$ and take $\gamma_i^* = 0$ for the remaining positions.

S5 Generate the labels by $y = X\beta^* + \epsilon + \gamma^*$.

These five steps produce a synthetic dataset (X, y); we will specify the particular parameters (n, p, t, σ) in each task. If we use a real dataset, the first step changes to [S1']:

S1' Given the whole data pool X_{real}, uniformly sample n data points from it to construct X.

In the plot legends, we will refer to our Lasso-based debugging method as "debugging." We may also invoke a postprocessing step on top of debugging, called "debugging + postprocess," which first runs the Lasso optimization algorithm to obtain $\widehat{\gamma}$ and an estimated support set $\hat{T}$, then removes the points $(X_{\hat{T},\cdot}, y_{\hat{T}})$ and runs ordinary least squares on the remaining points to obtain $\widehat{\beta}$.

2.6.1 Support Recovery

In this section, we design two experiments. The first experiment investigates the influence of the fraction of bugs $c_t := \frac{t}{n}$ on the three assumptions imposed in our theory and the resulting recovery rates. We will vary the design of X using different datasets. The second experiment compares debugging with four alternative regression methods, using the precision-recall metric. Note that we will take the tuning parameter $\lambda = 2\frac{\sqrt{\log 2(n-t)}}{n}$ for these experiments, since the other outlier detection methods we use for comparison do not propose a way to perform parameter tuning. We will explore the performance of the proposed algorithm for parameter selection in the next subsection.

Number of Bugs vs. Different Measurements

Our first experiment involves four different datasets with different values of n and c_t. We track the performance of the three assumptions (Assumptions 2.3–2.5) and the subset/exact recovery rates, which measure the fraction of experiments which result in subset/exact recovery. The first dataset is generated using the synthetic mechanism described at the beginning of Section 2.6, with $p = 15$. The other three datasets are chosen from the UCI Machine Learning Repository: Combined Cycle Power Plant[1],

[1] http://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant

temperature forecast[2], and YearPredictionMSD[3]. They are all associated to regression tasks, with varying feature dimensions (4, 21, and 90, respectively). In the temperature forecast dataset, we remove the attribute of station and date from the original dataset, since they are discrete objects. For each of the UCI datasets, after randomly picking n data points from the entire data pool, we normalize the subsampled dataset according to $X_{\cdot,j} = \frac{X_{\cdot,j} - \frac{1}{n}\sum_{i\in[n]} X_{i,j}}{\mathrm{std}[X_{\cdot,j}]}$, where std represents the standard deviation.

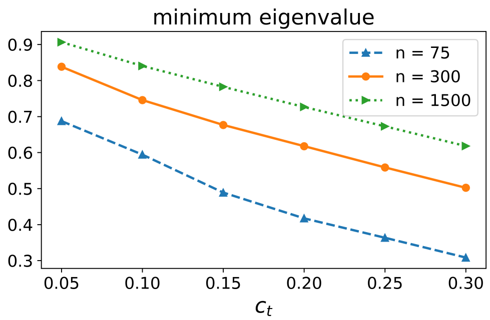
The results are displayed in Figure 2.1. For the minimum eigenvalue assumption, a key observation from all datasets is that the minimum eigenvalue becomes larger (improves) as n increases, and becomes smaller as c_t increases. For the mutual incoherence assumption, the synthetic dataset satisfies the condition with less than 15% outliers. The Combined Cycle Power Plant dataset has mutual incoherence close to 1 when c_t is approximately 20%-25%, and the mutual incoherence condition of the YearPredictionMSD dataset approaches 1 when c_t is approximately 5%. Therefore, we see that the validity of the assumption highly depends on the design of X. For the gamma-min condition, as c_t increases, we need more obvious (larger $\min_i |\gamma_i^*|$) outliers. Finally, with larger n and smaller c_t, the subset/exact recovery rate improves.

Effectiveness for Recovery

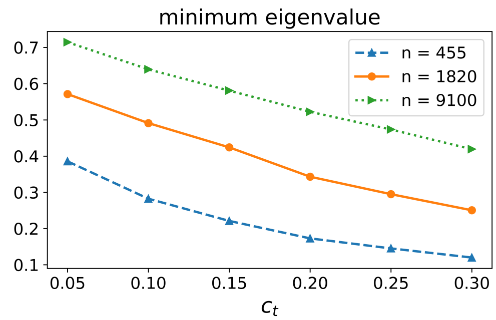
The second experiment compares our debugging method to other proposed methods in the robust statistics literature. We compare our method with the Fast LTS Rousseeuw and Van Driessen (2006), E-lasso Nguyen and Tran (2013), Simplified Θ-IPOD She and Owen (2011), and Least Squares methods. E-lasso is similar to our formulation, except it includes an additional penalty with β. The Simplified Θ-IPOD method iteratively

[2] `http://archive.ics.uci.edu/ml/datasets/Bias+correction+of+numerical+prediction+model+temperature+forecast`

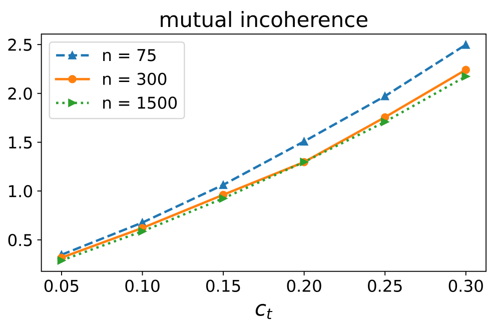
[3] `http://archive.ics.uci.edu/ml/datasets/YearPredictionMSD`



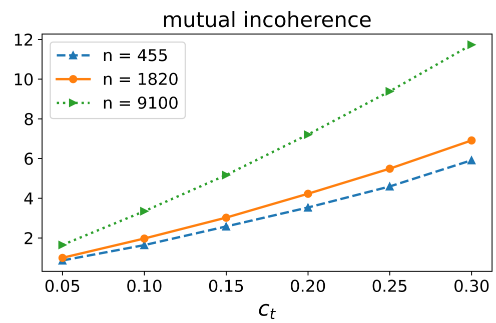
(a) Synthetic dataset



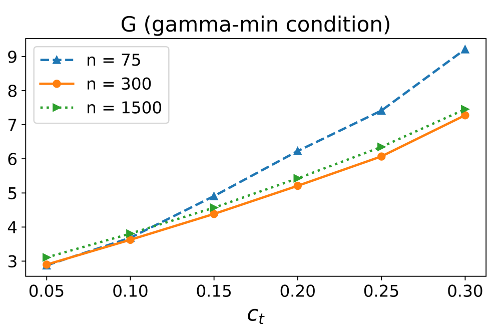
(b) YearPredictionMSD dataset



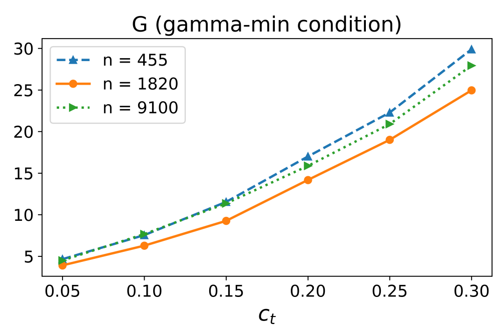
(c) Synthetic dataset



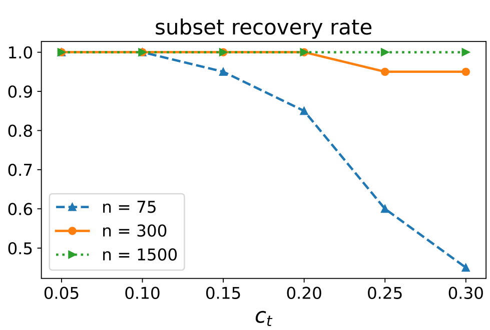
(d) YearPredictionMSD dataset



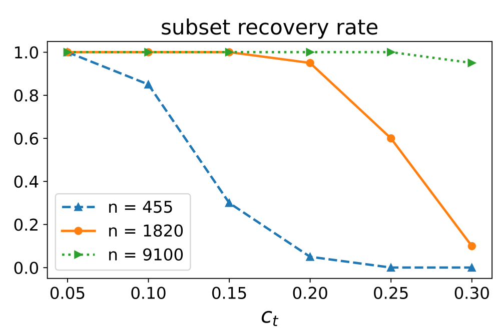
(e) Synthetic dataset



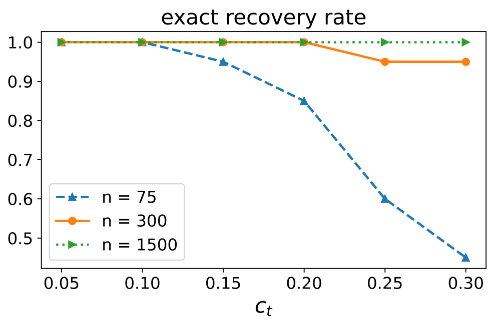
(f) YearPredictionMSD dataset



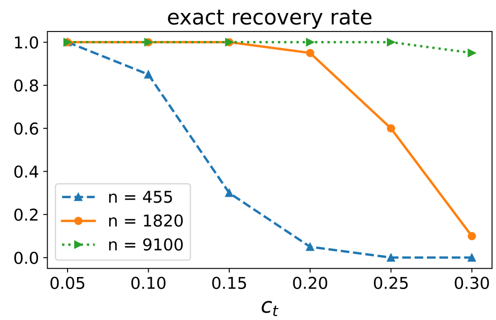
(g) Synthetic dataset



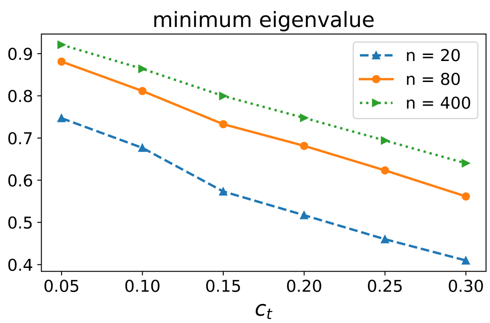
(h) YearPredictionMSD dataset



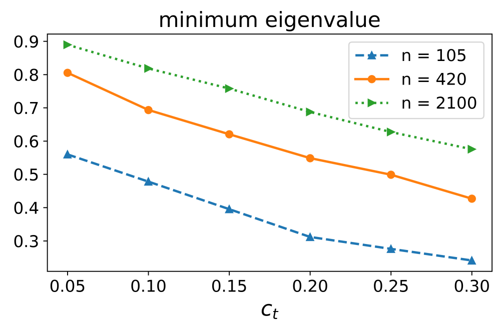
(i) Synthetic dataset



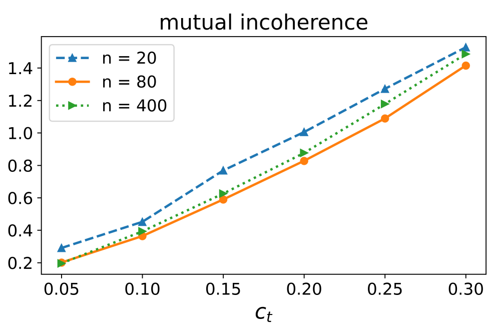
(j) YearPredictionMSD dataset



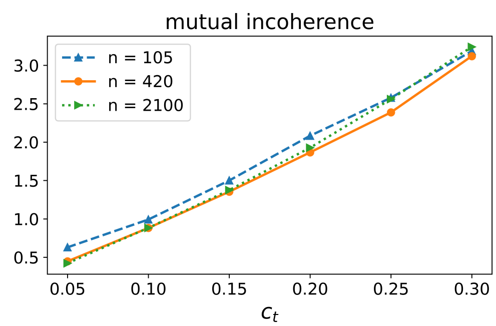
(k) Combined Cycle Power Plant dataset



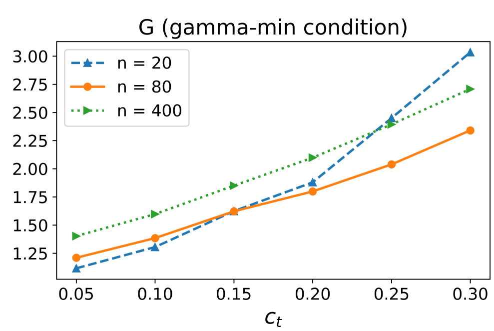
(l) Temperature forecast dataset



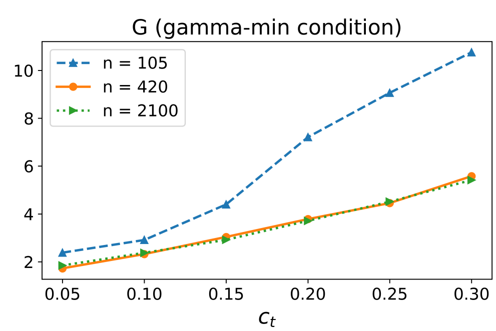
(m) Combined Cycle Power Plant dataset



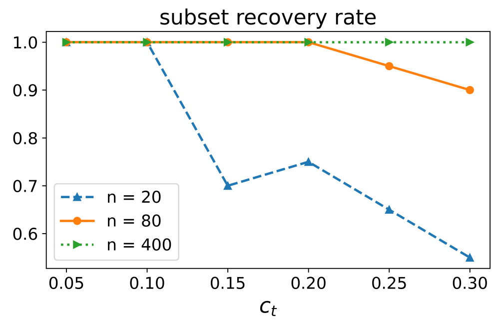
(n) Temperature forecast dataset



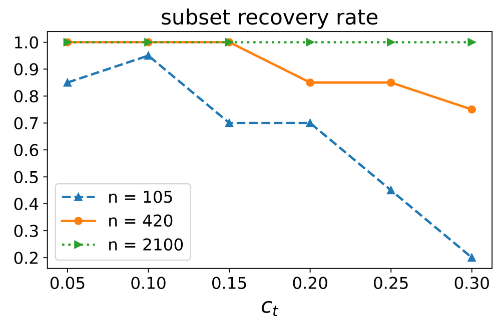
(o) Combined Cycle Power Plant dataset



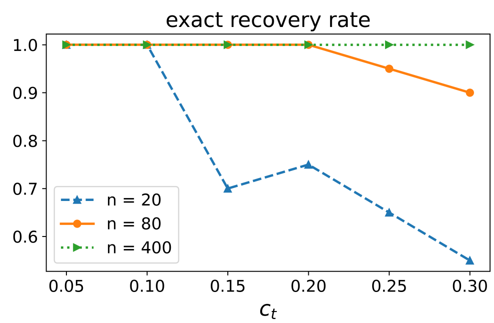
(p) Temperature forecast dataset



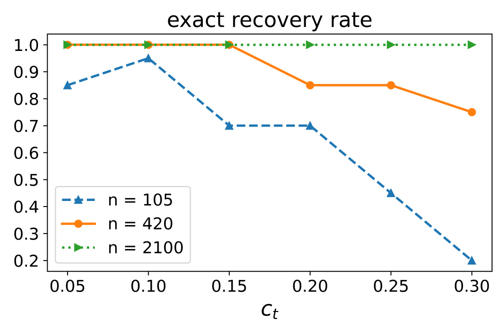
(q) Combined Cycle Power Plant dataset



(r) Temperature forecast dataset



(s) Combined Cycle Power Plant dataset



(t) Temperature forecast dataset

Figure 2.1: Five Measurements on Four Datasets. Three different n's are of values $5p$, $20p$, and $100p$. The variance σ is set to 0.1. The tuning parameter is set to $\lambda = 2\frac{\sqrt{\log 2(n-t)}}{n}$. Each dot is an average value of 20 random trials.

uses hard thresholding to eliminate the influence of outliers. For the experimental setup, we generate synthetic data with $n = 2000, t = 200, p = 15$, and $\sigma = 0.1$, but replace step [S4] by one of the following mechanisms for generating γ^*:

1. We generate $\gamma_i^*, i \in T$ by $\mathrm{Bernoulli}(\pm 1, 0.5) \cdot (10\sqrt{\log(2n)}\sigma + \mathrm{Unif}(0, 10))$.
2. We generate β' elementwise from $\mathrm{Unif}(-10, 10)$ and take $\gamma_i^* = x_i^\top(\beta' - \beta^*), i \in T$.

The first adversary is random, whereas the second adversary aims to attack the data by inducing the learner to fit another hyperplane. The precision/recall for Fast LTS and Least Squares are calculated by running the method once and applying various thresholds to clip $\widehat{\gamma}$. For the other three methods, we apply different tuning parameters, compute precision/recall for each result, and finally combine them to plot a macro precision-recall curve.

In the left panel of Figure 2.2, Least Squares and Fast LTS reach perfect AUC, while the other three methods have slightly lower scores. In the right panel of Figure 2.2, we see that debugging, E-lasso, and Fast LTS perform comparably well, and slightly better than Simplified Θ-IPOD. Not surprisingly, Least Squares performs somewhat worse, since it is not a robust procedure.

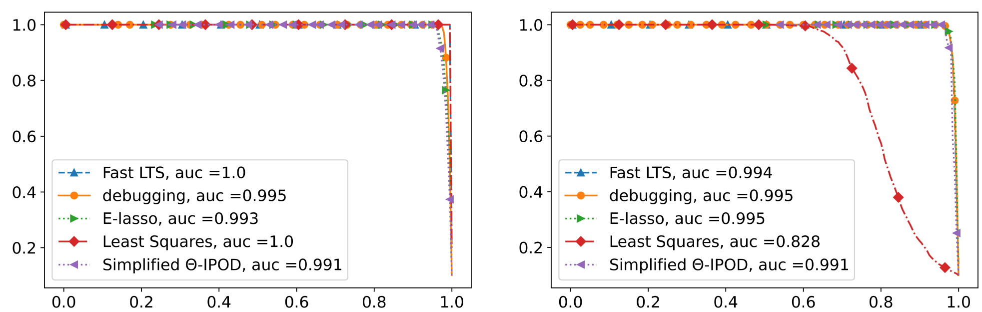


Figure 2.2: Precision Recall Curves over Different Regression Methods. The two plots correspond to the two settings described in the text for generating γ^*. To better view the curves, we only show the dots for every c positions, where c is an interger and different for different methods.

2.6.2 Tuning Parameter Selection

We now present two experimental designs for tuning parameter selection. The first experiment runs Algorithm 1 for both one- and two-pool cases. We will present the recovery rates for a range of n's and c_t's, showing the effectiveness of our algorithm in a variety of situations. The second experiment compares Algorithm 1 in one- and two-pool cases to cross-validation, which is a popular alternative for parameter tuning. Our results indicate that Algorithm 1 outperforms cross-validation in terms of support recovery performance.

We begin by describing the method used to generate the second data pool. Given the first data pool (X, y) and the ground-truth parameters (β^*, σ), we describe two pipelines to generate the second pool. The first pipeline checks m random points of the first pool, with steps [T1-T3]:

T1 Select m points uniformly at random from the first pool to construct $\widetilde{X}$ for the second pool.

T2 Generate $\widetilde{\epsilon} \in \mathbb{R}^m$, where each entry $\widetilde{\epsilon}_i$ is drawn i.i.d. from $\mathcal{N}(0, \sigma^2/L)$.

T3 Generate the labels by $\widetilde{y} = \widetilde{X}\beta^* + \widetilde{\epsilon}$.

When the debugger is able to query features of clean points from a distribution $\mathcal{P}_X$, we can use a second pipeline, where [T1] is replaced by [T1′]:

T1′ Independently draw m points from $\mathcal{P}_X$ to construct $\widetilde{X}$.

Verification of Algorithm 1

We use the default procedure for generating the synthetic dataset, with parameters $p = 15$, $\sigma = 0.1$, and $t = c_t n$, where c_t ranges from 0.05 to 0.4 in increments of 0.05. In all cases, we input $\bar{c} = 0.2$ and $\lambda_u = \frac{2\|P_X^{\perp} y\|_\infty}{n}$ in Algorithm 1.

Figure 2.3 displays the results for $n \in \{1, 2, 3, 4, 5, 10, 20, 30\} \cdot 10^3$. First, we see that Algorithm 1 achieves exact support recovery in all 20 trials in the yellow area. Second, the exact recovery rate increases with increasing n and decreasing c_t, showing that the algorithm is particularly useful for large-scale data sets. This trend can also be seen from the requirement on

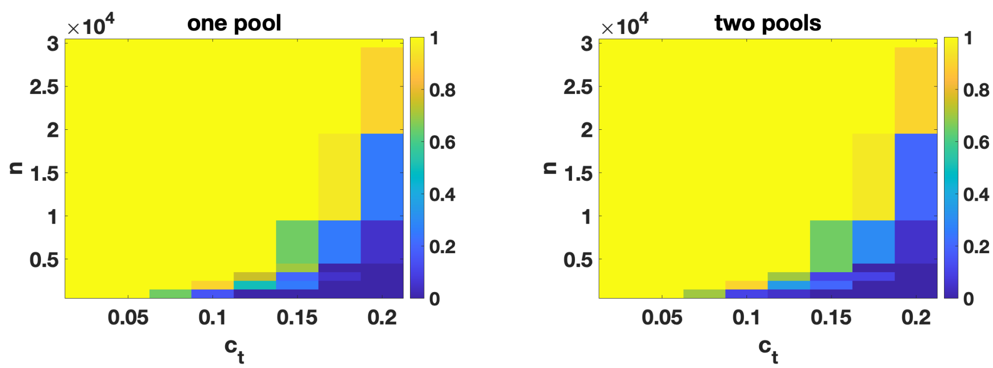


Figure 2.3: Exact Recovery Rate over 20 Trials. The recovery rate is shown in different cases varying by fraction of outliers c_t and n. The left subfigure is for one-pool case and the right subfigure is for two-pool case. We set $m = 100, L = 5$ for the second pool.

n imposed in Theorem 2.12. In particular, we see that the contour curve for the exact recovery rate matches the curve of $(1 - c_t)^{-\frac{1}{1-2c_n}}$ for some constant $c_n \in (0, \frac{1}{2})$. However, a downside of Algorithm 1 is that it does not fully take advantage of the second pool in the two-pool case, as the left panel and the right panel display similar results.

Effectiveness of Tuning Parameter Selection

We now compare our method for tuning parameter selection to cross-validation. We also use the postprocessing step described at the beginning of the section. Four measurements are presented, including two recovery rates, the ℓ_2-error of $\widehat{\beta}$, and the runtime. In both the one- and two-pool cases, we use our default methods for generating synthetic data, and we set $\bar{c} = 0.2$ for all the experiments.

The cross-validation method for the one-pool case splits the dataset into training and testing datasets with the ratio of $8 : 2$, then selects λ with the smallest test error, $\|X_{test}\widehat{\beta} - y_{test}\|_2$. The procedure for the two-pool case is to run the Lasso-based debugging method with a list of candidate λ's and test it on the second pool. Finally, we select the λ value with the smallest test error, $\|\widetilde{X}\widehat{\beta} - \widetilde{y}\|_2$. We use 15 candidate values for λ, spaced evenly on a log scale between 10^{-6} and $\lambda_u = \frac{2\|P_X^\perp y\|_\infty}{n}$.

Figure 2.4 compares the results in the one-pool case. We note that cross-validation does not perform very well for all the measurements except $\|\widehat{\beta} - \beta^*\|_2$. Specifically, it does not work at all for subset support recovery, since cross-validation tends to choose very small λ values. For the ℓ_2-error, we see that for small values of c_t, our algorithm can select a suitable choice of λ, so that after removing outliers, we can fit the remaining points very well. This is why the debugging + postprecessing methods gives the lowest error. As c_t increases, our debugging method shows poorer performance in terms of support recovery, resulting in larger ℓ_2-error for $\widehat{\beta}$. Although cross-validation seems to perform well, carefully designed adversaries may still destroy the good performance of cross-validation, since its test dataset could be made to contain numerous buggy points.

Figure 2.5 displays the results for the two-pool experiments, which are qualitatively similar to the results of the one-pool experiments. We emphasize that our method works well for support recovery; furthermore, the methods exhibit comparable performance in terms of the ℓ_2-error. The slightly larger error of our debugging method can be attributed to the bias which arises from using an ℓ_1-norm instead of an ℓ_0-norm.

2.6.3 Experiments with Clean Points

We now focus on debugging methods involving a second clean pool. We have three experimental designs: First, we study the influence of m on support recovery. Second, we compare debugging with alternative methods suggested in the literature. Third, we study the performance of our proposed MILP debugger, where we compare it to three other simple strategies. Different strategies for selecting clean points correspond to changing step [T1] in the setup described above.

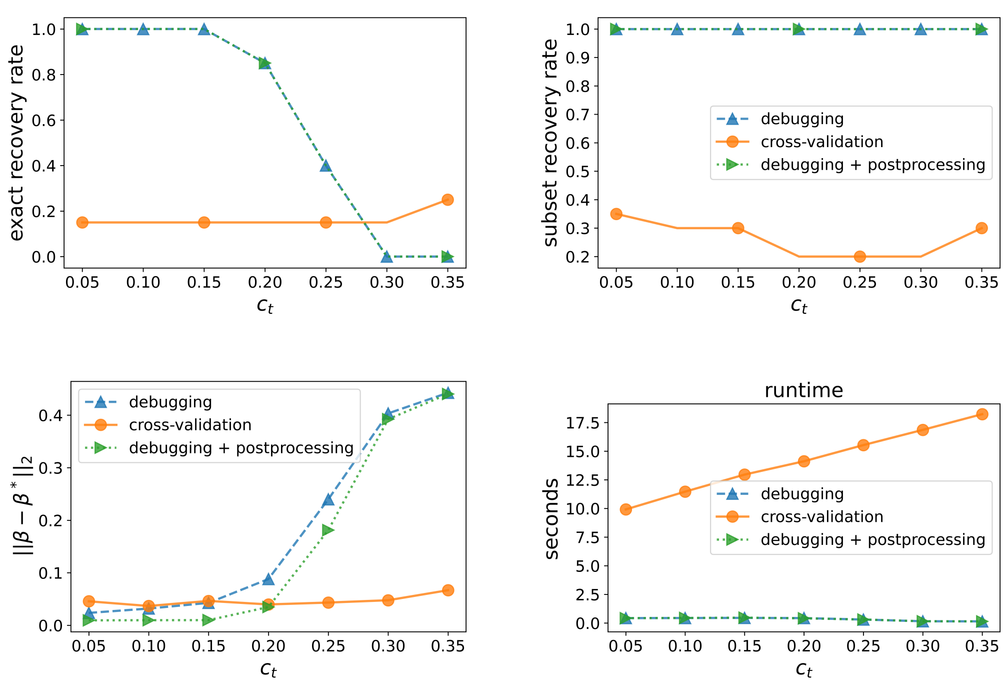


Figure 2.4: Effectiveness of Tuning Parameter Selection (One Pool). Each dot is the average result of 20 random trials. We set $n = 2000$, $p = 15$, and $\sigma = 0.1$.

Number of Clean Points vs. Exact Recovery

In this subsection, we present two experiments involving synthetic and YearPredictionMSD datasets, respectively ,to see how m affects the exact recovery rate. Recall that the pipeline for generating the first pool is described at the beginning of Section 2.6. For the second pool, we use steps [T'1, T2, T3] for the synthetic dataset, where we assume $\mathcal{P}_X$ is standard Gaussian. We take steps [T1-T3] for YearPredictionMSD to check the sample points in the first pool.

Recall that the YearPredictionMSD dataset is designed to predict the release year of a song from audio features. The dataset consists of 515,345 songs, each with 90 audio features. Therefore, for both experiments, we

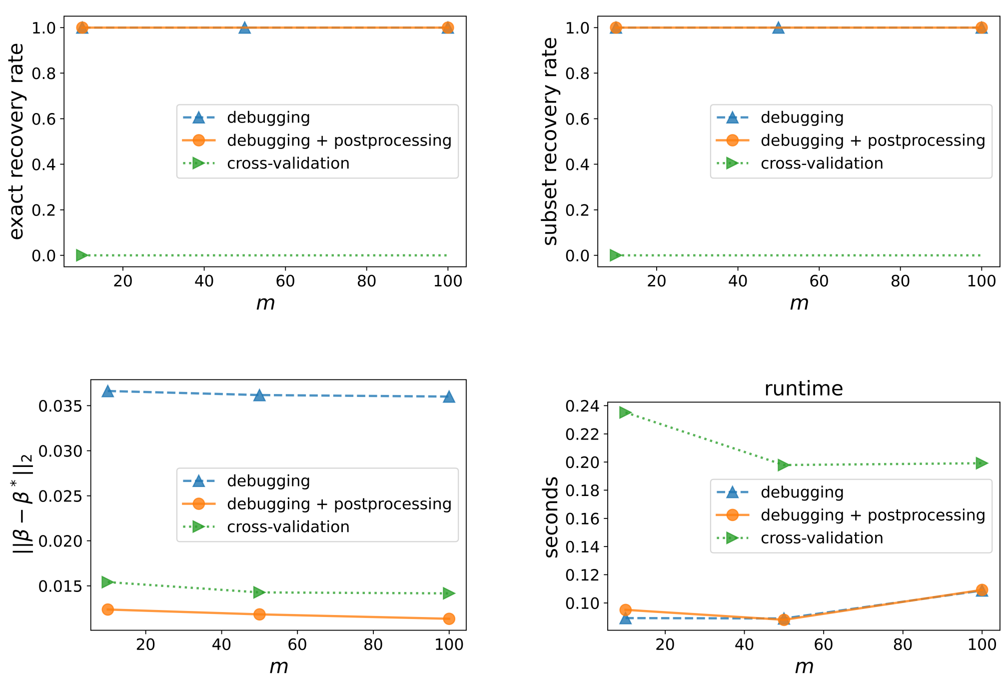


Figure 2.5: Effectiveness on Tuning Parameter Selection (Two Pools). Each dot is the average result of 20 random trials. We set $n = 1000, p = 15, t = 100, L = 5$, and $\sigma = 0.1$.

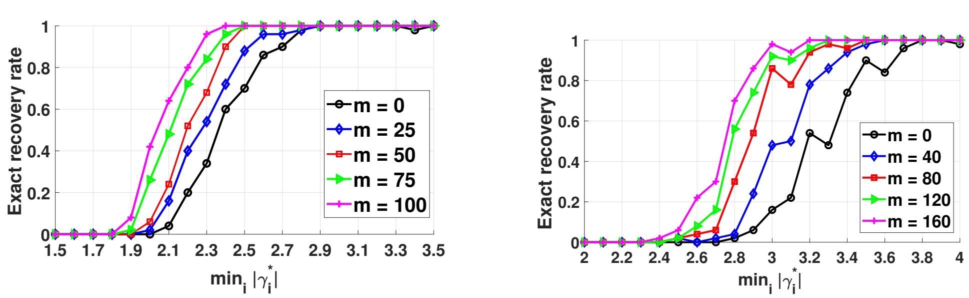


Figure 2.6: Minimal Gamma vs. Exact Recovery Rate on Synthetic Data. We run 50 trials for each dot and compute the average.

set $n = 500, t = 50, p = 90, \sigma = 0.1$, and $L = 10$, and take $\lambda = 2.5\frac{\sqrt{\log(n-t)}}{n}$.

From Figure 2.6, we see that the phenomena are similar for the two different design matrices. In particular, increasing the number of clean points helps with exact recovery. For instance, in the left subfigure, for $m = 0$, when $\min_i |\gamma_i^*| > 2.9$, the exact recovery rate goes to 1. For $m = 100$, the exact recovery rate goes to 1 when $\min_i |\gamma_i^*| > 2.4$. Also, the slope of the curve for larger m is sharper. Thus, adding a second pool helps relax the gamma-min condition.

Comparisons to Methods with Clean Points

In this experiment, we compare the debugging method for two pools with other methods suggested by the machine learning literature. We generate synthetic data using the default first-pool setup with $n = 1000, p = 15, t = 100$, and $\sigma = 0.1$, and we run [T1-T3] to generate the second pool using different values of m. For our proposed debugging method, we use Algorithm 1 to select the tuning parameter. We compare the following methods: (1) debugging + postprocessing, (2) least squares, (3) simplified noisy neural network, and (4) semi-supervised eigvec. The least squares solution is applied using $\left\{ \begin{pmatrix} X \\ \widetilde{X} \end{pmatrix}, \begin{pmatrix} y \\ \widetilde{y} \end{pmatrix} \right\}$.

The simplified noisy neural network method borrows an idea from Veit et al. Veit et al. (2017), which is designed for image classification tasks for a datasets with noisy and clean points. This work introduced two kinds of networks and combines them together: the "Label Cleaning Network," used to correct the labels, and the "Image Classifier," which classifies images using CNN features as inputs and corrected labels as outputs. Each of them is associated with a loss, and the goal is to minimize the sum of the losses. Let $w \in \mathbb{R}, \beta_1 \in \mathbb{R}^d$, and $\beta_2 \in \mathbb{R}^d$ be the variables to be optimized. For our linear regression setting, we design the "Label Cleaning Network" by defining $\hat{c}_i = y_i w - x_i^\top \beta_1$ as the corrected labels for both noisy and clean datasets. Then we define the loss $\mathcal{L}_{clean} = \sum_{i \in cleanset} |\tilde{y}_i - y_i w - x_i^\top \beta_1|$. The "Image Classifier" is modified to the regression setting using predictions of $x_i^\top \beta_2$ and the squared loss. Therefore, the classification loss can be formalized as $\mathcal{L}_{classify} = \sum_{i \in cleanset}(x_i^\top \beta_2 - \tilde{y}_i)^2 + \sum_{i \in noisyset}(x_i^\top \beta_2 - \hat{c}_i)$. Together, the optimization problem becomes

$$\min_{\substack{\beta_1 \in \mathbb{R}^d, \beta_2 \in \mathbb{R}^d \\ w \in \mathbb{R}}} \left\{ \sum_{i \in cleanset} \{(x_i^\top \beta_2 - \tilde{y}_i)^2 + |\tilde{y}_i - w y_i - x_i^\top \beta_1|\} \right. \\ \left. + \sum_{i \in noisyset} (x_i^\top \beta_2 - w y_i - x_i^\top \beta_1)^2 \right\}.$$

We use gradient descent to do the optimization, and initialize it with $w = 0$ and $\beta_1 = \beta_2 = \beta_{ls}$. The optimizer $\widehat{\beta}_2$ is used for further predictions. We then calculate $\widehat{\gamma} = y - X\widehat{\beta}_2$. For gradient descent, we will validate multiple step sizes and choose the one with the best performance on the squared loss of the clean pool.

The method "semi-supervised eigvec" is from Fergus et al. Fergus et al. (2009), and is designed for the semi-supervised classification problem. It also contains an experimental setting that involves noisy and clean data. To further apply the ideas in our linear regression setting, we make the following modifications: Define the loss function as

$$J(f) = f^\top L f + \left(f - \begin{pmatrix} y \\ \tilde{y} \end{pmatrix}\right)^\top \Lambda \left(f - \begin{pmatrix} y \\ \tilde{y} \end{pmatrix}\right),$$

where $L = D - W(\varepsilon)$ is the graph Laplacian matrix and Λ is a diagonal matrix whose diagonal elements are $\Lambda_{ii} = \lambda$ for clean points and $\Lambda_{ii} = \frac{\lambda}{c}$ for noisy points. In the classification setting, $f \in \mathbb{R}^{n+m}$ is to be optimized. The idea is to constrain the elements of f by injecting smoothness/similarity using the Laplacian matrix L. Since we assume the linear regression model, we can further plug in $f = \begin{pmatrix} X \\ \tilde{X} \end{pmatrix} \beta$. Our goal is then to estimate β by minimizing $J(\beta)$. As suggested in the original paper, we use the range of values $\varepsilon \in [0, 1, 1, 5], c \in [1, 10, 50]$, and $\lambda \in [1, 10, 100]$. We will evaluate all 36 possible combinations and pick the one with the smallest squared loss on the clean pool.

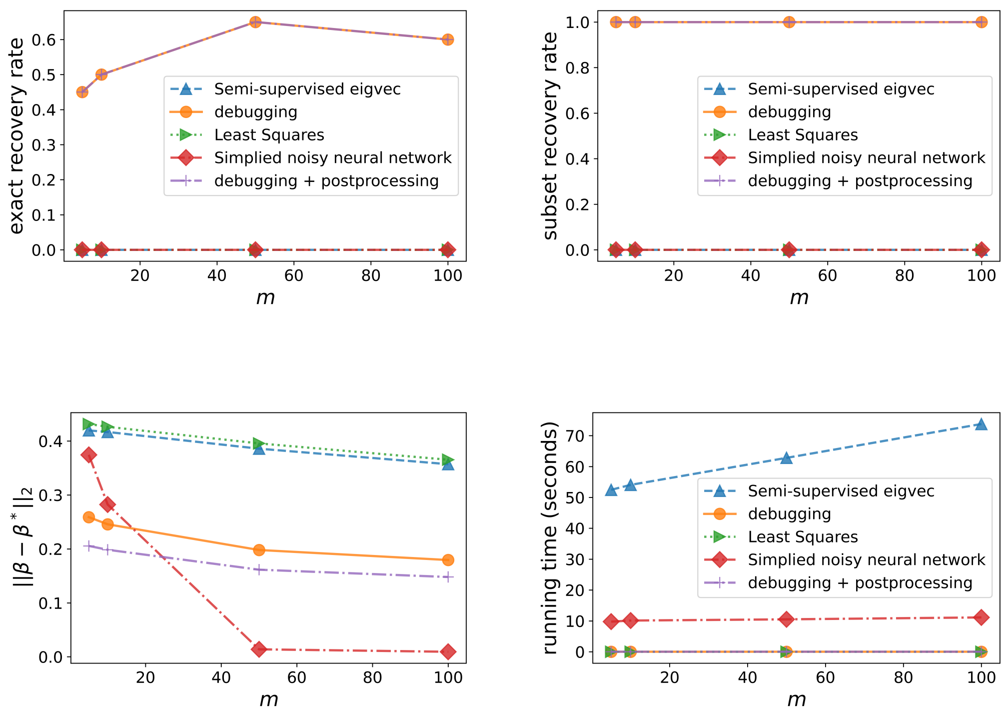


Figure 2.7: Comparison to Methods involving Clean Points. Each dot is the average result of 20 random trials. We use the synthetic data setting, with $n = 500, p = 15, \sigma = 0.1, t = 0.1n$, and $\min_i |\gamma_i^*| = 10\sqrt{\log 2n}\sigma$. The clean data pool is randomly chosen from the first pool without replacement; we query the labels of these chosen points.

The results are shown in Figure 2.7. We observe that only the debugging method is effective for support recovery, as we have carefully designed our method for this goal. The method from Veit et al. Veit et al. (2017)

works best in terms of ℓ_2-error of β, especially when m is large. The semi-supervised method, like least squares, does not perform well, possibly because it does not consider replacing/removing the influence of the noisy dataset.

Effectiveness on Second Pool Design

We now provide experiments to investigate the design of the clean pool, corresponding to Section 2.5. We use the Concrete Slump dataset[4], where $p = 7$. We limit our study to small datasets, since the runtime of the MILP optimizer is quite long. We report the performance of the MILP debugging method in both noiseless and noisy settings. In our experiments, we compare the performance of the MILP debugger to a random debugger and a natural debugging method: adding high-leverage points into the second pool. In other words, D.milp selects m clean points to query from running the MILP (2.18); D.leverage selects the m points with the largest values of $x_i^\top (X^\top X)^{-1} x_i$; and D.random randomly chooses m points from the first pool without replacement. After choosing the clean pool, the debugger applies the Lasso-based algorithm. In Zhang et al. Zhang et al. (2018), all the second pool points are chosen either randomly or artificially. Therefore, we may consider D.random as an implementation of the method in Zhang et al. Zhang et al. (2018), which will be compared to our D.milp.

In the noiseless setting, we define β^* to be the least squares solution computed from all data points. We randomly select n data points as the x_i's. For D.milp and D.leverage, since the bug generator knows their strategies or the selected D, it generates bugs according to the optimization problem (2.17). Let $T \subseteq [n]$ be the index set of the t largest $|u_i|$'s, for $i = 1, \ldots, n$. The bug generator takes $\gamma_T^* = u_T$ if the solution u is nonzero, and otherwise randomly generates a subset T of size t to create $\gamma_T^* = \vec{1}$. Thus, $y_i = x_i^\top \beta^* + \gamma^*$. For D.onepool, the bug generator follows

[4] https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test

the above description with $D = \emptyset$. The orange bars indicate whether the bug generator succeeds in exact recovery in the one-pool case. For D.random, the bug generator generates bugs using the same mechanism as for D.onepool. Note the above bug generating methods are the "worst" in the sense of signed support recovery: The debuggers run (2.14) using their selected X_D. From Figure 2.8, there is an obvious advantage of D.milp over D.onepool and D.leverage. This suggests improved performance of our MILP algorithm. D.random is sometimes successful even when n and t are small because the bug generator cannot control the randomness, but it performs worse than D.milp overall.

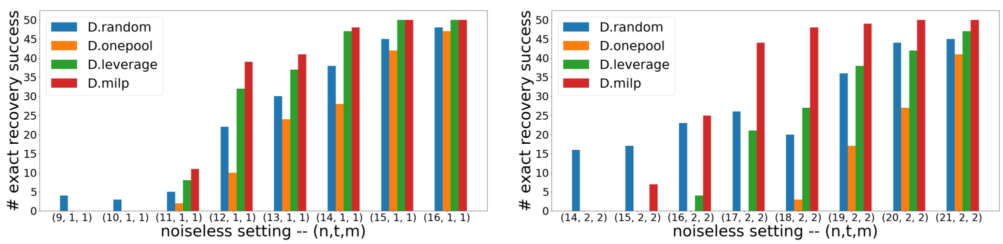


Figure 2.8: Comparison between D.milp and Other Debugging Strategies in Noiseless Settings. Each setting is an average over 50 random trials.

In the noisy setting, we define β^* to be the least squares solution computed using the entire data set. We randomly select n data points as the x_i's. For D.milp and D.leverage, since the bug generator knows their strategies or the selected D, it generates bugs via the optimization problem (2.17): taking $\gamma_T^* = u_T$ if the solution u is nonzero for T being the indices of the largest t elements of $|u|$, and otherwise randomly generating a subset T of size t to create $\gamma_T^* = \vec{1}$. Thus, $y_i = x_i^\top \beta^* + \gamma^* + \mathcal{N}(0, 0.01)$. Note that having $\gamma_T^* = u_T$ if the solution u is nonzero gives incorrect signed support recovery, which is proved in Appendix A.1.5. This is related to what we have claimed in Remark 2.16 above. For D.onepool, the bug generator follows the above description with $D = \emptyset$. The orange bars indicate whether the bug generator succeeds in exact recovery in the one-pool case. For

D.random, since it is not deterministic, the bug generator does not know D and acts in the same way as in the one-pool case. Note that the above bug generating methods are the "worst" in the sense of signed support recovery. From Figure 2.9, there is an obvious advantage of D.milp over D.onepool and D.leverage. Our theory only guarantees the success of D.milp in the *noiseless* setting, so the experimental results for the noisy setting are indeed encouraging.

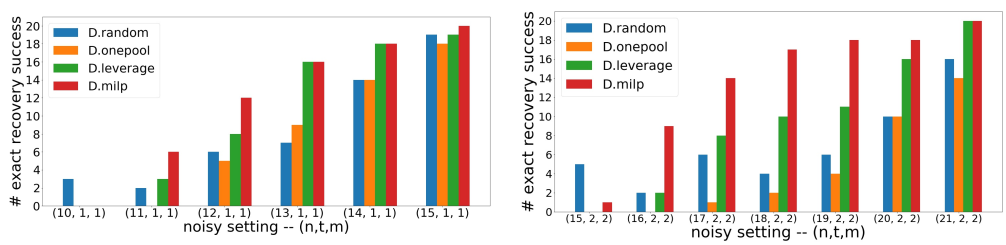


Figure 2.9: Comparison between MILP Strategy and Others. In each setting, we run 20 random simulations.

Debugging in practice: The algorithm for minimax optimization has been executed by running all $\binom{n}{m}$ possible choices of clean points for the outer loop; for each outer loop, we then run the inner maximization. For optimal debugging in practice, i.e., n, t, and m being large, some recent work provides methods for efficiently solving the minimax MILP Tang et al. (2016). Note that the MILP debugger can be easily combined to other heuristic methods: one can run the MILP, and if there is a nonzero solution, we can follow it to add clean points. Otherwise, we can switch to other methods, such as choosing random points or high-leverage points.

2.7 Conclusion

We have developed theoretical results for machine learning debugging via M-estimation and discussed sufficient conditions under which support recovery may be achieved. As shown by our theoretical results and illustrative examples, a clean data pool can assist debugging. We have also designed a tuning parameter algorithm which is guaranteed to obtain exact support recovery when the design matrix satisfies a certain concentration property. Finally, we have analyzed a competitive game between the bug generator and the debugger, and analyzed a mixed integer optimization strategy for the debugger. Empirical results show the success of the tuning parameter algorithm and proposed debugging strategy.

Our work raises many interesting future directions. First, the question of how to optimally choose the weight parameter η remains open. Second, although we have mentioned several efficient algorithms for bilevel mixed integer programming, we have not performed a thorough comparison of these algorithms for our specific problem. Third, although our MILP strategy for second pool design has been experimentally found to be effective in a noisy setting, we do not have corresponding theoretical guarantees. Fourth, our proposed debugging strategy is a one-shot method, and designing adaptive methods for choosing the second pool constitutes a fascinating research direction. Finally, the analysis of our tuning parameter algorithm suggests that a geometrically decreasing series might be used as a grid choice for more general tuning parameter selection methods, e.g., cross validation—in practice, one may not need to test candidate parameters on a large grid chosen linearly from an interval. Lastly, it would be very interesting to extend the ideas in this work to regression or classification settings where the underlying data do not follow a simple linear model.

3 ON THE IDENTIFIABILITY OF MIXTURES OF RANKING MODELS

Parameter estimation in mixtures of ranking models have been studied in last decade. While identifiability results for probabilistic models such as the Plackett-Luce (PL) and Mallows models have been established, the problem of identifiability has still remained largely open for a standard mixture with the Bradley-Terry-Luce (BTL) model. In our work, we fill this gap for two mixtures of BTL models using tools from algebraic geometry. We provide two conditions that may be checked to verify the identifiability of all mixtures of ranking models that can be transformed into polynomial systems, and then study several concrete examples. Our main result and techniques on generic identifiability can be applied more broadly to general polynomial machine learning models.

3.1 Introduction

Ranking is an important topic in machine learning, where the goal is to rank collections of items based on votes involving smaller subsets of items. Four categories are commonly studied by researchers: pointwise ranking (Balabanović and Shoham, 1997; Resnick et al., 1994; Yu et al., 2009), pairwise ranking (Jamieson and Nowak, 2011; Tsukida and Gupta, 2011; Wauthier et al., 2013; Chen et al., 2013; Chen and Suh, 2015), and listwise ranking (Pendergrass and Bradley, 1959; Khetan and Oh, 2016; Zhao and Xia, 2018). Pointwise ranking collects data based on ratings of items from users. Pairwise ranking collects data based on votes between pairs of items from users. Listwise ranking collects data by asking users to linearly order lists of items. Although such voter data is popular in areas such as advertising, recommendation systems, and player rankings (Wauthier et al., 2013), noise in ratings can be problematic even when user preferences might be consistent. For example, user A might rate movies i, j, and k as 1,4, and 7 out of 10, respectively, while user B might rate the same movies as 7,8, and 9.

Most research focuses on inferring one potential ordering in user-unspecified settings. However, this may turn out to be a bad assumption in reality, as we now explain: Imagine a case where users rate movies. Adults may have different preferences than children; women may have different preferences than men. Therefore, several potential orderings might better explain the preferences of users. Another issue arises from inconsistent voting, where the data includes votes—even from the same user—which involve "cycles," such as $i_1 \succ i_2$ (the notation $\succ$ means the i_1th item is preferred than the i_2th item), $i_2 \succ i_3$, and $i_3 \succ i_1$. A possible explanation for such inconsistencies is statistical noise, which is absorbed in probabilistic models such as the Bradley-Terry-Luce (BTL) (Bradley and Terry, 1952; Luce, 1959) or Thurstone models (Thurstone, 1927). These methods may cater to a large portion of users or describe an accurate ordering for a large portion of objects, but the low accuracy of predicting new votes sometimes hints that multiple user types are actually present in the population. In such cases, using a mixture model to fit the data may lead to higher prediction accuracy.

We briefly overview existing literature on mixture models. Topics such as clustering, mixtures of Gaussians, and mixtures of linear regressions are well-studied, e.g., McLachlan and Basford (1988); Reynolds (2009); Yi et al. (2014); Li and Liang (2018). Our work falls into the category of mixed logit models, also known as the mixtures of multinomial logistic models (MNLs). Within this category, Train (2009) introduced the mixed logit model as a method for discrete choices, while Ge (2008) discussed simulation approaches for mixed logit models using the Metropolis-Hastings algorithm, and Arora et al. (2013) studied the application of mixed logit models on topic modeling. We focus on the identifiability problem of mixtures of ranking models, assuming a known number of user types but not the detailed user information for each vote. Some related work includes Wu et al. (2015), who studied mixtures models for pairwise ranking problems, where a large number of comparisons of individual users need to be collected. However, we do not require the information of individual users; Oh and Shah (2014) demand more from the known individuals to a few comparisons done by the same type of user. Note that we only assume a simple sample is collected from an unknown user. Other related work includes Chierichetti et al. (2018), who analyzed the identifiability on uniform mixtures of MNLs with pairwise and triplet comparisons; Iannario (2010), who showed the identifiability of mixtures of shifted binomial and uniform discrete models; Awasthi et al. (2014), who proved the uniqueness of Mallows mixture models of two components; Chierichetti et al. (2015), who generalized these results to arbitrary numbers of components (this model is also studied by Lu and Boutilier (2014)); and Zhao et al. (2016), who addressed the identifiability of mixtures of Plackett-Luce models. Although BTL models are quite popular in applications, the identifiability problem for mixtures of BTL models has still remained largely open. Indeed, the techniques proposed in the aforementioned papers, which generally use higher-order moment information, seem to fail for mixtures of BTL models, since pairwise comparisons alone cannot provide such information.

In this study, we affirm the identifiability of mixtures of BTL model in all but a subset of cases of measure zero, which we call "generic identifiability":

Proposition (Informal). *Suppose we have more than 5 items to rank. The two mixtures of BTL models is generically identifiable* (*up to reordering*) *without knowing the mixture probability.*

Suppose we have more than 5 items to rank. Given the mixture probability in the grid $\{0.01, 0.02, \ldots, 0.98, 0.99\}$, *the two mixtures of BTL models is generically* *identifiable* (*up to reordering*).

In the first statement of the proposition, we consider the parameter space of the ranking scores and the mixture probability. In the second statement, the mixture probability is assumed to be known and we consider the parameter space of the ranking scores only. For the second statement, for any mixture probability in $(0,1) \subseteq \mathbb{R}$, we can run some program which will be demonstrated in the paper later to check whether the mixtures of BTL models is generically identifiable or not. We checked a list of them from 0.01 to 0.99. In fact, we conjecture it is true for any mixture probability in $(0,1) \subseteq \mathbb{R}$ without the program checking.

Conjecture. *Given the mixture probability in $(0,1) \subseteq \mathbb{R}$, the two mixtures of BTL models is generically identifiable* (*up to reordering*).

As for the proof techniques, our observation is that the identifiability problem in this model is equivalent to the uniqueness of a polynomial system. Thanks to the development of algebraic geometry, this problem can be manipulated in Zariski topology and then be transferred to Lebesgue measure. More interestingly, besides BTL model, our results can serve for a number of mixtures of ranking models as long as it corresponds to some polynomial system, such as the models in Chierichetti et al. (2018) and Zhao et al. (2016). A higher level of our main result gives conditions under which the generic identifiability holds for polynomial machine learning problem.

Theorem. *Let $\mathscr{P}(\mathbf{t}, \mathbf{x})$ be a polynomial system with variables $\mathbf{x}$ and parameters $\mathbf{t}$. Under the following two assumptions, $\mathscr{P}(\mathbf{a}, \mathbf{x})$ has exactly ℓ solutions in $\mathbb{C}$* (*counted with multiplicity*) *for all $\mathbf{a} \in \mathbb{C}^m$ but a set of $\lambda_m^{\mathbb{C}}$-measure zero.*

Assumption 1. *$\mathscr{P}(\mathbf{a}, \mathbf{x})$ has at least ℓ solutions in $\mathbb{C}$ for all $\mathbf{a} \in \mathbb{C}^m$.*

Assumption 2. *There exists* $\mathfrak{a}' \in \mathbb{C}^m$ *such that* $\mathfrak{a}'$ *does not vanish any coefficients in the Gröbner basis of* $\mathscr{P}(\mathfrak{a}, \mathbf{x})$ *and* $\mathscr{P}(\mathfrak{a}', \mathbf{x})$ *has exactly* ℓ *solutions in* $\mathbb{C}$ (*counted with multiplicity*).

The remainder of the paper is organized as follows. Section 3.2 presents the background knowledge of generic identifiality and some mixtures of ranking models. Section 3.3 is devoted to our conclusions on generic identifiability for general polynomial system. Section 3.4 shows how we apply the conclusions on different mixtures of ranking models, with proving generic identifiability of mixtures of BTL models as an example. Section 3.5 gives conclusions and more discussions.

Notations We write $\lambda_m^{\mathbb{C}}$ to represent the Lebesgue measure on $\mathbb{C}^m$, i.e. the measure induced by the standard Euclidean metric on $\mathbb{C}^m$ given by

$$||z|| := \left(\sum_{i=1}^{m} |z_i|^2 \right)^{1/2} \quad \text{for any } z = (z_1, \dots, z_m) \in \mathbb{C}^m.$$

and write λ_m to represent the Lebesgue measure on $\mathbb{R}^m$, i.e. the measure induced by the standard Euclidean metric on $\mathbb{R}^m$ given by

$$||z|| := \left(\sum_{i=1}^{m} z_i^2 \right)^{1/2} \quad \text{for any } z = (z_1, \dots, z_m) \in \mathbb{R}^m.$$

Hereafter, we identify $\mathbb{R}^m$ as a (closed) subset of $\mathbb{C}^m$ via $\mathbb{R}^m \cong \mathbb{R}^m \times \{\mathbf{0}_m\} \subset \mathbb{R}^{2m} \cong \mathbb{C}^m$.

We write zero-set$(f(x))$ for some polynomial f to be $\{x \mid f(x) = 0\}$.

3.2 Background

We begin with the definition of generic identifiability of a equation system, followed with a number of mixtures of ranking models for their identifiablities.

3.2.1 Generic Identifiability

To start, we write the definition of identifiability. For a fixed parameter $\mathbf{c}$, we call an equation system $F(\boldsymbol{z};\mathbf{c}) = \mathbf{0}$ *identifiable* with variable $\boldsymbol{z}$ if there is a $\boldsymbol{z}^*$ satisfying $F(\boldsymbol{z}^*;\mathbf{c}) = \mathbf{0}$ and there is no $\boldsymbol{z}^\# \neq \boldsymbol{z}^*$ such that $F(\boldsymbol{z}^\#;\mathbf{c}) = \mathbf{0}$, or in other words if $F(\boldsymbol{z};\mathbf{c}) = 0$ has a unique solution.

Let $a_1, \dots, a_n$ and $b_1, \dots, b_n$ represent the scores of n items in two mixtures. As suggested by Oh and Shah (2014) and Chierichetti et al. (2018), there are some counterexample parameters for the identifiability of mixtures of rankings. For example, for some $n \geqslant 3$, the two latent uniform mixtures parameters are as follows:

1. $a_1 = a_2 = t, b_1 = b_2 = \frac{1}{t}, a_i = b_i = 1$ for each $i \in \{3, 4, \dots\}$.
2. $a_1 = b_2 = t, b_1 = a_2 = \frac{1}{t}, a_i = b_i = 1$ for each $i \in \{3, 4, \dots\}$.

Let $\mathbb{P}[i \succ j] = \frac{1}{2}\frac{a_i}{a_i+a_j} + \frac{1}{2}\frac{b_i}{b_i+b_j}$. It is not hard to verify that $\mathbb{P}[i \succ j]$ for all $i, j \in [n]$ are equal for the above two mixtures (See Theorem 2 in Chierichetti et al. (2018)). Therefore, for the parameters $\mathbf{a} = (t, t, 1, \dots, 1)$, $\mathbf{b} = (\frac{1}{t}, \frac{1}{t}, 1 \dots, 1)$ and the equation system $F(\mathbf{x}, \mathbf{y}; \mathbf{a}, \mathbf{b})$ is non-identifiable for any fixed t,

$$F(\mathbf{x}, \mathbf{y}; \mathbf{a}, \mathbf{b}) = \begin{cases} x_1 - t, \\ y_1 - t, \\ \frac{1}{2}\frac{x_1}{x_1+x_i} + \frac{1}{2}\frac{y_1}{y_1+y_i} - \frac{1}{2}, & \forall i = 2, 3, \dots, n, \\ \frac{1}{2}\frac{x_2}{x_2+x_i} + \frac{1}{2}\frac{y_2}{y_2+y_i} - \frac{1}{2}. & \forall i = 3, 4, \dots, n. \end{cases}$$

or

$$F(\mathbf{x}, \mathbf{y}; \mathbf{a}, \mathbf{b}) = \begin{cases} x_1 = t, \\ y_1 = t, \\ \frac{1}{2}\frac{x_1}{x_1+x_i} + \frac{1}{2}\frac{y_1}{y_1+y_i} = \frac{1}{2}, & \forall i = 2, 3, \dots, n, \\ \frac{1}{2}\frac{x_2}{x_2+x_i} + \frac{1}{2}\frac{y_2}{y_2+y_i} = \frac{1}{2}, & \forall i = 3, 4, \dots, n. \end{cases}$$

Note we switch the two forms of writing a equation system later in the paper, for both of which means $F(\mathbf{x}, \mathbf{y}; \mathbf{a}, \mathbf{b}) = \mathbf{0}$. This gives infinite choices of parameters $\mathbf{a}, \mathbf{b}$ by varying $t \in \mathbb{R}$ to have the non-identifiability. Nonetheless, we believe that such choices of $\mathbf{a}, \mathbf{b}$ are special. And we hope that researchers who design algorithms to learn the mixtures of ranking models, like EM algorithm, will not be worried about such non-identifiability cases since they cannot represent most of the cases. Here comes the generic identifiability in this study.

We define *non-identifiable* (*bad*) *parameter set* as the subset of parameters $\mathbf{c}' \in C$ such that $F(\mathbf{z}; \mathbf{c}') = \mathbf{0}$ is non-identifiable. Then, we call a equation system $F(\mathbf{z}; \mathbf{c}) = \mathbf{0}$ is *generic identifiable/unique* on C if the Lebesgue measure of C is positive and the Lebesgue measure of non-identifiable parameter set is zero.

On top of this, when we assume that there is at least one solution for the equation system, it is sufficient to have the generic identifiability of the equation system if we can show a linear transformation of that equation system, such as a subset of the equations, has a unique solution generically.

3.2.2 Illustrative Examples

We review some specific ranking models that we will revisit later in the paper and write specific definitions to their generic identifiabilities.

BTL model

Bradley-Terry-Luce (BTL) model was introduced by Bradley and Terry (Bradley and Terry, 1952) and then studied by Luce (Luce, 1959). It considers the comparison between two items i, j using the probabilistic model,

$$\mathbb{P}\left[i \succ j\right] = \frac{e^{s_i}}{e^{s_i} + e^{s_j}} = \frac{1}{1 + e^{-(s_i - s_j)}},$$

where s_i, s_j represent the ranking scores/weights of i and j. Another parameterization of it replaces the exponentials, leading to,

$$\mathbb{P}\left[i \succ j\right] = \frac{c_i}{c_i + c_j}.$$

As we can see, the comparison takes difference between s_i, s_j while it takes quotient between c_i and c_j. To rank a list of objects, one can apply the BTL model on every two of them, take the comparisons and run a convex optimization to estimate the ranking scores, thus implying the ranks. This procedure assumes that all the comparisons comply with one potential ranking.

Disobeying the above assumption, we consider mixtures of BTL models. Suppose there are two types of users and n objects. Denote U as the indicator of type of user: if a user is of the first type, then $U = 1$; if a user is of the second type, then $U = 2$. Let $\mathbf{a} := [a_1, a_2, \ldots, a_n]^\top$ be the ranking scores for type 1 users and $\mathbf{b} = [b_1, b_2, \ldots, b_n]^\top$ be the scores for type 2 users. We will use $\mathbf{a}_{i:j}$ and $\mathbf{b}_{i:j}$ to represent $(a_i, \ldots, a_j)$ and $(b_i, \ldots, b_j)$ respectively in the paper. Applying BTL model on each type, we write the conditional probability of the comparison between objects i and j,

$$\mathbb{P}\left[i \succ j | U = 1\right] = \frac{a_i}{a_i + a_j}, \quad \mathbb{P}\left[i \succ j | U = 2\right] = \frac{b_i}{b_i + b_j}. \tag{3.1}$$

Assume U follows the Bernoulli distribution $\mathtt{Bernoulli}(p_1, \{1,2\})$. Let $p_2 = 1 - p_1$. Then the mixtures of BTL models predict the comparison's outcome by the probability,

$$\begin{aligned} \eta_{i,j}(\theta) &:= \mathbb{P}[i \succ j] \\ &= \mathbb{P}[i \succ j | U = 1]\, \mathbb{P}[U = 1] + \mathbb{P}[i \succ j | U = 2]\, \mathbb{P}[U = 2] \\ &= p_1 \frac{a_i}{a_i + a_j} + p_2 \frac{b_i}{b_i + b_j}. \end{aligned} \tag{3.2}$$

We use $\boldsymbol{\eta}(\mathbf{a}, \mathbf{b})$ to represent the $\binom{n}{2}$-dimension vector filled in these $\eta_{i,j}(\mathbf{a}, \mathbf{b})$'s for $i < j$, and omit $(\mathbf{a}, \mathbf{b})$ when clear from the context.

Our interest is on whether the parameters $\mathbf{a}$ and $\mathbf{b}$ can be recovered given η in almost always cases. Formally, we write the equation system, and abbreviate $\eta_{i,j}(\mathbf{a}, \mathbf{b})$ to $\eta_{i,j}$ for notational convenience,

$$\forall i < j,\ i, j \in [n],\ p_1 \frac{a_i}{a_i + a_j} + p_2 \frac{b_i}{b_i + b_j} = \eta_{i,j}.$$

We can further scale up $\mathbf{a}_{1:n}$ by multiplying a same constant such that we get $a_1 = 1$ and similarly manipulate to $\mathbf{b}_{1:n}$ to have $b_1 = 1$. This won't influence the values of $\eta_{i,j}$'s.

Given p_1 and p_2, to determine the latent scores of two mixtures, we try to solve the following equation system for $\mathbf{x} := x_{1:n}, \mathbf{y} := y_{1:n}$ being variables,

$$\begin{cases} x_1 = y_1 = 1, \\ p_1 \dfrac{x_i}{x_i + x_j} + p_2 \dfrac{y_i}{y_i + y_j} = \eta_{i,j}. \quad \forall i < j,\ i, j \in [n]. \end{cases} \tag{3.3}$$

Let $Q_{\mathrm{BTL}}^{2n-2} := \mathbb{R}_+^{2n-2}$ be the domain of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n})$, where $\mathbb{R}_+$ denotes the positive real numbers. Then the set of bad parameters that do not have the uniqueness property is:

$$\begin{aligned} N^{2n-2}_{BTL} = \big\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{BTL} : \exists (\mathbf{a}^{\#}_{2:n}, \mathbf{b}^{\#}_{2:n}) \in Q^{2n-2}_{BTL}, \text{s.t.} \\ \big(\mathbf{a}^{\#}_{2:n} \neq \mathbf{a}_{2:n} \vee \mathbf{b}^{\#}_{2:n} \neq \mathbf{b}_{2:n}\big) \\ \wedge \big(\boldsymbol{\eta}(\mathbf{a}^{\#}, \mathbf{b}^{\#}) = \boldsymbol{\eta}(\mathbf{a}, \mathbf{b}) \text{ for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big) \big\} . \end{aligned} \tag{3.4}$$

Then we say this mixture of BTL models given p_1 is generic identifiable if the bad set is a zero measure set (w.r.t. the Lebesgue measure).

If p_1, p_2 are not given, then we also need to regard them as variables. And we will solve the equations system for $(\mathbf{x}, \mathbf{y}, p)$,

$$\begin{cases} x_1 = y_1 = 1, \\ p\dfrac{x_i}{x_i + x_j} + (1-p)\dfrac{y_i}{y_i + y_j} = \eta_{i,j}. \quad \forall i < j, \ i, j \in [n]. \end{cases} \tag{3.5}$$

We define the domain of p_1 as $(0, 1) \subseteq \mathbb{R}$. Let $Q^{2n-1}_{BTL,p} = Q^{2n-2}_{BTL} \times (0, 1)$. And the set of bad parameters now becomes to

$$\begin{aligned} N^{2n-1}_{BTL,p} = \big\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{BTL,p} : \exists (\mathbf{a}^{\#}_{2:n}, \mathbf{b}^{\#}_{2:n}, p^{\#}) \in Q^{2n-1}_{BTL,p}, \text{s.t.} \\ \big(\mathbf{a}^{\#}_{2:n} \neq \mathbf{a}_{2:n} \vee \mathbf{b}^{\#}_{2:n} \neq \mathbf{b}_{2:n} \vee p^{\#} \neq p_1\big) \\ \wedge \big(\boldsymbol{\eta}(\mathbf{a}^{\#}, \mathbf{b}^{\#}, p^{\#}) = \boldsymbol{\eta}(\mathbf{a}, \mathbf{b}, p_1) \text{ for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big) \big\} . \end{aligned} \tag{3.6}$$

Here the Lebesgue measure is considered in a $2n-1$-dimension space.

MNL model with 3-slate

A multinomial logistic model (MNL) over n items assigns probabilities for a slate, where a slate is a subset of all items. For example, BTL model gives the likelihood of an item being selected from a slate of size 2. Mathematically, for a slate $S = \{s_1, s_2, \dots, s_k\} \subset [n]$, we have

$$\mathbb{P}[s_i \text{ is selected}] = \frac{a_{s_i}}{\sum_{j=1}^{k} a_{s_j}},$$

where a_{s_i} is the weight/score of item s_i.

For two mixtures of MNL models with k-slate, assuming the mixtures follow Bernoulli distribution with parameter p_1 and p_2, then we get

$$\mathbb{P}[s_i \text{ is selected}] = p_1 \frac{a_{s_i}}{\sum_{j=1}^{k} a_{s_j}} + p_2 \frac{b_{s_i}}{\sum_{j=1}^{k} b_{s_j}}.$$

In this example, we consider the mixtures of MNL models with 3-slate for $n \geqslant 3$. Let $a_{1:n}$, $b_{1:n}$ be the score parameters of the two mixtures. Then, we obtain that

$$\forall i < j < k, i, j, k \in [n], \quad \eta_{i,j,k} = p_1 \frac{a_i}{a_i + a_j + a_k} + p_2 \frac{b_i}{b_i + b_j + b_k}. \tag{3.7}$$

Here, we choose to scale up $a_{1:n}$ by multiplying a constant such that we get $a_1 = 1$ and similarly manipulate $b_{1:n}$ to have $b_1 = 1$. This won't influence the values of $\eta_{i,j,k}$'s.

Given p_1 and p_2, to determine the scores of two mixtures, we try to solve the following equation system for $\boldsymbol{x} := x_{1:n}, \boldsymbol{y} := y_{1:n}$ being variables and for $n \geqslant 3$,

$$\begin{cases} x_1 = y_1 = 1, \\ p_1 \dfrac{x_i}{x_i + x_j + x_k} + p_2 \dfrac{y_i}{y_i + y_j + y_k} = \eta_{i,j,k}. \quad \forall i < j < k, i, j, k \in [n], \end{cases} \tag{3.8}$$

Let $Q_{\text{MNL}}^{2n-2} := \prod_{i=1}^{2n-2} [r_i, R_i] \subseteq \mathbb{R}^{2n-2}$ be the domain of $(\boldsymbol{a}_{2:n}, \boldsymbol{b}_{2:n})$. And we assume $R_i > r_i > 0$. This interval assumption of the domain is just to diverse the choices since we assume positive numbers in the last section. Then the set of bad parameters that do not have the identifiability property is:

$$\begin{aligned} N^{2n-2}_{MNL} = \big\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{MNL} : \exists \left(\mathbf{a}^{\#}_{2:n}, \mathbf{b}^{\#}_{2:n}\right) \in Q^{2n-2}_{MNL}, \text{s.t.} \\ (\mathbf{a}^{\#}_{2:n} \neq \mathbf{a}_{2:n} \vee \mathbf{b}^{\#}_{2:n} \neq \mathbf{b}_{2:n}) \quad \wedge \\ \big(\forall i \in [n], j \in [n], k \in [n],\ \eta_{i,j,k}(\mathbf{a}^{\#}, \mathbf{b}^{\#}) = \eta_{i,j,k}(\mathbf{a}, \mathbf{b}) \\ \text{for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big)\big\}\,. \end{aligned} \tag{3.9}$$

We will later show N^{2n-2}_{MNL} has Lebesgue measure zero for the mixtures of MNL models being identifiable.

Similar to previous, when we consider p as an variable in the polynomial functions, our domain $Q^{2n-1}_{MNL,p}$ becomes to $Q^{2n-2}_{MNL,p} \times (0,1) \subseteq \mathbb{R}^{2n-1}$. And we define the set of bad parameters as

$$\begin{aligned} N^{2n-1}_{MNL,p} = \big\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{MNL,p} : \exists \left(\mathbf{a}^{\#}_{2:n}, \mathbf{b}^{\#}_{2:n}, p^{\#}\right) \in Q^{2n-1}_{MNL,p}, \text{s.t.} \\ (\mathbf{a}^{\#}_{2:n} \neq \mathbf{a}_{2:n} \vee \mathbf{b}^{\#}_{2:n} \neq \mathbf{b}_{2:n} \vee p^{\#} \neq p_1) \quad \wedge \\ \big(\forall i \in [n], j \in [n], k \in [n],\ \eta_{i,j,k}(\mathbf{a}^{\#}, \mathbf{b}^{\#}, p^{\#}) = \eta_{i,j,k}(\mathbf{a}, \mathbf{b}, p_1) \\ \text{for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big)\big\}\,. \end{aligned} \tag{3.10}$$

Plackett-Luce model

The Plackett-Luce model (Plackett (1975),Luce (1959)) is another statistical model for ranking. It assigns probabilities to all the orderings/rankings of n items, where a ordering/ranking is $i_1 \succ i_2 \succ \cdots \succ i_n$ as a permutation of $\{1, 2, \ldots, n\}$. For n items, we can have $n!$ possible orderings. In particular, we have

$$\mathbb{P}\left[i_1 \succ i_2 \succ \cdots \succ i_n\right] = \frac{a_{i_1}}{a_{i_1} + a_{i_2} + \cdots + a_{i_n}} \times \frac{a_{i_2}}{a_{i_2} + a_{i_3} + \cdots + a_{i_n}} \times \cdots \times \frac{a_{i_{n-1}}}{a_{i_{n-1}} + a_{i_n}}.$$

Here, we choose to scale up $a_{1:n}$ by multiplying a constant such that we get $a_1 + a_2 + \cdots + a_n = 1$ and similarly manipulate $b_{1:n}$ to have $b_1 + b_2 + \cdots + b_n = 1$.

Given p_1 and p_2, to determine the scores of two mixtures, we try to solve the following system of equations for $\mathbf{x}, \mathbf{y}$,

$$\begin{cases} \forall \sigma \in \mathfrak{S}_n,\ \eta_{\sigma(1),\sigma(2),\ldots,\sigma(n)} = p_1 \prod_{i=1}^{n-1} \frac{x_{\sigma(i)}}{\sum_{j=i}^{n} x_{\sigma(j)}} + p_2 \prod_{i=1}^{n-1} \frac{y_{\sigma(i)}}{\sum_{j=i}^{n} y_{\sigma(j)}}, \\ x_1 = 1 - \sum_{i=2}^{n} x_i, \\ y_1 = 1 - \sum_{i=2}^{n} y_i, \end{cases} \tag{3.11}$$

where $\mathfrak{S}_n$ is the set of all permutations of $\{1, 2, \ldots, n\}$.

Let

$$Q_{\mathrm{PL}}^{2n-2} := \Big\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in \mathbb{R}^{2n-2} : (\forall i \in \{2, \ldots, n\},\ 0 < a_i < 1) \wedge (\forall i \in \{2, \ldots, n\}, 0 < b_i < 1) \wedge \sum_{i=2}^{n} a_i < 1 \wedge \sum_{i=2}^{n} b_i < 1 \Big\}$$

be the domain of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n})$. Note Q_{PL}^{2n-2} is a polytope of a positive volume. We set $\mathbf{a}_{2:n}$ and $\mathbf{b}_{2:n}$ as the free parameters and get $a_1 = 1 - \sum_{i=2}^{n} a_i$, $b_1 = 1 - \sum_{i=2}^{n} b_i$.

Let $\boldsymbol{\eta}_{\mathfrak{S}_n}$ be all $n!$ probabilities of the $n!$ possible linear orderings/rankings among n items. Then the set of bad parameters that do achieve the identifiability property is:

$$
\begin{aligned}
N_{PL}^{2n-2} = \Big\{ & (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{PL}^{2n-2} : \exists \left(\mathbf{a}_{2:n}^{\#}, \mathbf{b}_{2:n}^{\#}\right) \in Q_{PL}^{2n-2}, \text{s.t.} \\
& (\mathbf{a}_{2:n}^{\#} \neq \mathbf{a}_{2:n} \vee \mathbf{b}_{2:n}^{\#} \neq \mathbf{b}_{2:n}) \quad \wedge \\
& \big(\forall i < j, i, j \in [n],\ \boldsymbol{\eta}_{\mathfrak{S}_n}(\mathbf{a}^{\#}, \mathbf{b}^{\#}) = \boldsymbol{\eta}_{\mathfrak{S}_n}(\mathbf{a}, \mathbf{b}) \\
& \quad \text{for } a_1^{\#} = 1 - \sum_{i=2}^{n} a_i^{\#}, b_1^{\#} = 1 - \sum_{i=2}^{n} b_i^{\#}, a_1 = 1 - \sum_{i=2}^{n} a_i, b_1 = 1 - \sum_{i=2}^{n} b_i \Big) \Big\}.
\end{aligned}
\tag{3.12}
$$

We will later show N_{PL}^{2n-2} has Lebesgue measure zero to argue the generic identifiability of the Plackett-Luce model.

Considering the space of parameters $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p)$, we then define $Q_{PL,p}^{2n-1} := Q_{PL}^{2n-2} \times (0, 1) \subseteq \mathbb{R}^{2n-1}$. The equation system becomes

$$
\begin{cases}
\forall \sigma \in \mathfrak{S}_n,\ \eta_{\sigma(1),\sigma(2),\ldots,\sigma(n)} = p \prod_{i=1}^{n-1} \frac{x_{\sigma(i)}}{\sum_{j=i}^{n} x_{\sigma(j)}} + (1-p) \prod_{i=1}^{n-1} \frac{y_{\sigma(i)}}{\sum_{j=i}^{n} y_{\sigma(j)}}, \\
x_1 = 1 - \sum_{i=2}^{n} x_i, \\
y_1 = 1 - \sum_{i=2}^{n} y_i.
\end{cases}
\tag{3.13}
$$

And the bad set in this parameter space becomes

$$
\begin{aligned}
N_{PL,p}^{2n-1} = \Big\{ & (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{PL,p}^{2n-1} : \exists \left(\mathbf{a}_{2:n}^{\#}, \mathbf{b}_{2:n}^{\#}, p^{\#}\right) \in Q_{PL,p}^{2n-1}, \text{s.t.} \\
& (\mathbf{a}_{2:n}^{\#} \neq \mathbf{a}_{2:n} \vee \mathbf{b}_{2:n}^{\#} \neq \mathbf{b}_{2:n} \vee p^{\#} \neq p_1) \quad \wedge \\
& \big(\forall i < j, i, j \in [n],\ \boldsymbol{\eta}_{\mathfrak{S}_n}(\mathbf{a}^{\#}, \mathbf{b}^{\#}, p^{\#}) = \boldsymbol{\eta}_{\mathfrak{S}_n}(\mathbf{a}, \mathbf{b}, p_1) \\
& \quad \text{for } a_1^{\#} = 1 - \sum_{i=2}^{n} a_i^{\#}, b_1^{\#} = 1 - \sum_{i=2}^{n} b_i^{\#}, a_1 = 1 - \sum_{i=2}^{n} a_i, b_1 = 1 - \sum_{i=2}^{n} b_i \Big) \Big\}.
\end{aligned}
\tag{3.14}
$$

3.3 Main Results

In this section, we state our main results on generic identifiability of polynomial systems. We will consider three cases sequentially in three subsections. The first one is the complex case, where the parameters lie in the field $\mathbb{C}$ of complex numbers. Since $\mathbb{C}$ is algebraically closed, we can directly apply many results from algebraic geometry. The rest are the real cases, where the parameters lie in the field $\mathbb{R}$ of real numbers or its subsets of positive (or infinity) Lebesgue measure. The theorems we get for the real cases are based on that of complex case followed with more careful proofs. We may use some algebraic terminology in this section and the definitions of them are listed in Appendix A.2.1 for readers' convenience.

3.3.1 Complex Case

Now we work over the field $\mathbb{C}$ of complex numbers.

Let $\mathscr{P}(\mathbf{t},\mathbf{x}) \subset \mathbb{C}[\mathbf{t}][\mathbf{x}] = \mathbb{C}[\mathbf{t},\mathbf{x}]$ be a set of polynomials in variables $\mathbf{x} = (x_1,\dots,x_n)$ with coefficients given by polynomials in parameters $\mathbf{t} = (t_1,\dots,t_m)$, i.e. each element of $\mathscr{P}(\mathbf{t},\mathbf{x})$ is of the form

$$f(\mathbf{t},\mathbf{x}) = \sum_{\mathbf{e}\in\mathbb{N}^n} f_{\mathbf{e}}(\mathbf{t})\mathbf{x}^{\mathbf{e}},$$

such that $f_{\mathbf{e}}(\mathbf{t})$ is a polynomial in $\mathbf{t}$ with coefficients in $\mathbb{C}$.

Let $\succ$ be a block order on $\mathbb{C}[\mathbf{t},\mathbf{x}]$ such that $\mathbf{x} \succ \mathbf{t}$ (e.g. lexicographic order[1]), i.e.

$$\mathbf{t}^{\mathbf{d}_1}\mathbf{x}^{\mathbf{e}_1} \succ \mathbf{t}^{\mathbf{d}_2}\mathbf{x}^{\mathbf{e}_2} \Leftrightarrow \mathbf{e}_1 \succ \mathbf{e}_2 \text{ or } (\mathbf{e}_1 = \mathbf{e}_2 \text{ and } \mathbf{t}^{\mathbf{d}_1} \succ_{\mathbf{t}} \mathbf{t}^{\mathbf{d}_2})$$

where $\succ_{\mathbf{t}}$ is an arbitrary order on $\mathbb{C}[\mathbf{t}]$. Let $I(\mathbf{t},\mathbf{x}) \subset \mathbb{C}[\mathbf{t},\mathbf{x}]$ be ideal gener-

[1]More details of lexicographic order can be found in Cox et al. (2015).

ated by $\mathscr{P}(\mathbf{t},\mathbf{x})$ and

$$G(\mathbf{t},\mathbf{x}) = \{g_1(\mathbf{t},\mathbf{x}),\dots,g_s(\mathbf{t},\mathbf{x})\}$$

be a Gröbner basis[2] of $I(\mathbf{t},\mathbf{x})$ with respect to the block order $\succ$. Let $\mathrm{Bad}(\mathbf{t}) = \{h_1(\mathbf{t}),\dots,h_r(\mathbf{t})\}$ be the set of non-zero polynomials in $\mathbf{t}$ appearing as coefficients of some $g_i(\mathbf{t},\mathbf{x})$, then

$$Z(\mathbf{t}) := \bigcup_{i=1}^{r} \text{zero-set}(h_i(\mathbf{t})) \subset \mathbb{A}^m_{\mathbb{C}} \tag{3.15}$$

is a Zariski closed proper subset, which has $\lambda^{\mathbb{C}}_m$-measure zero by Lemma A.39. In most cases this subset is efficiently computable using, e.g. Magma.

Example 3.1. *For the set of polynomials* $\{x_1x_2 - 2 = 0, tx_1x_2 + x_1 - 1 = 0\}$, *one of its Gröbner basis with respect to the block order* $\succ$ *is* $\{x_1 + (2t-1) = 0, (2t-1)x_2 + 2 = 0\}$ *and hence*

$$\mathrm{Bad}(\mathbf{t}) = \{2t-1\}, Z(\mathbf{t}) = \{1/2\} \subset \mathbb{A}^1_{\mathbb{C}}$$

An observation from this example is that in general $\mathrm{Bad}(\mathbf{t})$ *is NOT equivalent to the set of non-zero polynomials in* $\mathbf{t}$ *appearing as coefficients of some element in* $\mathscr{P}(\mathbf{t},\mathbf{x})$ *(in this example, it is* $\{t\}$*), i.e. they may even define different Zariski closed subsets in* $\mathbb{A}^m_{\mathbb{C}}$.

Assumption 3.2. $\mathscr{P}(\mathbf{a},\mathbf{x})$ *has at least* ℓ *solutions in* $\mathbb{C}$ *for all* $\mathbf{a} \in \mathbb{C}^m$.

Assumption 3.3. *There exists* $\mathbf{a}' \in \mathbb{C}^m - Z(\mathbf{t})$ *such that* $\mathscr{P}(\mathbf{a}',\mathbf{x})$ *has exactly* ℓ *solutions in* $\mathbb{C}$ *(counted with multiplicity), where* $Z(\mathbf{t}) \subset \mathbb{C}^m$ *is the* $\lambda^m_{\mathbb{C}}$*-measure zero subset determined by* (3.15).

Theorem 3.4. *Under Assumptions 3.2 and 3.3,* $\mathscr{P}(\mathbf{a},\mathbf{x})$ *has exactly* ℓ *solutions in* $\mathbb{C}$ *(counted with multiplicity) for all* $\mathbf{a} \in \mathbb{C}^m$ *but a set of* $\lambda^{\mathbb{C}}_m$*-measure zero.*

[2] More details on Gröbner basis can be found in Chapter 2 of Cox et al. (2015).

The proof is divided into two steps. The first step show that the conclusion holds "generically" with respect to the Zariski topology and the second step translates the genericness in Zariski topology in terms of Lebesgue measure.

Proof of Theorem 3.4. Geometrically one can view $\mathscr{P}(\mathbf{t}, \mathbf{x})$ as a family of affine varieties sitting in $\mathbb{A}^n_{\mathbb{C}}$,

$$\begin{array}{ccc} \mathscr{Q} & \hookrightarrow & \mathbb{A}^m_{\mathbb{C}} \times \mathbb{A}^n_{\mathbb{C}} \\ \downarrow & & \\ \mathbb{A}^m_{\mathbb{C}} & & \end{array}$$

parametrized by $\mathbb{A}^m_{\mathbb{C}}$ such that the fiber $\mathscr{Q}_{\mathbf{a}}$ over $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}}$ is (isomorphic to) the affine variety $\mathbb{V}(\mathscr{P}(\mathbf{a}, \mathbf{x})) \subset \mathbb{A}^n_{\mathbb{C}}$. Formally $\mathscr{P}(\mathbf{a}, \mathbf{x})$ is the image of $\mathscr{P}(\mathbf{t}, \mathbf{x})$ under the evaluation map

$$\varphi_{\mathbf{a}} : \mathbb{C}[\mathbf{t}, \mathbf{x}] \to \mathbb{C}[\mathbf{x}] \text{ defined by } \mathbf{t} \mapsto \mathbf{a}, \mathbf{x} \mapsto \mathbf{x}.$$

Fix an open embedding $\mathbb{A}^n_{\mathbb{C}} \hookrightarrow \mathbb{P}^n_{\mathbb{C}}$ given by $(x_1, \ldots, x_n) \mapsto [x_0 = 1 : x_1 : \ldots : x_n]$. Consider the (relative) projective closure of the above diagram with respect to $\mathbf{x}$ and this embedding

$$\begin{array}{ccc} \overline{\mathscr{Q}} & \hookrightarrow & \mathbb{A}^m_{\mathbb{C}} \times \mathbb{P}^n_{\mathbb{C}} \\ \downarrow & & \\ \mathbb{A}^m_{\mathbb{C}} & & \end{array}$$

i.e. $\overline{\mathscr{Q}}_{\mathbf{a}}|_{x_0=1} = \mathscr{Q}_{\mathbf{a}}$ for any $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}}$. By the minimality of $\overline{\mathscr{Q}_{\mathbf{a}}}$, it follows that

$$\overline{\mathscr{Q}_{\mathbf{a}}} \subset \overline{\mathscr{Q}}_{\mathbf{a}} \text{ for } \mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} \tag{3.16}$$

In general (3.16) is NOT an equality, i.e. $\overline{\mathscr{Q}}_{\mathbf{a}}$ is NOT the projective closure of $\mathscr{Q}_{\mathbf{a}}$. For example, for the single polynomial $(1 - tx)x = 0$, $\overline{\mathscr{Q}_0}$ is defined by $x = 0$ in $\mathbb{P}^1$ (and hence consists of one point $[0:1]$) while $\overline{\mathscr{Q}}_0$ is defined by $x_0 x = 0$ in $\mathbb{P}^1$ (and hence consists of two points $[0:1]$ and $[1:0]$). This is the source of many troubles. However, we could prove that generically (3.16) is an equality (in the previous example, this is an equality outside $t = 0$).

Lemma 3.5. $\overline{\mathscr{Q}_{\mathbf{a}}} = \overline{\mathscr{Q}}_{\mathbf{a}}$ *for all* $\mathbf{a} \in \mathbb{C}^m - Z(\mathbf{t})$.

Proof of Lemma 3.5. Indeed, we have

1. $\mathscr{Q}_{\mathbf{a}} = \mathbb{V}(\mathscr{P}(\mathbf{a}, \mathbf{x})) = \mathbb{V}(I(\mathbf{a}, \mathbf{x})) = \mathbb{V}(f_1(\mathbf{a}, \mathbf{x}), \dots, f_s(\mathbf{a}, \mathbf{x})) \subset \{\mathbf{a}\} \times \mathbb{A}^n_{\mathbb{C}} \cong \mathbb{A}^n_{\mathbb{C}}$. By our choice of $\mathbf{a}$, it follows that $I(\mathbf{a}, \mathbf{x}) \neq 0$ and then (Fortuna et al., 2001, Theorem 2.1) guarantees that $G(\mathbf{a}, \mathbf{x})$ is a Gröbner basis of $I(\mathbf{a}, \mathbf{x})$. Thus

$$\overline{\mathscr{Q}_{\mathbf{a}}} = \mathbb{V}({}^h(f_1(\mathbf{a}, \mathbf{x})), \dots, {}^h(f_s(\mathbf{a}, \mathbf{x}))) \subset \mathbb{P}^n_{\mathbb{C}}$$

   by (Cox et al., 2015, Chapter 8, Theorem 4 and 8), where ${}^h(f_i(\mathbf{a}, \mathbf{x}))$ is the homogenization of $f_i(\mathbf{a}, \mathbf{x})$ with respect to $\mathbf{x}$.

2. $\mathscr{Q} = \mathbb{V}(\mathscr{P}(\mathbf{t}, \mathbf{x})) = \mathbb{V}(I(\mathbf{t}, \mathbf{x})) = \mathbb{V}(f_1(\mathbf{t}, \mathbf{x}), \dots, f_s(\mathbf{t}, \mathbf{x})) \subset \mathbb{A}^m_{\mathbb{C}} \times \mathbb{A}^n_{\mathbb{C}}$ and hence

$$\overline{\mathscr{Q}} = \mathbb{V}({}^h f_1(\mathbf{t}, \mathbf{x}), \dots, {}^h f_s(\mathbf{t}, \mathbf{x})) \subset \mathbb{A}^m_{\mathbb{C}} \times \mathbb{P}^n_{\mathbb{C}}$$

   by (Cox et al., 2015, Chapter 8, Theorem 4 and 8), where ${}^h f_i(\mathbf{t}, \mathbf{x})$ is the homogenization of $f_i(\mathbf{t}, \mathbf{x})$ with respect to $\mathbf{x}$. This implies that

$$\overline{\mathscr{Q}}_{\mathbf{a}} = \mathbb{V}(({}^h f_1(\mathbf{t}, \mathbf{x}))(\mathbf{a}), \dots, ({}^h f_s(\mathbf{t}, \mathbf{x}))(\mathbf{a})) \subset \{\mathbf{a}\} \times \mathbb{P}^n_{\mathbb{C}} \cong \mathbb{P}^n_{\mathbb{C}}$$

Note that $\mathbf{a} \in \mathbb{C}^m - Z(\mathbf{t})$ is a sufficient and necessary condition for

$$({}^h f_i(\mathbf{t}, \mathbf{x}))(\mathbf{a}) = {}^h(f_i(\mathbf{a}, \mathbf{x})) \text{ for } i = 1, \dots, s$$

This implies that $\overline{\mathscr{Q}_{\mathbf{a}}} = \overline{\mathscr{Q}}_{\mathbf{a}}$ for all $\mathbf{a} \in \mathbb{C}^m - Z(\mathbf{t})$. $\square$

Step 1 We prove that $\mathscr{P}(\mathbf{a}, \mathbf{x})$ has exactly ℓ solutions in $\mathbb{C}$ (counted with multiplicity) for generic $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t})$.

More precisely, there exists a Zariski open dense subset $U_m \subset \mathbb{A}^m_{\mathbb{C}}$ such that $\mathscr{P}(\mathbf{a}, \mathbf{x})$ has exactly ℓ solutions in $\mathbb{C}$ (counted with multiplicity) for any $\mathbf{a} \in U_m$.

For any $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t})$, let $\mathbf{P}_{\mathbf{a}}(z) \in \mathbb{Q}[z]$ the Hilbert polynomial of $\overline{\mathscr{Q}}_{\mathbf{a}} \subset \mathbb{P}^n_{\mathbb{C}}$. Then by definition

$$\begin{gathered}
\mathscr{P}(\mathbf{a}, \mathbf{x}) \text{ has exactly } \ell \text{ solutions in } \mathbb{C} \text{ (counted with multiplicity)} \\
\Updownarrow \\
\dim \mathscr{Q}_{\mathbf{a}} = 0 \text{ and } \deg \mathscr{Q}_{\mathbf{a}} = \ell \\
\Updownarrow \\
\dim \overline{\mathscr{Q}_{\mathbf{a}}} = 0 \text{ and } \deg \overline{\mathscr{Q}_{\mathbf{a}}} = \ell \\
\Updownarrow \\
\dim \overline{\mathscr{Q}}_{\mathbf{a}} = 0 \text{ and } \deg \overline{\mathscr{Q}}_{\mathbf{a}} = \ell \\
\Updownarrow \\
\mathbf{P}_{\mathbf{a}}(z) = \ell
\end{gathered}$$

where the first equivalence holds since $\mathbb{C}$ is algebraically closed; the second equivalence follows from (Caniglia et al., 1991, Proposition 1.11); the third equivalence follows from Lemma 3.5 and the fourth equivalence follows from (Hartshorne, 1977, Proof of Corollary III.9.10). Hence, it is equivalent to show $\mathbf{P}_{\mathbf{a}}(z) = \ell$ for generic $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t})$. Note that Assumption 3.3 implies

$$\mathbf{P}_{\mathbf{a}'}(z) = \ell.$$

Claim 3.6. $\mathbf{P}_{\mathbf{a}'}(z)$ *is minimal in* $\mathscr{P} := \{\mathbf{P}_{\mathbf{a}}(z) : \mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t})\}$ *with respect to* $\succ$ *(in the sense of Definition A.35).*

If this is true, then by the existence of flattening stratification (see, e.g. (Nitsure, 2005, Theorem 5.13)), we know

$$S_\ell := \{\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t}) : \mathbf{P}_{\mathbf{a}}(z) = \ell\} \subset \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t}) \subset \mathbb{A}^m_{\mathbb{C}}$$

is a Zariski open dense subset. So we can simply take $U_m = S_\ell$. Thus, for **Step 1**, the only left thing to prove is Claim 3.6.

Proof of Claim 3.6. For any $\mathbf{a} \in \mathbb{A}^m_{\mathbb{C}} - Z(\mathbf{t})$, let $r := \dim \overline{\mathscr{Q}}_{\mathbf{a}}$. By (Hartshorne, 1977, Proof of Corollary III.9.10)

1. If $r > 0$, then we have

$$\mathbf{P}_{\mathbf{a}}(z) = \frac{\deg \overline{\mathscr{Q}}_{\mathbf{a}}}{r!} z^r + \text{lower degree terms},$$

   and hence $\mathbf{P}_{\mathbf{a}}(n) > \mathbf{P}_{\mathbf{a}'}(n) = \ell$ for $n \gg 0, n \in \mathbb{Z}$, i.e. $\mathbf{P}_{\mathbf{a}}(z) \succ \mathbf{P}_{\mathbf{a}'}(z)$.

2. If $r = 0$, then by Assumption 3.2, we have

$$\mathbf{P}_{\mathbf{a}}(z) = \deg \overline{\mathscr{Q}}_{\mathbf{a}} = \deg \overline{\mathscr{Q}_{\mathbf{a}}} = \deg \mathscr{Q}_{\mathbf{a}} \geqslant \ell.$$

   since for the zero-dimensional affine variety $\mathscr{Q}_{\mathbf{a}}$, its degree is equal to the number of the solutions to its defining equations in $\mathbb{C}$ (counted with multiplicity). It suggests that $\mathbf{P}_{\mathbf{a}}(n) \geqslant \mathbf{P}_{\mathbf{a}'}(n) = \ell$ for $n \in \mathbb{Z}$, i.e. $\mathbf{P}_{\mathbf{a}}(z) \succ \mathbf{P}_{\mathbf{a}'}(z)$.

Together we finish the proof of Claim 3.6 and hence **Step 1**. □

Step 2 We argue that the complement of U_m in $\mathbb{C}^m$ has $\lambda^{\mathbb{C}}_m$-measure zero, i.e. $\lambda^{\mathbb{C}}_m(\mathbb{C}^m \setminus U_m) = 0$.

Note that $Z := \mathbb{A}^m_{\mathbb{C}} \setminus U_m \subsetneq \mathbb{A}^m_{\mathbb{C}}$ is a Zariski closed proper subset. Then we are done by Lemma A.39.

Finally, these two steps conclude the proof. □

Remark 3.7. *As we have seen before, $\mathscr{P}(\mathfrak{a}, x)$ has exactly ℓ solutions in $\mathbb{C}$ (counted with multiplicity) if and only if*

$$\dim \mathscr{Q}_{\mathfrak{a}} = 0 \text{ and } \deg \mathscr{Q}_{\mathfrak{a}} = \ell.$$

If the coefficients appearing in $\mathscr{P}(\mathfrak{a}, x)$ are all in $\mathbb{Q}$, this dimension and degree condition can be checked with the help of Magma using the commands

$$\texttt{Dimension}(\mathscr{Q}_{\mathfrak{a}}) \text{ and } \texttt{Degree}(\mathscr{Q}_{\mathfrak{a}})$$

This is because Magma works well over the field $\mathbb{Q}$ of rational numbers. More details will be shown in the next section.

Remark 3.8. *Our condition $\mathfrak{a} \in \mathbb{C}^m - Z(\mathbf{t})$ is only a sufficient condition for Lemma 3.5 (and hence Theorem 3.4) to hold, partly due to the usage of (Fortuna et al., 2001, Theorem 2.1). Nevertheless it is a measure-zero condition. However, we want to emphasis that it is always necessary to exclude some "bad" parameter. As we already see that for the example $(1 - tx)x = 0$, if $t = 0$ then we have a unique solution in $\mathbb{C}$, but this does not hold generically.*

3.3.2 Real Case

Now we consider the real case. Our goal is to obtain a variant of Theorem 3.4 in the real case. The arguments in complex case cannot be applied directly because $\mathbb{R}$ is not algebraically closed.

The starting point is the lemma below.

Lemma 3.9 ((Okamoto, 1973), Lemma)**.** *For any non-zero polynomial $f \in \mathbb{C}[x_1, \ldots, x_m]$, we have*

$$\mathbb{V}(f) \cap \mathbb{R}^m \subset \mathbb{R}^m$$

is of λ_m-measure zero.

Proof of Lemma 3.9. This is proved by induction on m.

If $m = 1$, then $\mathbb{V}(f) \subset \mathbb{C}$ is a finite set and so is $\mathbb{V}(f) \cap \mathbb{R}$. Trivially $\mathbb{V}(f) \cap \mathbb{R} \subset \mathbb{R}$ is of λ_1-measure zero. Suppose the conclusion holds for $m-1$ and we need to show

$$N_m := \mathbb{V}(f) \cap \mathbb{R}^m = \{(x_1, \ldots, x_m) \in \mathbb{R}^m : f(x_1, \ldots, x_m) = 0\} \subset \mathbb{R}^m$$

is of λ_m-measure zero. Consider the cross-section of N_m at the point $(x_1, \ldots, x_{m-1})$,

$$C(x_1, \ldots, x_{m-1}) := \{x_m \in \mathbb{R} : f(x_1, \ldots, x_{m-1}, x_m) = 0\} \subset \mathbb{R}.$$

It is a finite set since f is non-trivial. Let $d > 0$ be the degree of f with respect to the variable x_m. Write

$$f(x_1, \ldots, x_m) = \sum_{i=0}^{d} f_i(x_1, \ldots, x_{m-1}) x_m^i,$$

and

$$\begin{aligned} N_{m-1} &:= \{(x_1, \ldots, x_{m-1}) \in \mathbb{R}^{m-1} : f_i(x_1, \ldots, x_{m-1}) = 0 \text{ for } i = 0, \ldots, d\} \\ &= \bigcap_{i=0}^{d} \mathbb{V}(f_i) \cap \mathbb{R}^{m-1} \subset \mathbb{R}^{m-1}. \end{aligned}$$

By induction hypothesis, at least one of $\mathbb{V}(f_i) \cap \mathbb{R}^{m-1} \subset \mathbb{R}^{m-1}$ is of λ_{m-1}-

measure zero, then $\lambda_{m-1}(N_{m-1}) = 0$. Now it follows that

$$\begin{aligned}
\lambda_m(N_m) &= \int_{\mathbb{R}^m} \mathbf{1}_{N_m}\, d\lambda_m(x_1, \dots, x_m) \\
&= \int_{\mathbb{R}^{m-1}} \left(\int_{\mathbb{R}} \mathbf{1}_{C(x_1,\dots,x_{m-1})} d\lambda_1(x_m) \right) d\lambda_{m-1}(x_1, \dots, x_{m-1}) \\
&= \int_{\mathbb{R}^{m-1}} \lambda_1(C(x_1, \dots, x_{m-1}))\, d\lambda_{m-1}(x_1, \dots, x_{m-1}) \\
&= \int_{N_{m-1}} \lambda_1(C(x_1, \dots, x_{m-1}))\, d\lambda_{m-1}(x_1, \dots, x_{m-1}) \\
&\quad + \int_{\mathbb{R}^{m-1} \setminus N_{m-1}} \lambda_1(C(x_1, \dots, x_{m-1}))\, d\lambda_{m-1}(x_1, \dots, x_{m-1}),
\end{aligned}$$

where the second equality follows directly from Fubini's theorem (see, e.g. (Zorich, 2016, Chapter 11.4.1, Theorem)). The first term vanishes because of $\lambda_{m-1}(N_{m-1}) = 0$. The second vanishes since for any $(x_1, \dots, x_{m-1}) \notin N_{m-1}$, the set $C(x_1, \dots, x_{m-1})$ is a finite set and hence $\lambda_1(C(x_1, \dots, x_{m-1})) = 0$. Thus $\lambda_m(N_m) = 0$. □

Corollary 3.10. *The intersection of any Zariski closed proper subset of $\mathbb{A}^m_{\mathbb{C}}$ with $\mathbb{R}^m$ is of λ_m-measure zero.*

Proof of Corollary 3.10. Let $Z \subsetneq \mathbb{A}^m_{\mathbb{C}}$ be a Zariski closed proper subset, then we can write

$$Z = \mathbb{V}(f_1, \dots, f_r) = \bigcap_{i=1}^{r} \mathbb{V}(f_i)$$

for some polynomials $f_i \in \mathbb{C}[x_1, \dots, x_m]$. Since Z is proper, at least one of f_i's is non-zero, say f_1. By Lemma 3.9, $\mathbb{V}(f_1) \cap \mathbb{R}^m \subset \mathbb{R}^m$ is of λ_m-measure zero. As a subset of a set of λ_m-measure zero, Z is again of λ_m-measure zero (since the Lebesgue measure on $\mathbb{R}^m$ is complete). □

Using Corollary 3.10, we obtain a variant of Theorem 3.4 in real case.

Theorem 3.11. *Under Assumptions 3.2 and 3.3, $\mathscr{P}(\mathbf{a}, \mathbf{x})$ has exactly ℓ solutions in $\mathbb{C}$ (counted with multiplicity) for all $\mathbf{a} \in \mathbb{R}^m$ but a set of λ_m-measure zero.*

Proof of Theorem 3.11. We use the same notions in the proof of Theorem 3.4. It suffices to show

$$Z \cap \mathbb{R}^m \subset \mathbb{R}^m$$

is of λ_m-measure zero. This is true by Corollary 3.10. □

3.3.3 General Case

In general, it is not hard to see from the proof that Lemma 3.9 (and hence Corollary 3.10) still holds for any subset of $\mathbb{R}^m$ of positive (or infinity) Lebesgue measure. As a result, we obtain a variant of Theorem 3.4 in this generality.

Theorem 3.12. *Let $K \subset \mathbb{R}^m$ be a subset of positive (or infinity) λ_m-measure. Under Assumptions 3.2 and 3.3, $\mathscr{P}(\mathbf{a}, \mathbf{x})$ has exactly ℓ solutions in $\mathbb{C}$ (counted with multiplicity) for all $\mathbf{a} \in K$ but a set of λ_m-measure zero.*

Remark 3.13. *It is worth mentioning that Assumption 3.2 and 3.3 are independent of the specific parameter space K in Theorem 3.12. In particular, the chosen parameter $\mathbf{a}'$ in Assumption 3.3 need not to be in our parameter space K. This gives us a lot of freedom in practical application.*

Moreover, it is also clear from the proof that in Theorem 3.12, we could replace Assumption 3.2 by the following slightly weaker statement:

Assumption 3.2′. *$\mathscr{P}(\mathbf{a}, \mathbf{x})$ has at least ℓ solutions in $\mathbb{C}$ for all $\mathbf{a} \in K - Z_1$, where $Z_1 \subset \mathbb{A}^m_{\mathbb{C}}$ is a Zariski closed proper subset.*

Indeed, in the proof of Theorem 3.4, we can replace the parameter space $\mathbb{A}^m_{\mathbb{C}}$ by $\mathbb{A}^m_{\mathbb{C}} - Z_1$ since the existence of flattening stratification (see, e.g. (Nitsure, 2005, Theorem 5.13)) still holds in this case and Lemma A.39 implies that Z_1 is of $\lambda^{\mathbb{C}}_m$-measure zero. However, in general we could not replace "$Z_1 \subset \mathbb{A}^m_{\mathbb{C}}$ is a Zariski closed proper subset" by the more friendly phrase "$Z_1 \subset \mathbb{C}^m$ is a subset of $\lambda^{\mathbb{C}}_m$-measure zero" in Assumption 3.2′. This is because we need to apply the *existence of flattening stratification (see, e.g. (Nitsure, 2005, Theorem 5.13)) to* $\mathbb{A}^m_{\mathbb{C}} - Z_1$*, such subset* $Z_1 \subset \mathbb{C}^m$ *should, at least, be contained in some Zariski closed proper subset. But this is already false in one-dimensional case, e.g.* $\mathbb{N} \subset \mathbb{C}$ *is such a counterexample.*

3.4 Consequences for Specific Models

In the previous section, we provided a number of general results on the number of solutions to polynomial systems. In this section, we develop some concrete consequences of this general theory for the three specific mixtures ranking classes previously introduced in Section 3.2.

3.4.1 Mixtures of BTL

In this section, we study the mixtures of BTL models. Our result affirms that the two mixtures with pairwise comparisons has a unique solution generically, which was an unsolved problem in the literature. The first subsection is about the generic identifiability when p_1 and p_2 is given and the second subsection is about the generic identifiability when p is to be estimated.

Parameter space of $(a_{2:n}, b_{2:n})$

Now, we consider the case when p_1 and p_2 are given. Then the parameter is $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n})$, which can be freely chosen in the domain Q_{BTL}^{2n-2}. In particular, we consider $p_1 = 0.7$ in our particular example. To check cases with different p_1, we can execute the same procedures.

Recall that we know the $n(n-1)/2$ elements in $\mathbb{R}$,

$$\eta_{i,j} := p_1 \frac{a_i}{a_i + a_j} + p_2 \frac{b_i}{b_i + b_j}. \quad \forall i < j \in [n].$$

The estimation/ranking problem is to solve the equation system (3.3) in variables (x, y) and with coefficients (a, b) (because p_1, p_2 are given and η_{ij}'s depend on (a, b)).

Notice that (3.3) has at least one solution in $\mathbb{C}$, i.e. $(x, y) = (a, b)$ coming from the initial data. Our goal is to show it is also the unique solution of (3.3) in $\mathbb{C}$ for generic $(a_{2:n}, b_{2:n}) \in Q_{BTL}^{2n-2}$. To be precise, we write the proposition below.

Proposition 3.14. *If* $n \geqslant 5$ *and* $p_1 = 0.7, p_2 = 0.3$, *then* (3.3) *has a unique solution in* $\mathbb{C}$ *(counted with multiplicity) for all* $(a_{2:n}, b_{2:n}) \in Q_{BTL}^{2n-2}$ *but a set of* λ_{2n-2}*-measure zero, given by* $(x, y) = (a, b)$.

Proof of Proposition 3.14. This is proved by induction on n.

Case $n = 5$. In this case, we can expand (3.3) such that its coefficients are given by polynomials in (a, b):

$$\begin{cases} x_1 = 1, y_1 = 1, \\ (p_1 a_j(b_i + b_j) + p_2 b_j(a_i + a_j))x_i y_i - (p_1 a_i(b_i + b_j) + p_2 b_i(a_i + a_j))x_j y_j + \\ (p_1 a_j(b_i + b_j) - p_2 b_i(a_i + a_j))x_i y_j - (p_1 a_i(b_i + b_j) - p_2 b_j(a_i + a_j))x_j y_i = 0, \\ \qquad \forall i < j \in [5] \end{cases} \tag{3.17}$$

Note this this is NOT a faithful transformation of (3.3) (but can only increase the number of solutions). To proceed, we need to determine $Z(a_{2:5}, b_{2:5})$ introduced in Lemma 3.5. This can be done by Magma.

Listing 3.1: Grönber basis of BTL models with fixed p

```
P<x1,x2,x3,x4,x5,y1,y2,y3,y4,y5,
  a1,a2,a3,a4,a5,b1,b2,b3,b4,b5>
:=FreeAlgebra(Rationals(),20,"lex");

I:=ideal<P|x1-1,y1-1,a1-1,b1-1,
(a1*b1+3/10*a2*b1+7/10*a1*b2)*(x1+x2)*(y1+y2)+
(a1+a2)*(b1+b2)*(-3/10*(x1+x2)*y1-7/10*x1*(y1+y2)),
(a1*b1+3/10*a3*b1+7/10*a1*b3)*(x1+x3)*(y1+y3)+
(a1+a3)*(b1+b3)*(-3/10*(x1+x3)*y1-7/10*x1*(y1+y3)),
```

```
(a1*b1+3/10*a4*b1+7/10*a1*b4)*(x1+x4)*(y1+y4)+
(a1+a4)*(b1+b4)*(-3/10*(x1+x4)*y1-7/10*x1*(y1+y4)),
(a1*b1+3/10*a5*b1+7/10*a1*b5)*(x1+x5)*(y1+y5)+
(a1+a5)*(b1+b5)*(-3/10*(x1+x5)*y1-7/10*x1*(y1+y5)),
(a2*b2+3/10*a3*b2+7/10*a2*b3)*(x2+x3)*(y2+y3)+
(a2+a3)*(b2+b3)*(-3/10*(x2+x3)*y2-7/10*x2*(y2+y3)),
(a2*b2+3/10*a4*b2+7/10*a2*b4)*(x2+x4)*(y2+y4)+
(a2+a4)*(b2+b4)*(-3/10*(x2+x4)*y2-7/10*x2*(y2+y4)),
(a2*b2+3/10*a5*b2+7/10*a2*b5)*(x2+x5)*(y2+y5)+
(a2+a5)*(b2+b5)*(-3/10*(x2+x5)*y2-7/10*x2*(y2+y5)),
(a3*b3+3/10*a4*b3+7/10*a3*b4)*(x3+x4)*(y3+y4)+
(a3+a4)*(b3+b4)*(-3/10*(x3+x4)*y3-7/10*x3*(y3+y4)),
(a3*b3+3/10*a5*b3+7/10*a3*b5)*(x3+x5)*(y3+y5)+
(a3+a5)*(b3+b5)*(-3/10*(x3+x5)*y3-7/10*x3*(y3+y5)),
(a4*b4+3/10*a5*b4+7/10*a4*b5)*(x4+x5)*(y4+y5)+
(a4+a5)*(b4+b5)*(-3/10*(x4+x5)*y4-7/10*x4*(y4+y5))>;

GroebnerBasis(I);
```

This gives

$$\mathrm{Bad}(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) = \{3a_ib_i - 4a_jb_i - 7a_jb_j,\ 10a_ib_i + 7a_ib_j + 3a_jb_i : \forall i \neq j \in [5]\} \tag{3.18}$$

Based on Theorem 3.12, we claim by checking Assumption 3.2′ and 3.3 for (3.17) that (3.17) has exactly 3 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) \in Q^8_{BTL}$ but a set of λ_8-measure zero.

Assumption 3.2′: This is clear since

$$\begin{gathered}(\mathbf{x}_{1:5}, \mathbf{y}_{1:5}) = (\mathbf{a}_{1:5}, \mathbf{b}_{1:5}),\\ \left(a_1, 0, 0, 0, 0; b_1, \left(\frac{1-\eta_{1j}}{\eta_{1j}-p_1}\right)_{j=2,\dots,5}\right), \left(a_1, \left(\frac{1-\eta_{1j}}{\eta_{1j}-p_2}\right)_{j=2,\dots,5}; b_1, 0, 0, 0, 0\right)\end{gathered} \tag{3.19}$$

are three (distinct) solutions of (3.17) for all $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) \in Q^8_{BTL} - Z$, where

$Z \subset \mathbb{A}^8_{\mathbb{C}}$ is the Zariski closed proper subset defined by

$$\cup_{j=2}^{5}\text{zero-set}(\eta_{1j}(\mathbf{a},\mathbf{b})-p_1)\bigcup\cup_{j=2}^{5}\text{zero-set}(\eta_{1j}(\mathbf{a},\mathbf{b})-p_2)\bigcup\text{zero-set}(1-\eta_{12}(\mathbf{a},\mathbf{b}))\subset\mathbb{A}^8_{\mathbb{C}}$$

Assumption 3.3: Choose $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5}) = (1, 2, 3, 4, 5; 1, 8, 9, 3, 2)$. It is routine to check that $(\mathbf{a}'_{2:5}, \mathbf{b}'_{2:5}) \in \mathbb{C}^8 - Z(\mathbf{a}_{2:5}, \mathbf{b}_{2:5})$ using (3.18). Since the associated equation system (3.17) is of $\mathbb{Q}$-coefficient, we can use Magma (see (Bosma et al., 1997) and Remark 3.7) to check whether it has exactly 3 solutions in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5})$.

Listing 3.2: Dimension and degree computations of BTL models with fixed p

```
a:=[1,2,3,4,5];
b:=[1,8,9,3,2];
p1:=7/10;

k:=Rationals();
A<x1,x2,x3,x4,x5,y1,y2,y3,y4,y5>:=AffineSpace(k,10);
P:=Scheme(A,[
x1-1,
y1-1,
(a[1]*b[1]+p1*a[2]*b[1]+(1-p1)*a[1]*b[2])*(x1+x2)*(y1+y2)+
(a[1]+a[2])*(b[1]+b[2])*(-p1*(x1+x2)*y1-(1-p1)*x1*(y1+y2)),
(a[1]*b[1]+p1*a[3]*b[1]+(1-p1)*a[1]*b[3])*(x1+x3)*(y1+y3)+
(a[1]+a[3])*(b[1]+b[3])*(-p1*(x1+x3)*y1-(1-p1)*x1*(y1+y3)),
(a[1]*b[1]+p1*a[4]*b[1]+(1-p1)*a[1]*b[4])*(x1+x4)*(y1+y4)+
(a[1]+a[4])*(b[1]+b[4])*(-p1*(x1+x4)*y1-(1-p1)*x1*(y1+y4)),
(a[1]*b[1]+p1*a[5]*b[1]+(1-p1)*a[1]*b[5])*(x1+x5)*(y1+y5)+
(a[1]+a[5])*(b[1]+b[5])*(-p1*(x1+x5)*y1-(1-p1)*x1*(y1+y5)),
(a[2]*b[2]+p1*a[3]*b[2]+(1-p1)*a[2]*b[3])*(x2+x3)*(y2+y3)+
(a[2]+a[3])*(b[2]+b[3])*(-p1*(x2+x3)*y2-(1-p1)*x2*(y2+y3)),
(a[2]*b[2]+p1*a[4]*b[2]+(1-p1)*a[2]*b[4])*(x2+x4)*(y2+y4)+
(a[2]+a[4])*(b[2]+b[4])*(-p1*(x2+x4)*y2-(1-p1)*x2*(y2+y4)),
(a[2]*b[2]+p1*a[5]*b[2]+(1-p1)*a[2]*b[5])*(x2+x5)*(y2+y5)+
(a[2]+a[5])*(b[2]+b[5])*(-p1*(x2+x5)*y2-(1-p1)*x2*(y2+y5)),
(a[3]*b[3]+p1*a[4]*b[3]+(1-p1)*a[3]*b[4])*(x3+x4)*(y3+y4)+
(a[3]+a[4])*(b[3]+b[4])*(-p1*(x3+x4)*y3-(1-p1)*x3*(y3+y4)),
(a[3]*b[3]+p1*a[5]*b[3]+(1-p1)*a[3]*b[5])*(x3+x5)*(y3+y5)+
(a[3]+a[5])*(b[3]+b[5])*(-p1*(x3+x5)*y3-(1-p1)*x3*(y3+y5)),
(a[4]*b[4]+p1*a[5]*b[4]+(1-p1)*a[4]*b[5])*(x4+x5)*(y4+y5)+
(a[4]+a[5])*(b[4]+b[5])*(-p1*(x4+x5)*y4-(1-p1)*x4*(y4+y5))
```

```
]);

Dimension(P);
-> 0

Degree(P);
-> 3
```

From Listing 3.2, `Dimension(P)=0` and `Degree(P)=3` means (3.17) has exactly 3 solutions in $\mathbb{C}$ (counted with multiplicity) for this choice of $(\mathbf{a}'_{2:5}, \mathbf{b}'_{2:5})$.

Thus by Theorem 3.12, we prove that (3.17) has exactly 3 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) \in Q^8_{BTL}$ but a set V_5 of λ_8-measure zero. Replacing V_5 by $V_5 \cup Z$ if necessary (since $Z \subset \mathbb{C}^8$ is of λ_8-measure zero by Lemma A.39), we may assume $Z \subset V_5$. This implies that for any $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) \in Q^8_{BTL} - V_5$, the 3 solutions of (3.17) are necessarily given by (3.19), of which the first one is always a solution of (3.3) while the last two are not allowed by (3.3). Altogether, this shows that (3.3) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}) \in Q^8_{BTL}$ but a set V_5 of λ_8-measure zero.

Case $n \geqslant 5$. Suppose the conclusion holds for $n-1$. We need to prove that

$$\begin{cases} x_1 = y_1 = 1 \\ p_1 \dfrac{x_i}{x_i + x_j} + p_2 \dfrac{y_i}{y_i + y_j} = \eta_{ij}, \ \forall i < j \in [n] \end{cases} \tag{3.20}$$

has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{BTL}$ but a set V_n of λ_{2n-2}-measure zero. We begin with splitting (3.20) into two parts:

$$\begin{cases} x_1 = y_1 = 1 \\ p_1 \dfrac{x_i}{x_i + x_j} + p_2 \dfrac{y_i}{y_i + y_j} = \eta_{ij},\ \forall i < j \in [n-1] \end{cases} \tag{3.21}$$

and

$$p_1 \frac{x_i}{x_i + x_n} + p_2 \frac{y_i}{y_i + y_n} = \eta_{in},\ \forall i = 1, \dots, n-1. \tag{3.22}$$

By induction hypothesis, there exists a λ_{2n-4}-measure zero subset $V_{n-1} \subset \mathbb{C}^{2n-4}$ such that (3.21) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for any $(\mathbf{a}_{2:n-1}, \mathbf{b}_{2:n-1}) \in Q_{BTL}^{2n-4} - V_{n-1}$, given by

$$(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}) = (\mathbf{a}_{1:n-1}, \mathbf{b}_{1:n-1}).$$

Plugging this solution into (3.22) and simplifying, we obtain

$$(\eta_{in} - 1)a_i b_i + (\eta_{in} - p_1)a_i y_n + (\eta_{in} - p_2)b_i x_n + \eta_{in} x_n y_n = 0, \quad \forall i = 1, \dots, n-1. \tag{3.23}$$

Since $\eta_{1n} \neq 0$, by the case of $i = 1$ in (3.23), we have

$$x_n y_n = \frac{(\eta_{1n} - 1)a_1 b_1 + (\eta_{1n} - p_1)a_1 y_n + (\eta_{1n} - p_2)b_1 x_n}{\eta_{1n}}.$$

Plugging it into the cases of $i = 2, 3$ in (3.23), we further have a system of linear equations in variables (x_n, y_n)

$$\begin{cases} (\eta_{1n}(\eta_{2n} - p_2)b_2 - \eta_{2n}(\eta_{1n} - p_2))x_n + (\eta_{1n}(\eta_{2n} - p_1)a_2 - \eta_{2n}(\eta_{1n} - p_1))y_n \\ = c_2, \\ (\eta_{1n}(\eta_{3n} - p_2)b_3 - \eta_{3n}(\eta_{1n} - p_2))x_n + (\eta_{1n}(\eta_{3n} - p_1)a_3 - \eta_{3n}(\eta_{1n} - p_1))y_n \\ = c_3. \end{cases} \tag{3.24}$$

where

$$c_i := \frac{\eta_{in}}{\eta_{1n}}(\eta_{1n} - 1) - (\eta_{in} - 1)x_i y_i.$$

Denote the coefficients matrix of (3.24) by

$$A := \begin{pmatrix} \eta_{1n}(\eta_{2n} - p_2)b_2 - \eta_{2n}(\eta_{1n} - p_2) & \eta_{1n}(\eta_{2n} - p_1)a_2 - \eta_{2n}(\eta_{1n} - p_1) \\ \eta_{1n}(\eta_{3n} - p_2)b_3 - \eta_{3n}(\eta_{1n} - p_2) & \eta_{1n}(\eta_{3n} - p_1)a_3 - \eta_{3n}(\eta_{1n} - p_1) \end{pmatrix}$$

and define $W_n := \{(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in \mathbb{C}^{2n-2} : \det(A) = 0\}$ and $V_n := (V_{n-1} \times \mathbb{C}^2) \cup W_n \subset \mathbb{C}^{2n-2}$. Note that V_n is also a λ_{2n-2}-measure zero subset. We finish the proof by claiming that (3.20) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{BTL}^{2n-2}$ but the set V_n. Indeed, for any $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{BTL}^{2n-2} - V_n$, we have

1. since $(\mathbf{a}_{2:n-1}, \mathbf{b}_{2:n-1}) \notin V_{n-1}$, (3.21) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by

$$(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}) = (\mathbf{a}_{1:n-1}, \mathbf{b}_{1:n-1}).$$

2. since $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \notin W_n$, (3.24) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by

$$(x_n, y_n) = (a_n, b_n).$$

This shows that (3.20) has a unique solution in $\mathbb{C}$ (counted with multiplicity) since $(x, y) = (a, b)$ is always a solution of it. □

Remark 3.15. *For $p_1 = 0.5$, one can argue similarly that generically the equation system (3.3) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity), given by $(x, y) = (a, b)$ and (b, a). The procedure is very similar to the case where $p_1 = 0.7$ so we omit the details here. For other choice of $p_1 \neq 0.5$, one can try the same procedure to check the generic identifiability of (3.3) for any specific rational value for p_1. As we have tried, for the choice of $p_1 \in \{0.01, 0.02, \dots, 0.99, 1\} \backslash \{0.5\}$, we could always obtain the generic identifiability. For $p_1 = 0.5$, we obtain the generic identifiability up to reordering, i.e., exactly two solutions generically.*

Remark 3.16. *We discuss about the tightness of n. For $n \leqslant 3$, since the number of equations is less than the number of variables, in general we won't expect the identifiability. When $n = 4$, the number of equations equals to the number of equations, which we are not sure whether the generic identifiability holds or not in this case.*

Parameter space of $(a_{2:n}, b_{2:n}, p_1)$

Instead of fixing p_1 (and hence p_2), we could also view p_1 as a parameter. In this case, the parameter space becomes $(a_{2:n}, b_{2:n}, p_1) \in Q^{2n-1}_{BTL,p}$ and we consider the variant (3.5) of (3.3), which is an equation system having (x, y, p) as variables and $(a_{2:n}, b_{2:n}, p_1)$ as coefficients.

Proposition 3.17. *If $n \geqslant 5$, then (3.5) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(a_{2:n}, b_{2:n}, p_1) \in Q^{2n-1}_{BTL,p}$ but a set of λ_{2n-1}-measure zero, given by $(x, y, p) = (a, b, p_1)$ and $(x, y, p) = (b, a, 1 - p_1)$.*

Proof of Proposition 3.17. Same as before, we first translate (3.5) into (equivalent) polynomial one such that its coefficients are given by polynomials in $(\mathbf{a}, \mathbf{b}, p_1)$

$$\begin{cases} x_1 = y_1 = 1, \\ (c_{ij} - d_{ij})x_i y_i + (c_{ij} - p d_{ij})x_i y_j + (c_{ij} - (1-p)d_{ij})x_j y_i + \quad c_{ij} x_j y_j = 0, \\ \hfill \forall i < j \in [n] \\ t_{ij}(x_i + x_j) = 1, \; h_{ij}(y_i + y_j) = 1, \hfill \forall i < j \in [n] \end{cases} \tag{3.25}$$

where

$$c_{ij} := p_1 a_i (b_i + b_j) + (1 - p_1) b_i (a_i + a_j) \text{ and } d_{ij} := (a_i + a_j)(b_i + b_j).$$

This is an equation system in variables $(\mathbf{x}, \mathbf{y}, p, \mathbf{t}, \mathbf{h})$ and with coefficients $(\mathbf{a}, \mathbf{b}, p_1)$. Indeed, as a result of introducing new variables and equations to (3.5) to prevent its denominators from zeros, it follows that for any $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1)$ the equation system (3.5) is equivalent to the equation system (3.25). In particular, (3.5) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) if and only if (3.25) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity).

Note that (3.25) always has 2 (distinct) solutions in $\mathbb{C}$

$$\begin{aligned} (\mathbf{x}, \mathbf{y}, p, \mathbf{t}, \mathbf{h}) = & \left(\mathbf{a}, \mathbf{b}, p_1, \left(\frac{1}{a_i + a_j} \right)_{i,j}, \left(\frac{1}{b_i + b_j} \right)_{i,j} \right) \\ \text{or } & \left(\mathbf{b}, \mathbf{a}, 1 - p_1, \left(\frac{1}{b_i + b_j} \right)_{i,j}, \left(\frac{1}{a_i + a_j} \right)_{i,j} \right) \end{aligned}$$

for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{BTL,p} - Z_n$, where $Z_n \subset \mathbb{A}^{2n-1}_{\mathbb{C}}$ is the Zariski closed proper subset (since it is defined by a single polynomial $p_1 - 0.5 = 0$) defined by

$$Z_n := \left\{ (\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in \mathbb{A}^{2n-1}_{\mathbb{C}} : p_1 - 0.5 = 0 \right\} \subset \mathbb{A}^{2n-1}_{\mathbb{C}}$$

As before, this proposition can be proved by induction on n.

Case $n = 5$. In the case where $n = 5$, we consider the following subset of equations in (3.25).

$$\begin{cases} x_1 = y_1 = 1, \\ (c_{ij} - d_{ij})x_i y_i + (c_{ij} - p d_{ij})x_i y_j + (c_{ij} - (1-p)d_{ij})x_j y_i + c_{ij} x_j y_j = 0, \\ \hfill \forall i < j \in [5] \\ t_{23}(x_2 + x_3) = 1,\ t_{15}(x_1 + x_5) = 1,\ h_{23}(y_2 + y_3) = 1. \end{cases} \tag{3.26}$$

It is clear that (3.25) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) if (3.26) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity). So it suffices to consider (3.26). Based on Theorem 3.12, it suffices to check Assumption 3.2′ and Assumption 3.3 for (3.26).

For Assumption 3.2′: as we see before, (3.26) has at least 2 solutions in $\mathbb{C}$ for all $(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}, p_1) \in Q^9_{BTL,p} - Z_5$.

For Assumption 3.3: we need to find a specific $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5}, p'_1)$ such that

1. $(\mathbf{a}'_{2:5}, \mathbf{b}'_{2:5}, p'_1) \in \mathbb{C}^9 - Z(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}, p_1)$.
2. the associated equation system (3.26) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity).

Both of them can be checked by Magma. We choose $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5}, p'_1) = (1, 2, 3, 4, 5; 1, 8, 9, 3, 2; 0.7)$.

For the first one, we need to determine $\text{Bad}(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}, p_1)$. This can be done via running Listing 3.3.

Listing 3.3: Grönber basis of BTL models with variable p

```
P<x1,x2,x3,x4,x5,y1,y2,y3,y4,y5,p,t23,h15,h23,
  a1,a2,a3,a4,a5,b1,b2,b3,b4,b5,p1>
:=FreeAlgebra(Rationals(),25,"lex");

I:=ideal<P|x1-a1,y1-b1,a1-1,b1-1,
```

```
(p1*a1*(b1+b2)+(1-p1)*b1*(a1+a2)-(a1+a2)*(b1+b2))*x1*y1+
(p1*a1*(b1+b2)+(1-p1)*b1*(a1+a2)-p*(a1+a2)*(b1+b2))*x1*y2+
(p1*a1*(b1+b2)+(1-p1)*b1*(a1+a2)-(1-p)*(a1+a2)*(b1+b2))*x2*y1
-(p1*a1*(b1+b2)+(1-p1)*b1*(a1+a2))*x2*y2,
(p1*a1*(b1+b3)+(1-p1)*b1*(a1+a3)-(a1+a3)*(b1+b3))*x1*y1+
(p1*a1*(b1+b3)+(1-p1)*b1*(a1+a3)-p*(a1+a3)*(b1+b3))*x1*y3+
(p1*a1*(b1+b3)+(1-p1)*b1*(a1+a3)-(1-p)*(a1+a3)*(b1+b3))*x3*y1
-(p1*a1*(b1+b3)+(1-p1)*b1*(a1+a3))*x3*y3,
(p1*a1*(b1+b4)+(1-p1)*b1*(a1+a4)-(a1+a4)*(b1+b4))*x1*y1+
(p1*a1*(b1+b4)+(1-p1)*b1*(a1+a4)-p*(a1+a4)*(b1+b4))*x1*y4+
(p1*a1*(b1+b4)+(1-p1)*b1*(a1+a4)-(1-p)*(a1+a4)*(b1+b4))*x4*y1
-(p1*a1*(b1+b4)+(1-p1)*b1*(a1+a4))*x4*y4,
(p1*a1*(b1+b5)+(1-p1)*b1*(a1+a5)-(a1+a5)*(b1+b5))*x1*y1+
(p1*a1*(b1+b5)+(1-p1)*b1*(a1+a5)-p*(a1+a5)*(b1+b5))*x1*y5+
(p1*a1*(b1+b5)+(1-p1)*b1*(a1+a5)-(1-p)*(a1+a5)*(b1+b5))*x5*y1
-(p1*a1*(b1+b5)+(1-p1)*b1*(a1+a5))*x5*y5,
(p1*a2*(b2+b3)+(1-p1)*b2*(a2+a3)-(a2+a3)*(b2+b3))*x2*y2+
(p1*a2*(b2+b3)+(1-p1)*b2*(a2+a3)-p*(a2+a3)*(b2+b3))*x2*y3+
(p1*a2*(b2+b3)+(1-p1)*b2*(a2+a3)-(1-p)*(a2+a3)*(b2+b3))*x3*y2
-(p1*a2*(b2+b3)+(1-p1)*b2*(a2+a3))*x3*y3,
(p1*a2*(b2+b4)+(1-p1)*b2*(a2+a4)-(a2+a4)*(b2+b4))*x2*y2+
(p1*a2*(b2+b4)+(1-p1)*b2*(a2+a4)-p*(a2+a4)*(b2+b4))*x2*y4+
(p1*a2*(b2+b4)+(1-p1)*b2*(a2+a4)-(1-p)*(a2+a4)*(b2+b4))*x4*y2
-(p1*a2*(b2+b4)+(1-p1)*b2*(a2+a4))*x4*y4,
(p1*a2*(b2+b5)+(1-p1)*b2*(a2+a5)-(a2+a5)*(b2+b5))*x2*y2+
(p1*a2*(b2+b5)+(1-p1)*b2*(a2+a5)-p*(a2+a5)*(b2+b5))*x2*y5+
(p1*a2*(b2+b5)+(1-p1)*b2*(a2+a5)-(1-p)*(a2+a5)*(b2+b5))*x5*y2
-(p1*a2*(b2+b5)+(1-p1)*b2*(a2+a5))*x5*y5,
(p1*a3*(b3+b4)+(1-p1)*b3*(a3+a4)-(a3+a4)*(b3+b4))*x3*y3+
(p1*a3*(b3+b4)+(1-p1)*b3*(a3+a4)-p*(a3+a4)*(b3+b4))*x3*y4+
(p1*a3*(b3+b4)+(1-p1)*b3*(a3+a4)-(1-p)*(a3+a4)*(b3+b4))*x4*y3
-(p1*a3*(b3+b4)+(1-p1)*b3*(a3+a4))*x4*y4,
(p1*a3*(b3+b5)+(1-p1)*b3*(a3+a5)-(a3+a5)*(b3+b5))*x3*y3+
(p1*a3*(b3+b5)+(1-p1)*b3*(a3+a5)-p*(a3+a5)*(b3+b5))*x3*y5+
(p1*a3*(b3+b5)+(1-p1)*b3*(a3+a5)-(1-p)*(a3+a5)*(b3+b5))*x5*y3
-(p1*a3*(b3+b5)+(1-p1)*b3*(a3+a5))*x5*y5,
(p1*a4*(b4+b5)+(1-p1)*b4*(a4+a5)-(a4+a5)*(b4+b5))*x4*y4+
(p1*a4*(b4+b5)+(1-p1)*b4*(a4+a5)-p*(a4+a5)*(b4+b5))*x4*y5+
(p1*a4*(b4+b5)+(1-p1)*b4*(a4+a5)-(1-p)*(a4+a5)*(b4+b5))*x5*y4
-(p1*a4*(b4+b5)+(1-p1)*b4*(a4+a5))*x5*y5,
t23*(x2+x3)-1,
h15*(y1+y5)-1,
h23*(y2+y3)-1>;

GroebnerBasis(I);
```

From the output of the above code, we obtain

$$\mathrm{Bad}(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}, p_1) = \begin{cases} a_i(1+a_i)(1+b_i),\ b_i(1-p_1)+a_i(b_i+p_1-1)-1, & \forall i \in [5] \\ a_i(p_1-1)-b_ip_1-1,\ b_i(a_i+1-p_1)+a_ip_1-1, & \forall i \in [5] \\ (a_i+a_j)(b_i+b_j), & \forall i < j \in [5] \\ a_j(b_j+b_ip_1)+a_ib_j(1-p_1). & \forall i \neq j \in [5] \end{cases} \tag{3.27}$$

by which we can verify that $(\mathbf{a}'_{2:5}, \mathbf{b}'_{2:5}, p'_1) \in \mathbb{C}^9 - Z(\mathbf{a}_{2:5}, \mathbf{b}_{2:5}, p_1)$.

For the second one, we use Magma and Remark 3.7 to check whether (3.26) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5}, p'_1)$.

Listing 3.4: Dimension and degree computations of BTL models with variable p

```
a:=[1,2,3,4,5];
b:=[1,8,9,3,2];
p1:=3/10;
p2:=7/10;

k:=Rationals();
A<x1,x2,x3,x4,x5,y1,y2,y3,y4,y5,t23,h15,h23,p>:=AffineSpace(k,14);
P:=Scheme(A,[
x1-1,
y1-1,
(p1*a[1]*(b[1]+b[2])+(1-p1)*b[1]*(a[1]+a[2])-(a[1]+a[2])*(b[1]+b[2]))*x1*y1
+(p1*a[1]*(b[1]+b[2])+(1-p1)*b[1]*(a[1]+a[2])-p*(a[1]+a[2])*(b[1]+b[2]))*x1*y2
+(p1*a[1]*(b[1]+b[2])+(1-p1)*b[1]*(a[1]+a[2])-(1-p)*(a[1]+a[2])*(b[1]+b[2]))*x2*y1
+(p1*a[1]*(b[1]+b[2])+(1-p1)*b[1]*(a[1]+a[2]))*x2*y2,
(p1*a[1]*(b[1]+b[3])+(1-p1)*b[1]*(a[1]+a[3])-(a[1]+a[3])*(b[1]+b[3]))*x1*y1
+(p1*a[1]*(b[1]+b[3])+(1-p1)*b[1]*(a[1]+a[3])-p*(a[1]+a[3])*(b[1]+b[3]))*x1*y3
+(p1*a[1]*(b[1]+b[3])+(1-p1)*b[1]*(a[1]+a[3])-(1-p)*(a[1]+a[3])*(b[1]+b[3]))*x3*y1
+(p1*a[1]*(b[1]+b[3])+(1-p1)*b[1]*(a[1]+a[3]))*x3*y3,
(p1*a[1]*(b[1]+b[4])+(1-p1)*b[1]*(a[1]+a[4])-(a[1]+a[4])*(b[1]+b[4]))*x1*y1
+(p1*a[1]*(b[1]+b[4])+(1-p1)*b[1]*(a[1]+a[4])-p*(a[1]+a[4])*(b[1]+b[4]))*x1*y4
+(p1*a[1]*(b[1]+b[4])+(1-p1)*b[1]*(a[1]+a[4])-(1-p)*(a[1]+a[4])*(b[1]+b[4]))*x4*y1
+(p1*a[1]*(b[1]+b[4])+(1-p1)*b[1]*(a[1]+a[4]))*x4*y4,
(p1*a[1]*(b[1]+b[5])+(1-p1)*b[1]*(a[1]+a[5])-(a[1]+a[5])*(b[1]+b[5]))*x1*y1
+(p1*a[1]*(b[1]+b[5])+(1-p1)*b[1]*(a[1]+a[5])-p*(a[1]+a[5])*(b[1]+b[5]))*x1*y5
```

```
+(p1*a[1]*(b[1]+b[5])+(1-p1)*b[1]*(a[1]+a[5])-(1-p)*(a[1]+a[5])*(b[1]+b[5]))*x5*y1
+(p1*a[1]*(b[1]+b[5])+(1-p1)*b[1]*(a[1]+a[5]))*x5*y5,
(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3])-(a[2]+a[3])*(b[2]+b[3]))*x2*y2
+(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3])-p*(a[2]+a[3])*(b[2]+b[3]))*x2*y3
+(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3])-(1-p)*(a[2]+a[3])*(b[2]+b[3]))*x3*y2
+(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3]))*x3*y3,
(p1*a[2]*(b[2]+b[4])+(1-p1)*b[2]*(a[2]+a[4])-(a[2]+a[4])*(b[2]+b[4]))*x2*y2
+(p1*a[2]*(b[2]+b[4])+(1-p1)*b[2]*(a[2]+a[4])-p*(a[2]+a[4])*(b[2]+b[4]))*x2*y4
+(p1*a[2]*(b[2]+b[4])+(1-p1)*b[2]*(a[2]+a[4])-(1-p)*(a[2]+a[4])*(b[2]+b[4]))*x4*y2
+(p1*a[2]*(b[2]+b[4])+(1-p1)*b[2]*(a[2]+a[4]))*x4*y4,
(p1*a[2]*(b[2]+b[5])+(1-p1)*b[2]*(a[2]+a[5])-(a[2]+a[5])*(b[2]+b[5]))*x2*y2
+(p1*a[2]*(b[2]+b[5])+(1-p1)*b[2]*(a[2]+a[5])-p*(a[2]+a[5])*(b[2]+b[5]))*x2*y5
+(p1*a[2]*(b[2]+b[5])+(1-p1)*b[2]*(a[2]+a[5])-(1-p)*(a[2]+a[5])*(b[2]+b[5]))*x5*y2
+(p1*a[2]*(b[2]+b[5])+(1-p1)*b[2]*(a[2]+a[5]))*x5*y5,
(p1*a[3]*(b[3]+b[4])+(1-p1)*b[3]*(a[3]+a[4])-(a[3]+a[4])*(b[3]+b[4]))*x3*y3
+(p1*a[3]*(b[3]+b[4])+(1-p1)*b[3]*(a[3]+a[4])-p*(a[3]+a[4])*(b[3]+b[4]))*x3*y4
+(p1*a[3]*(b[3]+b[4])+(1-p1)*b[3]*(a[3]+a[4])-(1-p)*(a[3]+a[4])*(b[3]+b[4]))*x4*y3
+(p1*a[3]*(b[3]+b[4])+(1-p1)*b[3]*(a[3]+a[4]))*x4*y4,
(p1*a[3]*(b[3]+b[5])+(1-p1)*b[3]*(a[3]+a[5])-(a[3]+a[5])*(b[3]+b[5]))*x3*y3
+(p1*a[3]*(b[3]+b[5])+(1-p1)*b[3]*(a[3]+a[5])-p*(a[3]+a[5])*(b[3]+b[5]))*x3*y5
+(p1*a[3]*(b[3]+b[5])+(1-p1)*b[3]*(a[3]+a[5])-(1-p)*(a[3]+a[5])*(b[3]+b[5]))*x5*y3
+(p1*a[3]*(b[3]+b[5])+(1-p1)*b[3]*(a[3]+a[5]))*x5*y5,
(p1*a[4]*(b[4]+b[5])+(1-p1)*b[4]*(a[4]+a[5])-(a[4]+a[5])*(b[4]+b[5]))*x4*y4
+(p1*a[4]*(b[4]+b[5])+(1-p1)*b[4]*(a[4]+a[5])-p*(a[4]+a[5])*(b[4]+b[5]))*x4*y5
+(p1*a[4]*(b[4]+b[5])+(1-p1)*b[4]*(a[4]+a[5])-(1-p)*(a[4]+a[5])*(b[4]+b[5]))*x5*y4
+(p1*a[4]*(b[4]+b[5])+(1-p1)*b[4]*(a[4]+a[5]))*x5*y5,
t23*(x2+x3)-1,
h15*(y1+y5)-1,
h23*(y2+y3)-1
]);


Dimension(P);
-> 0

Degree(P);
-> 2
```

From Listing 3.4, `Dimension(P)=0` and `Degree(P)=2` means (3.26) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for this choice of $(\mathbf{a}'_{1:5}, \mathbf{b}'_{1:5}, p'_1)$. Until now, we have finished checking the Assumption 3.3 for (3.26) and conclude the proof for $n = 5$.

Case $n \geqslant 6$. For the case $n \geqslant 6$, we can employ a same procedure as we did in the proof of Proposition 3.14. The idea is to split the equation system (3.5) into two parts. One only involves the variables $(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}, p)$, for which we have exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) generically, using induction step. The other part is the remaining equations in (3.5), from which we can build up a system of linear equations in (x_n, y_n) for each of the 2 solutions of $(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}, p)$. By the knowledge of linear algebra, generically (i.e. outside the set of parameters that vanish the coefficient matrix of this system of linear equations) (x_n, y_n) has a unique solution in $\mathbb{C}$ (counted with multiplicity). Altogether, this shows that (3.5) has 2 solutions in $\mathbb{C}$ (counted with multiplicity) and hence conclude the proof for the case $n \geqslant 6$. □

Remark 3.18. *In this case, $n \geqslant 5$ is tight. Since when $n \leqslant 4$, we have the number of equations be smaller than the number of variables and so we do not expect the identifiability.*

3.4.2 Mixtures of MNL Models with 3-slate

In this section, we will study the mixtures of MNL models with 3-slate. We will consider two parameter spaces as before. One is the space of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n})$ and the other is the space of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1)$.

Parameter space of $(a_{2:n}, b_{2:n})$

Now we go through our method for the generic identifiability in the first parameter space.

We will separately consider two cases $p_1 \neq 0.5$ and $p_1 = 0.5$. For $p_1 \neq 0.5$, we show that the equation system achieves generic identifiability. For $p_1 = 0.5$, we show that the equation system achieves generic identifiability up to reordering. A proposition is rigorously written below.

Proposition 3.19. *Suppose $n \geqslant 4$.*

1. *If* $p_1 = 0.3, p_2 = 0.7$*, then* (3.8) *has a unique solution in* $\mathbb{C}$ (*counted with multiplicity*) *for all* $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{\text{MNL}}^{2n-2}$ *but a set of* λ_{2n-2}*-measure zero, given by* $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$.

   That is, if $p_1 = 0.7, p_2 = 0.3$*, then the generic identifiability of two mixtures of MNL models with 3-slate holds.*

2. *If* $p_1 = p_2 = 0.5$*, then* (3.8) *has a unique solution* (*up to reordering*) *in* $\mathbb{C}$ (*counted with multiplicity*) *for all* $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{\text{MNL}}^{2n-2}$ *but a set of* λ_{2n-2}*-measure zero, given by* $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$ *or* $(\mathbf{x}, \mathbf{y}) = (\mathbf{b}, \mathbf{a})$.

   That is, if $p_1 = p_2 = 0.5$*, then the generic identifiability* (*up to reordering*) *of two mixtures of MNL models with 3-slate holds.*

Proof of Proposition 3.19. It suffices to prove the case $p_1 = 0.7$ since the remaining case $p_1 = 0.5$ can be treated in exactly the same way. As before, this proposition is proved by induction on n.

Case $n = 4$**.** In this case, we can expand (3.8) such that its coefficients are given by polynomials in $(\mathbf{a}, \mathbf{b})$:

$$\begin{cases} x_1 = y_1 = 1, \\ (p_1 x_i(y_i + y_j + y_k) + p_2 y_i(x_i + x_j + x_k))(b_i + b_j + b_k)(a_i + a_j + a_k) = \\ (p_1 a_i(b_i + b_j + b_k) + p_2 b_i(a_i + a_j + a_k))\,(x_i + x_j + x_k)(y_i + y_j + y_k), \\ \quad \forall i < j < k \in [4]. \end{cases} \tag{3.28}$$

Note this this is NOT a faithful transformation of (3.8) (but can only increase the number of solutions). To proceed, we need to determine $Z(\mathbf{a}_{2:5}, \mathbf{b}_{2:5})$ introduced in Lemma 3.5. This can be done by Magma.

Listing 3.5: Gröbner basis of MNL models with fixed p

```
P<x1,x2,x3,x4,y1,y2,y3,y4,
  a1,a2,a3,a4,b1,b2,b3,b4>:=FreeAlgebra(Rationals(),16,"lex");

I:=ideal<P|x1-1,y1-1,a1-1,b1-1,
```

```
(3/10*x1*(y1+y2+y3)+(1-3/10)*y1*(x1+x2+x3))*(b1+b2+b3)*(a1+a2+a3)-
(3/10*a1*(b1+b2+b3)+(1-3/10)*b1*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
(3/10*x2*(y1+y2+y3)+(1-3/10)*y2*(x1+x2+x3))*(b1+b2+b3)*(a1+a2+a3)-
(3/10*a2*(b1+b2+b3)+(1-3/10)*b2*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
(3/10*x1*(y1+y2+y4)+(1-3/10)*y1*(x1+x2+x4))*(b1+b2+b4)*(a1+a2+a4)-
(3/10*a1*(b1+b2+b4)+(1-3/10)*b1*(a1+a2+a4))*(x1+x2+x4)*(y1+y2+y4),
(3/10*x2*(y1+y2+y4)+(1-3/10)*y2*(x1+x2+x4))*(b1+b2+b4)*(a1+a2+a4)-
(3/10*a2*(b1+b2+b4)+(1-3/10)*b2*(a1+a2+a4))*(x1+x2+x4)*(y1+y2+y4),
(3/10*x1*(y1+y3+y4)+(1-3/10)*y1*(x1+x3+x4))*(b1+b3+b4)*(a1+a3+a4)-
(3/10*a1*(b1+b3+b4)+(1-3/10)*b1*(a1+a3+a4))*(x1+x3+x4)*(y1+y3+y4),
(3/10*x3*(y1+y3+y4)+(1-3/10)*y3*(x1+x3+x4))*(b1+b3+b4)*(a1+a3+a4)-
(3/10*a3*(b1+b3+b4)+(1-3/10)*b3*(a1+a3+a4))*(x1+x3+x4)*(y1+y3+y4),
(3/10*x2*(y2+y3+y4)+(1-3/10)*y2*(x2+x3+x4))*(b2+b3+b4)*(a2+a3+a4)-
(3/10*a2*(b2+b3+b4)+(1-3/10)*b2*(a2+a3+a4))*(x2+x3+x4)*(y2+y3+y4),
(3/10*x3*(y2+y3+y4)+(1-3/10)*y3*(x2+x3+x4))*(b2+b3+b4)*(a2+a3+a4)-
(3/10*a3*(b2+b3+b4)+(1-3/10)*b3*(a2+a3+a4))*(x2+x3+x4)*(y2+y3+y4)>;

GroebnerBasis(I);
```

From the output, we have

$$\mathrm{Bad}(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) =$$
$$\begin{cases}
(1 + a_i + a_4)(1 + b_i + b_4), \quad \forall i = 2, 3, \\
10a_2 b_2 + 7(a_3 + a_4)b_2 + 3a_2(b_3 + b_4) \\
79 + 49a_2 + 70a_4 + 9b_2 - 28a_2 b_3 - 49a_3 b_3 - 49a_4 b_3 + 58b_4 - 49a_3 b_4 \\
3a_2 b_2 - 4a_2(b_3 + b_4) - 7(a_3 + a_4)(b_3 + b_4) \\
51 + 21a_2 + 42a_4 - 19b_2 + 28a_3 b_2 - 21a_3 b_3 - 21a_4 b_3 + 30b_4 - 21a_3 b_4 \\
7a_2 b_2 + (a_3 + a_4)(4b_2 - 3(b_3 + b_4)) \\
1 - a_2 b_3 + 2b_4 + a_2 b_4 + a_3(b_4 - b_2) + a_4(2 + b_2 + b_3 + 2b_4) \\
-60 - 21a_3 - 42a_4 + 70a_2 b_2 + 49a_4 b_2 + 40b_2 - 9b_3 - 18b_4 + 21a_2 b_4 \\
a_i(1 + b_i + b_j) + (1 + a_j)(b_i - b_4) - a_4(1 + b_j + b_4) \quad \forall (i, j) \in \{(2, 3), (3, 2)\} \\
a_2(1 + b_2 + b_4) + a_3(1 + b_3 + b_4) + (1 + a_4)(2 + b_2 + b_3 + 2b_4) \\
-21a_4 - (40 + 70a_2 + 49a_3)b_2 - 3(7a_2 b_3 + 3b_4) \\
79 + 70a_j + 9b_i - 7a_i(-7 + 3b_i - 4b_j) + 58b_j + 49a_j b_j, \quad \forall (i, j) \in \{(2, 3), (2, 4), (3, 4)\} \\
51 + 42a_j - 19b_i - 28a_j b_i - 7a_3(+7b_i - 3) + 30b_j + 21a_j b_j, \quad \forall (i, j) \in \{(2, 3), (2, 4), (3, 4)\} \\
7a_3(13 + 3b_2 + 10b_3) + (79 + 49a_2)(b_3 - b_4) - 7a_4(13 + 3b_2 + 10b_4) \\
-7a_3(-3 + 7b_2) + 7a_4(-3 + 7b_2) - 3(-3 + 7a_2)(b_3 - b_4) \\
7a_3(10 + 7b_3) + 2(29 + 14a_2)(b_3 - b_4) - 7a_4(10 + 7b_4) \\
20 + 7a_2 + 7a_3 + 14a_4 + 3b_2 + 3b_3 + 6b_4 \\
(3 + 7a_1 + 7a_i + 7a_4 + 3b_i + 3b_4)(7a_i - 7a_4 + 3b_i - 3b_4), \quad \forall i = 2, 3 \\
7(a_2 + a_4)b_3 + a_3(3b_2 + 10b_3 + 3b_4) \\
7a_2(b_2 + b_4) + 7a_4(b_2 + b_4) + a_3(4b_2 - 3b_3 + 4b_4) \\
7a_3 b_3 - a_2(3b_2 - 4b_3 + 3b_4) - a_4(3b_2 - 4b_3 + 3b_4) \\
7a_3(b_2 + b_4) + a_2(10b_2 + 3b_3 + 10b_4) + a_4(10b_2 + 3b_3 + 10b_4) \\
30 - 40b_i - 70a_i b_i + 21a_4 - 49a_4 b_i + 9b_4 - 21a_i b_4 \quad \forall i = 2, 3 \\
30 + 49a_3 b_2 - 7a_4(-6 + 7b_2) + 21a_2 b_3 + 18b_4 - 21a_2 b_4 \\
30 + 49a_3 b_2 + 21a_2 b_3 - 40b_2 + 21a_4 + 9b_4 \\
100 + 91a_4 + 30b_3 + 21a_4 b_3 + 79b_4 + 70a_4 b_4 + 7a_3(10 + 7b_4) \\
100 + 91a_4 + 79b_2 - 21a_3 b_2 - 49b_2 + 70a_2 - 49a_2 b_3 - 70a_3 b_3 - 70a_4 b_3 + 79b_4 - 70a_3 b_4 \\
10 + 7a_i + 7a_j + 3b_i + 3b_j, \quad \forall (i, j) \in \{(2, 3), (2, 4), (3, 4)\} \\
10 + 14a_4 + 6b_4
\end{cases} \tag{3.29}$$

Based on Theorem 3.12, we claim by checking Assumption 3.2′ and 3.3 for (3.28) that (3.28) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{MNL}$ but a set of λ_6-measure zero.

Assumption 3.2′: This is clear since

$$(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4}) \tag{3.30}$$

is a solution of (3.28) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{MNL}$.

Assumption 3.3: Choose $(\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}) = (1, 2, 3, 4; 1, 5, 4, 2)$. It is routine to check that $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}) \in \mathbb{C}^6 - Z(\mathbf{a}_{2:4}, \mathbf{b}_{2:4})$ using (3.29). Since the associated equation system (3.28) is of $\mathbb{Q}$-coefficient, we can use Magma (see (Bosma et al., 1997) and Remark 3.7) to check whether it has a unique solution in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4})$.

Listing 3.6: Dimension and degree computations of MNL model with fixed p

```
p1:=3/10;
a:=[1,2,3,4];
b:=[1,5,4,2];

k:=Rationals();
A<x1,x2,x3,x4,y1,y2,y3,y4>:=AffineSpace(k,8);
P:=Scheme(A,
[
x1-1,
y1-1,
(p1*x1*(y1+y2+y3)+(1-p1)*y1*(x1+x2+x3))*(b[1]+b[2]+b[3])*(a[1]+a[2]+a[3])
-(p1*a[1]*(b[1]+b[2]+b[3])+(1-p1)*b[1]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
(p1*x2*(y1+y2+y3)+(1-p1)*y2*(x1+x2+x3))*(b[1]+b[2]+b[3])*(a[1]+a[2]+a[3])
-(p1*a[2]*(b[1]+b[2]+b[3])+(1-p1)*b[2]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
(p1*x1*(y1+y2+y4)+(1-p1)*y1*(x1+x2+x4))*(b[1]+b[2]+b[4])*(a[1]+a[2]+a[4])
-(p1*a[1]*(b[1]+b[2]+b[4])+(1-p1)*b[1]*(a[1]+a[2]+a[4]))*(x1+x2+x4)*(y1+y2+y4),
(p1*x2*(y1+y2+y4)+(1-p1)*y2*(x1+x2+x4))*(b[1]+b[2]+b[4])*(a[1]+a[2]+a[4])
-(p1*a[2]*(b[1]+b[2]+b[4])+(1-p1)*b[2]*(a[1]+a[2]+a[4]))*(x1+x2+x4)*(y1+y2+y4),
(p1*x1*(y1+y3+y4)+(1-p1)*y1*(x1+x3+x4))*(b[1]+b[3]+b[4])*(a[1]+a[3]+a[4])
-(p1*a[1]*(b[1]+b[3]+b[4])+(1-p1)*b[1]*(a[1]+a[3]+a[4]))*(x1+x3+x4)*(y1+y3+y4),
(p1*x3*(y1+y3+y4)+(1-p1)*y3*(x1+x3+x4))*(b[1]+b[3]+b[4])*(a[1]+a[3]+a[4])
-(p1*a[3]*(b[1]+b[3]+b[4])+(1-p1)*b[3]*(a[1]+a[3]+a[4]))*(x1+x3+x4)*(y1+y3+y4),
(p1*x2*(y2+y3+y4)+(1-p1)*y2*(x2+x3+x4))*(b[2]+b[3]+b[4])*(a[2]+a[3]+a[4])
-(p1*a[2]*(b[2]+b[3]+b[4])+(1-p1)*b[2]*(a[2]+a[3]+a[4]))*(x2+x3+x4)*(y2+y3+y4),
(p1*x3*(y2+y3+y4)+(1-p1)*y3*(x2+x3+x4))*(b[2]+b[3]+b[4])*(a[2]+a[3]+a[4])
-(p1*a[3]*(b[2]+b[3]+b[4])+(1-p1)*b[3]*(a[2]+a[3]+a[4]))*(x2+x3+x4)*(y2+y3+y4)
]);

Dimension(P);
-> 0
```

```
Degree(P);
-> 1
```

From Listing 3.6, `Dimension(P)=0` and `Degree(P)=1` means (3.28) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for this choice of $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4})$.

Thus by Theorem 3.12, we prove that (3.28) (and hence (3.8)) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{MNL}$ but a set V_4 of λ_6-measure zero.

Case $n \geqslant 4$. Suppose the conclusion holds for $n-1$. We need to prove that

$$\begin{cases} x_1 = y_1 = 1, \\ p_1 x_i(y_i + y_j + y_k) + p_2 y_i(x_i + x_j + x_k) = \\ \eta_{i,j,k}(x_i + x_j + x_k)(y_i + y_j + y_k), \quad \forall i < j < k \in [n]. \end{cases} \tag{3.31}$$

has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{MNL}$ but a set V_n of λ_{2n-2}-measure zero. We begin with splitting (3.31) into two parts:

$$\begin{cases} x_1 = y_1 = 1, \\ p_1 x_i(y_i + y_j + y_k) + p_2 y_i(x_i + x_j + x_k) = \\ \eta_{i,j,k}(x_i + x_j + x_k)(y_i + y_j + y_k) = 0, \quad \forall i < j < k \in [n-1]. \end{cases} \tag{3.32}$$

and

$$\begin{aligned} &p_1 x_i(y_i + y_j + y_n) + p_2 y_i(x_i + x_j + x_n) = \\ &\eta_{i,j,n}(x_i + x_j + x_n)(y_i + y_j + y_n), \quad \forall i < j \in [n-1] \end{aligned} \tag{3.33}$$

By induction hypothesis, there exists a λ_{2n-4}-measure zero subset $V_{n-1} \subset \mathbb{C}^{2n-4}$ such that (3.32) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for any $(\mathbf{a}_{2:n-1}, \mathbf{b}_{2:n-1}) \in Q^{2n-4}_{MNL} - V_{n-1}$, given by

$$(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}) = (\mathbf{a}_{1:n-1}, \mathbf{b}_{1:n-1}).$$

Plugging this solution into (3.33) and simplifying, we obtain

$$c_{i,j,n} + (p_2 b_i - \eta_{i,j,n}(b_i + b_j))x_n + (p_1 a_i - \eta_{i,j,n}(a_i + a_j))y_n - \eta_{i,j,n} x_n y_n = 0 \tag{3.34}$$

where

$$c_{i,j,n} := p_1 a_i (b_i + b_j) + p_2 b_i (a_i + a_j) - \eta_{i,j,n}(a_i + a_j)(b_i + b_j).$$

Since $\eta_{1,2,n} \neq 0$, by the case of $(i, j) = (1, 2)$ in (3.34), we have

$$x_n y_n = \frac{c_{1,2,n} + (p_2 b_1 - \eta_{1,2,n}(b_1 + b_2))x_n + (p_1 a_1 - \eta_{1,2,n}(a_1 + a_2))y_n}{\eta_{1,2,n}}$$

Plugging it into the cases of $(i, j) = (1, 3), (2, 3)$ in (3.34), we further have a system of linear equations in variables (x_n, y_n)

$$\begin{cases} d_{13} x_n + e_{13} y_n = f_2, \\ d_{23} x_n + e_{23} y_n = f_3. \end{cases} \tag{3.35}$$

where

$$\begin{aligned} d_{ij} &= \eta_{1,2,n}(p_2 b_i - \eta_{i,j,n}(b_i + b_j)) - \eta_{i,j,n}(p_2 b_1 - \eta_{1,2,n}(b_1 + b_2)) \\ e_{ij} &= \eta_{1,2,n}(p_1 a_i - \eta_{i,j,n}(a_i + a_j)) - \eta_{i,j,n}(p_1 a_1 - \eta_{1,2,n}(a_1 + a_2)) \end{aligned}$$

and f_2, f_3 are some constant. Denote the coefficients matrix of (3.35) by

$$A := \begin{pmatrix} d_{13} & e_{13} \\ d_{23} & e_{13} \end{pmatrix}$$

and define

$$
\begin{aligned}
W_n &:= \{(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in \mathbb{C}^{2n-2} : \det(A) = 0\} \subset \mathbb{C}^{2n-2}. \\
V_n &:= (V_{n-1} \times \mathbb{C}^2) \cup W_n \subset \mathbb{C}^{2n-2}.
\end{aligned}
$$

Note that V_n is a λ_{2n-2}-measure zero subset. We finish the proof by claiming that (3.31) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{MNL}$ but the set V_n. Indeed, for any $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{MNL} - V_n$, we have

1. since $(\mathbf{a}_{2:n-1}, \mathbf{b}_{2:n-1}) \notin V_{n-1}$, (3.32) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by

$$(\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}) = (\mathbf{a}_{1:n-1}, \mathbf{b}_{1:n-1}).$$

2. since $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \notin W_n$, (3.35) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by

$$(x_n, y_n) = (a_n, b_n).$$

This shows that (3.31) has a unique solution in $\mathbb{C}$ (counted with multiplicity) since $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$ is always a solution of it. □

Parameter space of $(a_{2:n}, b_{2:n}, p_1)$

In this case of MNL model with 3-slate in parameter space $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1)$, we study the equation system in variables $(\mathbf{x}, \mathbf{y}, p)$,

$$
\begin{cases}
x_1 = y_1 = 1, \\
p\dfrac{x_i}{x_i + x_j + x_k} + (1-p)\dfrac{y_i}{y_i + y_j + y_k} = \eta_{i,j,k}. \quad \forall i < j < k \in [n]
\end{cases}
\tag{3.36}
$$

Proposition 3.20. *If $n \geqslant 4$, then* (3.36) *has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{MNL,p}$ but a set of λ_{2n-1}-measure zero, given by $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{a}, \mathbf{b}, p_1)$ and $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{b}, \mathbf{a}, 1 - p_1)$.*

Proof of Proposition 3.20. It suffices to prove the initial case $n = 4$ since the case $n \geqslant 4$ can be treated by induction in exactly the same way as Proposition 3.19.

Case $n = 4$. In this case, we can expand (3.36) such that its coefficients are given by polynomials in $(\mathbf{a}, \mathbf{b})$:

$$\begin{cases} x_1 = y_1 = 1, \\ (px_i(y_i + y_j + y_k) + (1-p)y_i(x_i + x_j + x_k))(b_i + b_j + b_k)(a_i + a_j + a_k) = \\ (p_1 a_i(b_i + b_j + b_k) + (1-p_1)b_i(a_i + a_j + a_k))\,(x_i + x_j + x_k)(y_i + y_j + y_k), \\ \hfill \forall i < j < k \in [4] \\ t_{i,j,k}(x_i + x_j + x_k) = 1, \quad \forall i < j < k \in [4] \\ h_{i,j,k}(y_i + y_j + y_k) = 1. \quad \forall i < j < k \in [4] \end{cases} \tag{3.37}$$

Note this this is a faithful transformation of (3.36). To proceed, we need to determine $Z(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1)$ introduced in Lemma 3.5. This can be done by Magma.

Listing 3.7: Gröbner basis of MNL models with variable p

```
P<x1,x2,x3,x4,y1,y2,y3,y4,p,a1,a2,a3,a4,b1,b2,b3,b4,
t123,t124,t134,t234,h123,h124,h134,h234,p1>
:=FreeAlgebra(Rationals(),26,"lex");

I:=ideal<P|x1-1,y1-1,a1-1,b1-1,
(p*x1*(y1+y2+y3)+(1-p)*y1*(x1+x2+x3))*(b1+b2+b3)*(a1+a2+a3)-
(p1*a1*(b1+b2+b3)+(1-p1)*b1*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
(p*x2*(y1+y2+y3)+(1-p)*y2*(x1+x2+x3))*(b1+b2+b3)*(a1+a2+a3)-
(p1*a2*(b1+b2+b3)+(1-p1)*b2*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
(p*x1*(y1+y2+y4)+(1-p)*y1*(x1+x2+x4))*(b1+b2+b4)*(a1+a2+a4)-
(p1*a1*(b1+b2+b4)+(1-p1)*b1*(a1+a2+a4))*(x1+x2+x4)*(y1+y2+y4),
(p*x2*(y1+y2+y4)+(1-p)*y2*(x1+x2+x4))*(b1+b2+b4)*(a1+a2+a4)-
(p1*a2*(b1+b2+b4)+(1-p1)*b2*(a1+a2+a4))*(x1+x2+x4)*(y1+y2+y4),
```

```
(p*x1*(y1+y3+y4)+(1-p)*y1*(x1+x3+x4))*(b1+b3+b4)*(a1+a3+a4)-
(p1*a1*(b1+b3+b4)+(1-p1)*b1*(a1+a3+a4))*(x1+x3+x4)*(y1+y3+y4),
(p*x3*(y1+y3+y4)+(1-p)*y3*(x1+x3+x4))*(b1+b3+b4)*(a1+a3+a4)-
(p1*a3*(b1+b3+b4)+(1-p1)*b3*(a1+a3+a4))*(x1+x3+x4)*(y1+y3+y4),
(p*x2*(y2+y3+y4)+(1-p)*y2*(x2+x3+x4))*(b2+b3+b4)*(a2+a3+a4)-
(p1*a2*(b2+b3+b4)+(1-p1)*b2*(a2+a3+a4))*(x2+x3+x4)*(y2+y3+y4),
(p*x3*(y2+y3+y4)+(1-p)*y3*(x2+x3+x4))*(b2+b3+b4)*(a2+a3+a4)-
(p1*a3*(b2+b3+b4)+(1-p1)*b3*(a2+a3+a4))*(x2+x3+x4)*(y2+y3+y4),
t123*(x1+x2+x3)-1,
t124*(x1+x2+x4)-1,
t134*(x1+x3+x4)-1,
t234*(x2+x3+x4)-1,
h123*(y1+y2+y3)-1,
h124*(y1+y2+y4)-1,
h134*(y1+y3+y4)-1,
h234*(y2+y3+y4)-1>;

GroebnerBasis(I);
```

This gives

$$\mathrm{Bad}(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) =$$

$$\begin{cases}
-1 - 2a_4(1-p_1) - 2b_4p_1 \\
(1 + a_2 + a_3)(1 + b_2 + b_3) \\
(1 + a_2 + a_4)(1 + b_2 + b_4) \\
(1 + a_3 + a_4)(1 + b_3 + b_4) \\
(a_2 + a_3 + a_4)(b_2 + b_3 + b_4) \\
a_2 - a_4(1-p_1) - a_2p_1 + (b_2 - b_4)p_1 \\
a_3 - a_4(1-p_1) - a_3p_1 + (b_3 - b_4)p_1 \\
-1 - b_2 + a_3b_2 - a_2(1-b_3) - a_4(1-b_3) + a_3b_3 - b_4 + a_3b_4 \\
a_2(1 + b_2 + b_3) + (1 + a_3)(b_2 - b_4) - a_4(1 + b_3 + b_4) \\
a_3(1 + b_2 + b_3) + (1 + a_2)(b_3 - b_4) - a_4(1 + b_2 + b_4) \\
(a_3 + a_4)b_2(1-p_1) + a_2(b_2 + (b_3 + b_4)p_1) \\
(a_3 + a_4)b_1(1-p_1) + a_1(1 + (b_3 + b_4)p_1) \\
(a_2 + a_3)b_1(1-p_1) + a_1(1 + (b_2 + b_3)p_1) \\
(a_2 + a_4)b_1(1-p_1) + a_1(1 + (b_2 + b_4)p_1) \\
(a_3 + a_4)b_2(1-p_1) + a_2(b_2 + (b_3 + b_4)p_1) \\
(a_4 + a_2)b_3(1-p_1) + a_3(b_3 + b_2p_1 + b_3p_1 + b_4p_1) \\
a_4 - b_2 - a_3b_2 - a_4p_1 + a_3b_2p_1 + b_4p_1 - a_2(1 + b_2 + b_3p_1) \\
(a_2b_3 - b_4)p_1 - b_2(1 - a_3 + a_3p_1) - b_1(1 + a_2 + a_4 - a_4p_1) \\
1 + b_2 + a_4(1 + b_2)(1-p_1) + b_4p_1 + a_2(1 + b_2 + b_4p_1) \\
a_2(b_3 + b_4)(1-p_1) + (a_3 + a_4)(b_3 + b_4 + b_2p_1) \\
a_4(b_2 + b_4) + a_2(b_2 + b_4 + b_3p_1) + a_3(b_2 + b_4 - b_2p_1 - b_3p_1 - b_4p_1) \\
a_2(b_3 + b_4)(1-p_1) + (a_3 + a_4)(b_3 + b_4 + b_2p_1) \\
(1 + a_2)b_4p_1 + b1(1 + a_2 + a_4 - a_4p_1) + b_2(1 + a_2 + a_4 - a_4p_1) \\
(1 + a_2)b_3p_1 + b1(1 + a_2 + a_3 - a_3p_1) + b_2(1 + a_2 + a_3 - a_3p_1) \\
a_2(1 + b_2 + b_4) + a_3(1 + b_3 + b_4) + (1 + a_4)(2 + b_2 + b_3 + 2b_4) \\
1 - a_3b_2(1-p_1) + a_4(2 + b_2)(1-p_1) - a_2b_3p_1 + 2b_4p_1 + a_2b_4p_1 \\
-2 - 2a_4 - a_2(1-p_1) - a_3(1-p_1) + 2a_4p_1 - (b_2 + b_3 + 2b_4)p_1 \\
a_4(1 + b_2)p_1 + b_4(1 + a_2 + a_4 - p_1 - a_2p_1) \\
a_3(1 + b_2)p_1 + b_3(1 + a_2 + a_3 - p_1 - a_2p_1) \\
a_4(1 + b_2)p_1 + b_4(1 + a_4 + a_2(1-p_1) - p_1) \\
a_3(1 + b_2)(1-p_1) - a_4(1 + b_2)(1-p_1) + (1 + a_2)(b_3 - b_4)p_1 \\
2 + a_2 + a_3 + 2a_4 - a_3p_1 - 2a_4p_1 + (b_3 + (2 + a_2)b_4)p_1 + b_2(1 + a_2 + a_4 - a_4p_1) \\
-a_4b_3 + b_4 - a_2b_3(1-p_1) + a_4p_1 - b_4p_1 - a_3(b_3 + b_4 + b_2p_1) \\
1 - a_2b_3 + b_3p_1 + a_2b_3p_1 + b_4p_1 + a_4(1 - b_3 - p_1) - a_3(b_3 + p_1 + b_2p_1 + b_3p_1 + b_4p_1 - 1) \\
(1 + a_2)(b_3 - b_4)(1-p_1) + a_3(b_3 + p_1 + b_2p_1) - a_4(b_4 + p_1 + b_2p_1) \\
1 + a_2(b_2 + b_4) + a_4(1 + b_2 + b_4 - p_1) + b_3p_1(1 + a_2) + b_4p_1 + a_3(1 + b_2(1-p_1) + b_4(1-p_1) - p_1 - b_3p_1)
\end{cases} \tag{3.38}$$

Based on Theorem 3.12, we claim by checking Assumption 3.2′ and 3.3 for (3.37) that (3.37) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in Q^7_{MNL,p}$ but a set of λ_7-measure zero.

Assumption 3.2′: This is clear since

$$\begin{aligned}(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}, \mathbf{t}, \mathbf{h}, p) &= \left(\mathbf{a}_{1:4}, \mathbf{b}_{1:4}, \left(\frac{1}{a_i + a_j + a_k}\right)_{i<j<k}, \left(\frac{1}{b_i + b_j + b_k}\right)_{i<j<k}, p_1\right) \\ (\mathbf{x}_{1:4}, \mathbf{y}_{1:4}, \mathbf{t}, \mathbf{h}, p) &= \left(\mathbf{b}_{1:4}, \mathbf{a}_{1:4}, \left(\frac{1}{b_i + b_j + b_k}\right)_{i<j<k}, \left(\frac{1}{a_i + a_j + a_k}\right)_{i<j<k}, 1 - p_1\right)\end{aligned} \tag{3.39}$$

are two (distinct) solutions of (3.37) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in Q^7_{MNL,p} - Z$, where $Z \subset \mathbb{A}^7_{\mathbb{C}}$ is the Zariski closed proper subset defined by

$$\{(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in \mathbb{A}^6_{\mathbb{C}} \times \mathbb{A}^1_{\mathbb{C}} : p_1 - 0.5 = 0\} \subset \mathbb{A}^7_{\mathbb{C}}$$

Assumption 3.3: Choose $(\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}, p'_1) = (1, 2, 3, 4; 1, 5, 4, 2; 0.7)$. It is routine to check that $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}, p'_1) \in \mathbb{C}^6 \times (0, 1) - Z(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1)$ using (3.38). Since the associated equation system (3.37) is of $\mathbb{Q}$-coefficient, we can use Magma (see (Bosma et al., 1997) and Remark 3.7) to check whether it has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}, p'_1)$.

Listing 3.8: Dimension and degree computations of MNL models with variable p

```
a:=[1,2,3,4];
b:=[1,5,4,2];
p1:=7/10;

k:=Rationals();
A<x1,x2,x3,x4,y1,y2,y3,y4,p,
  t123,t124,t134,t234,
  h123,h124,h134,h234>:=AffineSpace(k,17);
P:=Scheme(A,[x1-1,y1-1,
(p*x1*(y1+y2+y3)+(1-p)*y1*(x1+x2+x3))*(b[1]+b[2]+b[3])*(a[1]+a[2]+a[3])-
(p1*a[1]*(b[1]+b[2]+b[3])+(1-p1)*b[1]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
(p*x2*(y1+y2+y3)+(1-p)*y2*(x1+x2+x3))*(b[1]+b[2]+b[3])*(a[1]+a[2]+a[3])-
(p1*a[2]*(b[1]+b[2]+b[3])+(1-p1)*b[2]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
(p*x1*(y1+y2+y4)+(1-p)*y1*(x1+x2+x4))*(b[1]+b[2]+b[4])*(a[1]+a[2]+a[4])-
(p1*a[1]*(b[1]+b[2]+b[4])+(1-p1)*b[1]*(a[1]+a[2]+a[4]))*(x1+x2+x4)*(y1+y2+y4),
```

```
(p*x2*(y1+y2+y4)+(1-p)*y2*(x1+x2+x4))*(b[1]+b[2]+b[4])*(a[1]+a[2]+a[4])-
(p1*a[2]*(b[1]+b[2]+b[4])+(1-p1)*b[2]*(a[1]+a[2]+a[4]))*(x1+x2+x4)*(y1+y2+y4),
(p*x1*(y1+y3+y4)+(1-p)*y1*(x1+x3+x4))*(b[1]+b[3]+b[4])*(a[1]+a[3]+a[4])-
(p1*a[1]*(b[1]+b[3]+b[4])+(1-p1)*b[1]*(a[1]+a[3]+a[4]))*(x1+x3+x4)*(y1+y3+y4),
(p*x3*(y1+y3+y4)+(1-p)*y3*(x1+x3+x4))*(b[1]+b[3]+b[4])*(a[1]+a[3]+a[4])-
(p1*a[3]*(b[1]+b[3]+b[4])+(1-p1)*b[3]*(a[1]+a[3]+a[4]))*(x1+x3+x4)*(y1+y3+y4),
(p*x2*(y2+y3+y4)+(1-p)*y2*(x2+x3+x4))*(b[2]+b[3]+b[4])*(a[2]+a[3]+a[4])-
(p1*a[2]*(b[2]+b[3]+b[4])+(1-p1)*b[2]*(a[2]+a[3]+a[4]))*(x2+x3+x4)*(y2+y3+y4),
(p*x3*(y2+y3+y4)+(1-p)*y3*(x2+x3+x4))*(b[2]+b[3]+b[4])*(a[2]+a[3]+a[4])-
(p1*a[3]*(b[2]+b[3]+b[4])+(1-p1)*b[3]*(a[2]+a[3]+a[4]))*(x2+x3+x4)*(y2+y3+y4),
(x1+x2+x3)*t123 -1,
(x1+x2+x4)*t124 -1,
(x1+x3+x4)*t134 -1,
(x2+x3+x4)*t234 -1,
(y1+y2+y3)*h123-1,
(y1+y2+y4)*h124-1,
(y1+y3+y4)*h134-1,
(y2+y3+y4)*h234-1
]);

Dimension(P);
-> 0

Degree(P);
-> 2
```

From Listing 3.8, `Dimension(P)=0` and `Degree(P)=2` means (3.37) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for this choice of $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}, p'_1)$.

Thus by Theorem 3.12, we prove that (3.37) (and hence (3.36)) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in Q^7_{MNL,p}$ but a set V_4 of λ_7-measure zero. $\square$

Remark 3.21. *For both parameter spaces in two mixture of MNL models with 3-slate, it is tight for the generic identifiability of the mixtures MNL models with 3-slate on* $n \geqslant 4$ *since when* $n \leqslant 3$*, we have less equations than variables, which makes the mixture model not identifiable.*

3.4.3 Mixtures of Plackett-Luce Models

In this section, we consider the mixtures of Plackett-Luce models. The first part is about the generic identifiability on the parameter space of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n})$ and the second part is on the parameter space of $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p)$. The second part has been studied by Zhao et al. (2016), where the authors take a tensor-decomposition technique.

Parameter space of $(a_{2:n}, b_{2:n})$

We now prove the generic identifiability of two mixtures of Plackett-Luce model (3.11) for any $n \geqslant 3$ with given p_1 and p_2. Note that (3.11) has at least one solution in $\mathbb{C}$ coming from the initial data $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$. Our goal is to show it is also the unique solution of (3.11) in $\mathbb{C}$. To be precise, we have the proposition below.

Proposition 3.22. *If* $n \geqslant 3$ *and* $p_1 = 0.7, p_2 = 0.3$*, then* (3.11) *has a unique solution in* $\mathbb{C}$ (*counted with multiplicity*) *for all* $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q_{PL}^{2n-2}$ *but a set of* λ_{2n-2}*-measure zero, given by* $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$.

Proof of Proposition 3.22. According to Section 3.2.2, we consider the following equation system, whose equations are not the honest equations in the mixtures of Plackett-Luce models (3.11) but only a linear combination of the part of them,

$$\begin{cases} \eta_{1,i,\cdot} = p_1 \dfrac{x_1 x_i}{1-x_1} + p_2 \dfrac{y_1 y_i}{1-y_1}, & \forall i \in \{2,3,4\} \\ \eta_{2,i,\cdot} = p_1 \dfrac{x_2 x_i}{1-x_2} + p_2 \dfrac{y_2 y_i}{1-y_2}, & \forall i \in \{1,3,4\} \\ \eta_{3,i,\cdot} = p_1 \dfrac{x_3 x_i}{1-x_3} + p_2 \dfrac{y_3 y_i}{1-y_3}, & \forall i \in \{1,2,4\} \\ \eta_{1,i,\cdot} = p_1 \dfrac{x_1 x_i}{1-x_1} + p_2 \dfrac{y_1 y_i}{1-y_1}, & \forall i \geqslant 5, \\ \eta_{2,i,\cdot} = p_1 \dfrac{x_2 x_i}{1-x_2} + p_2 \dfrac{y_2 y_i}{1-y_2}, & \forall i \geqslant 5, \\ \eta_{3,i,\cdot} = p_1 \dfrac{x_3 x_i}{1-x_3} + p_2 \dfrac{y_3 y_i}{1-y_3}, & \forall i \geqslant 5. \end{cases} \tag{3.40}$$

where $\eta_{k,l,\cdot}$ corresponds to the probability that $k \succ l \succ \mathit{others}$. The arithmetic logic behind (3.40) can be found in Appendix A.2.2. Clearly, the solutions of (3.11) are necessarily solutions of (3.40). And we know there is always a destined solution of (3.11) from how we defined $\boldsymbol{\eta}$. This implies that it suffices to prove the generic identifiability of (3.40).

To apply the results from Section 3.3, we first translate (3.40) into the following (equivalent) one such that the coefficients are polynomials in $(\mathbf{a}, \mathbf{b})$,

$$\begin{cases} (p_1a_1a_i(1-b_1)+p_2b_1b_i(1-a_1))(1-x_1)(1-y_1) = \\ \quad (1-a_1)(1-b_1)(p_1x_1x_i(1-y_1)+p_2y_1y_i(1-x_1)), \quad \forall i \in \{2,3,4\} \\ (p_1a_2a_i(1-b_2)+p_2b_2b_i(1-a_2))(1-x_2)(1-y_2) = \\ \quad (1-a_2)(1-b_2)(p_1x_2x_i(1-y_2)+p_2y_2y_i(1-x_2)), \quad \forall i \in \{1,3,4\} \\ (p_1a_3a_i(1-b_3)+p_2b_3b_i(1-a_3))(1-x_3)(1-y_3) = \\ \quad (1-a_3)(1-b_3)(p_1x_3x_i(1-y_3)+p_2y_3y_i(1-x_3)), \quad \forall i \in \{1,2,4\} \\ (p_1a_1a_i(1-b_1)+p_2b_1b_i(1-a_1))(1-x_1)(1-y_1) = \\ \quad (1-a_1)(1-b_1)(p_1x_1x_i(1-y_1)+p_2y_1y_i(1-x_1)), \quad \forall i \geqslant 5 \\ (p_1a_2a_i(1-b_2)+p_2b_2b_i(1-a_2))(1-x_2)(1-y_2) = \\ \quad (1-a_2)(1-b_2)(p_1x_2x_i(1-y_2)+p_2y_2y_i(1-x_2)), \quad \forall i \geqslant 5 \\ (p_1a_3a_i(1-b_3)+p_2b_3b_i(1-a_3))(1-x_3)(1-y_3) = \\ \quad (1-a_3)(1-b_3)(p_1x_3x_i(1-y_3)+p_2y_3y_i(1-x_3)), \quad \forall i \geqslant 5 \\ t_i(1-x_i) = 1, \quad \forall i \in [n], \\ h_i(1-y_i) = 1, \quad \forall i \in [n]. \end{cases} \tag{3.41}$$

i.e. (3.40) and (3.41) share the same solution(s). It then suffices to prove the generic identifiability of (3.41). We will fist consider the case where $n = 4$ and then make use of its result to prove the cases where $n \geqslant 5$.

Case $n = 4$. In this case, (3.41) reads as

$$\begin{cases} (p_1 a_1 a_i(1-b_1) + p_2 b_1 b_i(1-a_1))(1-x_1)(1-y_1) = \\ \quad (1-a_1)(1-b_1)(p_1 x_1 x_i(1-y_1) + p_2 y_1 y_i(1-x_1)), \quad \forall i \in \{2,3,4\} \\ (p_1 a_2 a_i(1-b_2) + p_2 b_2 b_i(1-a_2))(1-x_2)(1-y_2) = \\ \quad (1-a_2)(1-b_2)(p_1 x_2 x_i(1-y_2) + p_2 y_2 y_i(1-x_2)), \quad \forall i \in \{1,3,4\} \\ (p_1 a_3 a_i(1-b_3) + p_2 b_3 b_i(1-a_3))(1-x_3)(1-y_3) = \\ \quad (1-a_3)(1-b_3)(p_1 x_3 x_i(1-y_3) + p_2 y_3 y_i(1-x_3)), \quad \forall i \in \{1,2,4\} \\ t_i(1-x_i) = 1, \quad \forall i \in [4], \\ h_i(1-y_i) = 1, \quad \forall i \in [4]. \end{cases} \tag{3.42}$$

To apply Theorem 3.12 to (3.42), we need to check Assumptions 3.2 and 3.3 for (3.42).

For Assumption 3.2: It is clear that (3.42) has at least one solution in $\mathbb{C}$ given by $(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4})$.

For Assumption 3.3: we first compute the Gröbner basis of (3.42).

Listing 3.9: Gröbner basis of PL models with fixed p

```
P<x1,x2,x3,x4,y1,y2,y3,y4,t1,t2,t3,t4,h1,h2,h3,h4,
  a1,a2,a3,a4,b1,b2,b3,b4>
  :=FreeAlgebra(Rationals(),24,"lex");

I:=ideal<P|
(7/10*a1*a2*(1-b1)+(1-7/10)*b1*b2*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(7/10*x1*x2*(1-y1)+(1-7/10)*y1*y2*(1-x1)),
(7/10*a1*a3*(1-b1)+(1-7/10)*b1*b3*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(7/10*x1*x3*(1-y1)+(1-7/10)*y1*y3*(1-x1)),
(7/10*a1*a4*(1-b1)+(1-7/10)*b1*b4*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(7/10*x1*x4*(1-y1)+(1-7/10)*y1*y4*(1-x1)),
(7/10*a2*a1*(1-b2)+(1-7/10)*b2*b1*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(7/10*x2*x1*(1-y2)+(1-7/10)*y2*y1*(1-x2)),
(7/10*a2*a3*(1-b2)+(1-7/10)*b2*b3*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(7/10*x2*x3*(1-y2)+(1-7/10)*y2*y3*(1-x2)),
(7/10*a2*a4*(1-b2)+(1-7/10)*b2*b4*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(7/10*x2*x4*(1-y2)+(1-7/10)*y2*y4*(1-x2)),
(7/10*a3*a1*(1-b3)+(1-7/10)*b3*b1*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(7/10*x3*x1*(1-y3)+(1-7/10)*y3*y1*(1-x3)),
(7/10*a3*a2*(1-b3)+(1-7/10)*b3*b2*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(7/10*x3*x2*(1-y3)+(1-7/10)*y3*y2*(1-x3)),
```

```
(7/10*a3*a4*(1-b3)+(1-7/10)*b3*b4*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(7/10*x3*x4*(1-y3)+(1-7/10)*y3*y4*(1-x3)),
t1*(1-x1)-1,
t2*(1-x2)-1,
t3*(1-x3)-1,
t4*(1-x4)-1,
h1*(1-y1)-1,
h2*(1-y2)-1,
h3*(1-y3)-1,
h4*(1-y4)-1>;

GroebnerBasis(I);
```

From the output of the above code, we obtain

$$\mathrm{Bad}(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) = \begin{cases} 7a_i a_j(b_i - 1) + 3(a_i - 1)b_i b_j, & \forall i < j \in [4], \\ a_i - 1,\ b_i - 1. & \forall i \in [3]. \end{cases} \tag{3.43}$$

Choose $(\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}) = (1/10, 2/10, 3/10, 4/10; 1/20, 7/20, 9/20, 3/20)$. It is routine to check that $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}) \in \mathbb{C}^6 - Z(\mathbf{a}_{2:4}, \mathbf{b}_{2:4})$ using (3.43). Then we check that (3.42) has a unique solution in $\mathbb{C}$ (counted with multiplicity) by Magma in Listing 3.10.

Listing 3.10: Dimension and degree computations of PL models with fixed p

```
a:=[1/10,2/10,3/10,4/10];
b:=[1/20,7/20,9/20,3/20];
p1:=7/10;
p2:=3/10;

k:=Rationals();
A<x1,x2,x3,x4,y1,y2,y3,y4,t1,t2,t3,t4,h1,h2,h3,h4>:=AffineSpace(k,16);
P:=Scheme(A,
[
(p1*a[1]*a[2]*(1-b[1])+(1-p1)*b[1]*b[2]*(1-a[1]))*(1-x1)*(1-y1)-
(1-a[1])*(1-b[1])*(p1*x1*x2*(1-y1)+(1-p1)*y1*y2*(1-x1)),
(p1*a[1]*a[3]*(1-b[1])+(1-p1)*b[1]*b[3]*(1-a[1]))*(1-x1)*(1-y1)-
(1-a[1])*(1-b[1])*(p1*x1*x3*(1-y1)+(1-p1)*y1*y3*(1-x1)),
(p1*a[1]*a[4]*(1-b[1])+(1-p1)*b[1]*b[4]*(1-a[1]))*(1-x1)*(1-y1)-
(1-a[1])*(1-b[1])*(p1*x1*x4*(1-y1)+(1-p1)*y1*y4*(1-x1)),
(p1*a[2]*a[1]*(1-b[2])+(1-p1)*b[2]*b[1]*(1-a[2]))*(1-x2)*(1-y2)-
(1-a[2])*(1-b[2])*(p1*x2*x1*(1-y2)+(1-p1)*y2*y1*(1-x2)),
```

```
(p1*a[2]*a[3]*(1-b[2])+(1-p1)*b[2]*b[3]*(1-a[2]))*(1-x2)*(1-y2)-
(1-a[2])*(1-b[2])*(p1*x2*x3*(1-y2)+(1-p1)*y2*y3*(1-x2)),
(p1*a[2]*a[4]*(1-b[2])+(1-p1)*b[2]*b[4]*(1-a[2]))*(1-x2)*(1-y2)-
(1-a[2])*(1-b[2])*(p1*x2*x4*(1-y2)+(1-p1)*y2*y4*(1-x2)),
(p1*a[3]*a[1]*(1-b[3])+(1-p1)*b[3]*b[1]*(1-a[3]))*(1-x3)*(1-y3)-
(1-a[3])*(1-b[3])*(p1*x3*x1*(1-y3)+(1-p1)*y3*y1*(1-x3)),
(p1*a[3]*a[2]*(1-b[3])+(1-p1)*b[3]*b[2]*(1-a[3]))*(1-x3)*(1-y3)-
(1-a[3])*(1-b[3])*(p1*x3*x2*(1-y3)+(1-p1)*y3*y2*(1-x3)),
(p1*a[3]*a[4]*(1-b[3])+(1-p1)*b[3]*b[4]*(1-a[3]))*(1-x3)*(1-y3)-
(1-a[3])*(1-b[3])*(p1*x3*x4*(1-y3)+(1-p1)*y3*y4*(1-x3)),
t1*(1-x1)-1,
t2*(1-x2)-1,
t3*(1-x3)-1,
t4*(1-x4)-1,
h1*(1-y1)-1,
h2*(1-y2)-1,
h3*(1-y3)-1,
h4*(1-y4)-1
]);

Dimension(P);
-> 0

Degree(P);
-> 1
```

From Listing 3.10, `Dimension(P)=0` and `Degree(P)=1` means (3.42) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for this choice of $(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4})$.

Thus by Theorem 3.12, we prove that (3.42) (and hence (3.11)) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{PL}$ but a set V_4 of λ_6-measure zero.

Case $n \geqslant 5$. In the case $n \geqslant 5$, we consider the following two parts of (3.41): one is (3.42) and the other is

$$\begin{cases} (p_1 a_2 a_i (1-b_2) + p_2 b_2 b_i (1-a_2))(1-x_2)(1-y_2) \\ \quad = (1-a_2)(1-b_2)(p_1 x_2 x_i (1-y_2) + p_2 y_2 y_i (1-x_2)), \quad \forall i \geqslant 5, \\ (p_1 a_3 a_i (1-b_3) + p_2 b_3 b_i (1-a_3))(1-x_3)(1-y_3) \\ \quad = (1-a_3)(1-b_3)(p_1 x_3 x_i (1-y_3) + p_2 y_3 y_i (1-x_3)), \quad \forall i \geqslant 5. \end{cases} \tag{3.44}$$

From the case where $n = 4$, we know there exists a λ_6-measure zero set V_4 such that (3.42) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{PL} - V_4$, given by

$$(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4}).$$

To proceed, we determine (x_i, y_i) for each $i \geqslant 5$. Plugging $(\mathbf{x}_{2:3}, \mathbf{y}_{2:3}) = (\mathbf{a}_{2:3}, \mathbf{b}_{2:3})$ into (3.44) and simplifying, we obtain a system of linear equations in (x_i, y_i),

$$\begin{cases} c_2 x_i + d_2 y_i = c_2 a_i + d_2 b_i, \\ c_3 x_i + d_3 y_i = c_3 a_i + d_3 b_i. \end{cases} \tag{3.45}$$

where $c_i := p_1 a_i(1-b_i)$ and $d_i := p_2 b_i(1-a_i)$ for $i = 2, 3$. If the coefficient matrix of this system of linear equations is non-zero, i.e. $c_2 d_3 - c_3 d_2 \neq 0$ (this is a condition on a_2, a_3, b_2, b_3), then (3.45) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by $(x_i, y_i) = (a_i, b_i)$.

Altogether, we can define

$$V_n := \{(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in \mathbb{C}^{2n-2} : (\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in V_4 \text{ or } c_2 d_3 - c_3 d_2 = 0\} \subset \mathbb{C}^{2n-2}$$

which is of λ_{2n-2}-measure zero by Lemma A.39 since it is defined by a non-zero polynomial. From the arguments above, for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{PL} - V_n$, (3.41) (and hence (3.11)) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$. This finishes the proof. □

Remark 3.23. *For this case, $n \geqslant 4$ might not be tight for the two mixtures of PL models. For $n = 3$, we conjecture the generic identifiability also holds.*

Parameter space of $(a_{2:n}, b_{2:n}, p_1)$

In this subsection, we consider the Plackett-Luce model with the parameter space $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{PL,p}$. The equation system becomes (3.13). Note that (3.13) has at least 2 solutions in $\mathbb{C}$ coming from the initial data

$$(\mathbf{x}, \mathbf{y}, p) = (\mathbf{a}, \mathbf{b}, p_1) \text{ and } (\mathbf{x}, \mathbf{y}, p) = (\mathbf{b}, \mathbf{a}, 1 - p_1).$$

Our goal is to show the equation system (3.13) has exactly these two solutions in $\mathbb{C}$. To be precise, we have the proposition below.

Proposition 3.24. *If $n \geqslant 4$, then (3.13) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{PL,p}$ but a set of λ_{2n-1}-measure zero, given by $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{a}, \mathbf{b}, p_1)$ and $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{b}, \mathbf{a}, 1 - p_1)$.*

Proof of Proposition 3.24. As before, we consider the following part of (3.13),

$$\begin{cases} \eta_{1,i,\cdot} = p\dfrac{x_1 x_i}{1-x_1} + (1-p)\dfrac{y_1 y_i}{1-y_1}, & \forall i \in \{2,3,4\} \\ \eta_{2,i,\cdot} = p\dfrac{x_2 x_i}{1-x_2} + (1-p)\dfrac{y_2 y_i}{1-y_2}, & \forall i \in \{1,3,4\} \\ \eta_{3,i,\cdot} = p\dfrac{x_3 x_i}{1-x_3} + (1-p)\dfrac{y_3 y_i}{1-y_3}, & \forall i \in \{1,2,4\} \\ \eta_{1,i,\cdot} = p\dfrac{x_1 x_i}{1-x_1} + (1-p)\dfrac{y_1 y_i}{1-y_1}, & \forall i \geqslant 5, \\ \eta_{2,i,\cdot} = p\dfrac{x_2 x_i}{1-x_2} + (1-p)\dfrac{y_2 y_i}{1-y_2}, & \forall i \geqslant 5, \\ \eta_{3,i,\cdot} = p\dfrac{x_3 x_i}{1-x_3} + (1-p)\dfrac{y_3 y_i}{1-y_3}, & \forall i \geqslant 5. \end{cases} \tag{3.46}$$

Note that the equations in (3.46) are a linear combination of those in Plackett-Luce model (3.13). Therefore, it suffices to show the generic identifiability of (3.46) (up to reordering) for $n \geqslant 4$, i.e. it has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) given by $(\mathbf{a}, \mathbf{b}, p_1)$ and $(\mathbf{b}, \mathbf{a}, 1 - p_1)$.

As before, we first get rid of the denominators in (3.46) in an equivalent way by introducing new variables and equations on t_i, h_i and then multiply $(1 - a_i)(1 - b_i)$ to make the coefficients of (3.46) as polynomials in $(\mathbf{a}, \mathbf{b}, p_1)$. Finally, we obtain

$$\begin{cases} (p_1 a_1 a_i(1-b_1) + p_2 b_1 b_i(1-a_1))(1-x_1)(1-y_1) \\ \quad = (1-a_1)(1-b_1)(p x_1 x_i(1-y_1) + (1-p) y_1 y_i(1-x_1)), \quad \forall i \in \{2,3,4\} \\ (p_1 a_2 a_i(1-b_2) + p_2 b_2 b_i(1-a_2))(1-x_2)(1-y_2) \\ \quad = (1-a_2)(1-b_2)(p x_2 x_i(1-y_2) + (1-p) y_2 y_i(1-x_2)), \quad \forall i \in \{1,3,4\} \\ (p_1 a_3 a_i(1-b_3) + p_2 b_3 b_i(1-a_3))(1-x_3)(1-y_3) \\ \quad = (1-a_3)(1-b_3)(p x_3 x_i(1-y_3) + (1-p) y_3 y_i(1-x_3)), \quad \forall i \in \{1,2,4\} \\ (p_1 a_1 a_i(1-b_1) + p_2 b_1 b_i(1-a_1))(1-x_1)(1-y_1) \\ \quad = (1-a_1)(1-b_1)(p x_1 x_i(1-y_1) + (1-p) y_1 y_i(1-x_1)), \quad \forall i \geqslant 5, \\ (p_1 a_2 a_i(1-b_2) + p_2 b_2 b_i(1-a_2))(1-x_2)(1-y_2) \\ \quad = (1-a_2)(1-b_2)(p x_2 x_i(1-y_2) + (1-p) y_2 y_i(1-x_2)), \quad \forall i \geqslant 5, \\ (p_1 a_3 a_i(1-b_3) + p_2 b_3 b_i(1-a_3))(1-x_3)(1-y_3) \\ \quad = (1-a_3)(1-b_3)(p x_3 x_i(1-y_3) + (1-p) y_3 y_i(1-x_3)), \quad \forall i \geqslant 5. \\ t_i(1-x_i) = 1, \qquad \forall i \in [n], \\ h_i(1-y_i) = 1, \qquad \forall i \in [n]. \end{cases} \tag{3.47}$$

As before, we first consider the initial case $n = 4$ and the general case follows exactly in the same manner.

Case $n = 4$. In this case, (3.47) reads as:

$$\begin{cases} (p_1 a_1 a_i(1-b_1) + p_2 b_1 b_i(1-a_1))(1-x_1)(1-y_1) \\ \quad = (1-a_1)(1-b_1)(p x_1 x_i(1-y_1) + (1-p) y_1 y_i(1-x_1)), \quad \forall i \in \{2,3,4\} \\ (p_1 a_2 a_i(1-b_2) + p_2 b_2 b_i(1-a_2))(1-x_2)(1-y_2) \\ \quad = (1-a_2)(1-b_2)(p x_2 x_i(1-y_2) + (1-p) y_2 y_i(1-x_2)), \quad \forall i \in \{1,3,4\} \\ (p_1 a_3 a_i(1-b_3) + p_2 b_3 b_i(1-a_3))(1-x_3)(1-y_3) \\ \quad = (1-a_3)(1-b_3)(p x_3 x_i(1-y_3) + (1-p) y_3 y_i(1-x_3)), \quad \forall i \in \{1,2,4\} \\ t_i(1-x_i) = 1, \quad \forall i \in [4], \\ h_i(1-y_i) = 1, \quad \forall i \in [4]. \end{cases} \tag{3.48}$$

We will apply Theorem 3.12, for which we need to check Assumption 3.2′ and Assumption 3.3 for (3.48).

For Assumption 3.2′: we already know that (from how we define η and transform the equations)

$$(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}, p) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4}, p_1), (\mathbf{b}_{1:4}, \mathbf{a}_{1:4}, 1 - p_1)$$

are 2 (distinct) solutions of (3.48) for all $(a_{2:4}, b_{2:4}, p_1) \in Q^7_{PL,p} - Z$, where $Z \subset \mathbb{A}^7_{\mathbb{C}}$ is the Zariski closed proper subset defined by

$$Z := \left\{(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in \mathbb{A}^7_{\mathbb{C}} : p_1 - 0.5 = 0\right\}$$

For Assumption 3.3: We choose

$$(\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}, p'_1) = \left(\frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10}; \frac{1}{20}, \frac{14}{20}, \frac{2}{20}, \frac{3}{20}; \frac{7}{10}\right)$$

To proceed, we first compute the Gröbner basis with the Magma code 3.11.

Listing 3.11: Gröbner basis of PL models with fixed p

```
P<x1,x2,x3,x4,y1,y2,y3,y4,p,t1,t2,t3,t4,h1,h2,h3,h4,
  a1,a2,a3,a4,b1,b2,b3,b4,p1>
  :=FreeAlgebra(Rationals(),26,"lex");

I:=ideal<P|
(p1*a1*a2*(1-b1)+(1-p1)*b1*b2*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(p*x1*x2*(1-y1)+(1-p)*y1*y2*(1-x1)),
(p1*a1*a3*(1-b1)+(1-p1)*b1*b3*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(p*x1*x3*(1-y1)+(1-p)*y1*y3*(1-x1)),
(p1*a1*a4*(1-b1)+(1-p1)*b1*b4*(1-a1))*(1-x1)*(1-y1)-
(1-a1)*(1-b1)*(p*x1*x4*(1-y1)+(1-p)*y1*y4*(1-x1)),
(p1*a2*a1*(1-b2)+(1-p1)*b2*b1*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(p*x2*x1*(1-y2)+(1-p)*y2*y1*(1-x2)),
(p1*a2*a3*(1-b2)+(1-p1)*b2*b3*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(p*x2*x3*(1-y2)+(1-p)*y2*y3*(1-x2)),
(p1*a2*a4*(1-b2)+(1-p1)*b2*b4*(1-a2))*(1-x2)*(1-y2)-
(1-a2)*(1-b2)*(p*x2*x4*(1-y2)+(1-p)*y2*y4*(1-x2)),
(p1*a3*a1*(1-b3)+(1-p1)*b3*b1*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(p*x3*x1*(1-y3)+(1-p)*y3*y1*(1-x3)),
```

```
(p1*a3*a2*(1-b3)+(1-p1)*b3*b2*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(p*x3*x2*(1-y3)+(1-p)*y3*y2*(1-x3)),
(p1*a3*a4*(1-b3)+(1-p1)*b3*b4*(1-a3))*(1-x3)*(1-y3)-
(1-a3)*(1-b3)*(p*x3*x4*(1-y3)+(1-p)*y3*y4*(1-x3)),
(p1*a4*a1*(1-b4)+(1-p1)*b4*b1*(1-a4))*(1-x4)*(1-y4)-
(1-a4)*(1-b4)*(p*x4*x1*(1-y4)+(1-p)*y4*y1*(1-x4)),
(p1*a4*a2*(1-b4)+(1-p1)*b4*b2*(1-a4))*(1-x4)*(1-y4)-
(1-a4)*(1-b4)*(p*x4*x2*(1-y4)+(1-p)*y4*y2*(1-x4)),
t1*(1-x1)-1,
t2*(1-x2)-1,
t3*(1-x3)-1,
t4*(1-x4)-1,
h1*(1-y1)-1,
h2*(1-y2)-1,
h3*(1-y3)-1,
h4*(1-y4)-1>;

GroebnerBasis(I);
```

and determine $\text{Bad}(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1)$ based on the program outputs.

$$\text{Bad}(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) = \begin{cases} (a_i - 1)b_i b_j(p_1 - 1) + a_i a_j (1 - b_i) p_1, & \forall i < j \in [4], \\ a_i - 1,\ b_i - 1 & \forall i \in [3] \end{cases} \tag{3.49}$$

by which we can verify that the selected parameter

$$(\mathbf{a}'_{2:4}, \mathbf{b}'_{2:4}, p'_1) \notin Z(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1).$$

The Magma code in Listing 3.12 helps to claim that (3.48) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}, p'_1)$, given by $(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}, p) = (\mathbf{a}'_{1:4}, \mathbf{b}'_{1:4}, p'_1)$ or $(\mathbf{b}'_{1:4}, \mathbf{a}'_{1:4}, 1 - p'_1)$.

Listing 3.12: Dimension and degree computations of PL models with variable p

```
a:=[1/10,2/10,3/10,4/10];
b:=[1/20,14/20,2/20,3/20];
p1:=7/10;

k:=Rationals();
A<x1,x2,x3,x4,y1,y2,y3,y4,p,t1,t2,t3,t4,h1,h2,h3,h4>
:=AffineSpace(k,17);
```

```
P:=Scheme(A,
[
(p1*a[1]*a[2]*(1-b[1])+(1-p1)*b[1]*b[2]*(1-a[1]))*(1-x1)*(1-y1)
-(1-a[1])*(1-b[1])*(p*x1*x2*(1-y1)+(1-p)*y1*y2*(1-x1)),
(p1*a[1]*a[3]*(1-b[1])+(1-p1)*b[1]*b[3]*(1-a[1]))*(1-x1)*(1-y1)
-(1-a[1])*(1-b[1])*(p*x1*x3*(1-y1)+(1-p)*y1*y3*(1-x1)),
(p1*a[1]*a[4]*(1-b[1])+(1-p1)*b[1]*b[4]*(1-a[1]))*(1-x1)*(1-y1)
-(1-a[1])*(1-b[1])*(p*x1*x4*(1-y1)+(1-p)*y1*y4*(1-x1)),
(p1*a[2]*a[1]*(1-b[2])+(1-p1)*b[2]*b[1]*(1-a[2]))*(1-x2)*(1-y2)
-(1-a[2])*(1-b[2])*(p*x2*x1*(1-y2)+(1-p)*y2*y1*(1-x2)),
(p1*a[2]*a[3]*(1-b[2])+(1-p1)*b[2]*b[3]*(1-a[2]))*(1-x2)*(1-y2)
-(1-a[2])*(1-b[2])*(p*x2*x3*(1-y2)+(1-p)*y2*y3*(1-x2)),
(p1*a[2]*a[4]*(1-b[2])+(1-p1)*b[2]*b[4]*(1-a[2]))*(1-x2)*(1-y2)
-(1-a[2])*(1-b[2])*(p*x2*x4*(1-y2)+(1-p)*y2*y4*(1-x2)),
(p1*a[3]*a[1]*(1-b[3])+(1-p1)*b[3]*b[1]*(1-a[3]))*(1-x3)*(1-y3)
-(1-a[3])*(1-b[3])*(p*x3*x1*(1-y3)+(1-p)*y3*y1*(1-x3)),
(p1*a[3]*a[2]*(1-b[3])+(1-p1)*b[3]*b[2]*(1-a[3]))*(1-x3)*(1-y3)
-(1-a[3])*(1-b[3])*(p*x3*x2*(1-y3)+(1-p)*y3*y2*(1-x3)),
(p1*a[3]*a[4]*(1-b[3])+(1-p1)*b[3]*b[4]*(1-a[3]))*(1-x3)*(1-y3)
-(1-a[3])*(1-b[3])*(p*x3*x4*(1-y3)+(1-p)*y3*y4*(1-x3)),
(p1*a[4]*a[1]*(1-b[4])+(1-p1)*b[4]*b[1]*(1-a[4]))*(1-x4)*(1-y4)
-(1-a[4])*(1-b[4])*(p*x4*x1*(1-y4)+(1-p)*y4*y1*(1-x4)),
(p1*a[4]*a[2]*(1-b[4])+(1-p1)*b[4]*b[2]*(1-a[4]))*(1-x4)*(1-y4)
-(1-a[4])*(1-b[4])*(p*x4*x2*(1-y4)+(1-p)*y4*y2*(1-x4)),
t1*(1-x1)-1,
t2*(1-x2)-1,
t3*(1-x3)-1,
t4*(1-x4)-1,
h1*(1-y1)-1,
h2*(1-y2)-1,
h3*(1-y3)-1,
h4*(1-y4)-1
]);

Dimension(P);
-> 0

Degree(P);
-> 2
```

Altogether, we prove that (3.48) (and hence (3.13)) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathfrak{a}_{2:4}, \mathfrak{b}_{2:4}, p_1) \in Q^7_{PL,p}$ but a set V_4 of λ_7-measure zero.

Case $n \geqslant 4$. In the case $n \geqslant 4$, we consider the following two parts of (3.47): one is (3.48) and the other is

$$\begin{cases} (p_1 a_2 a_i(1-b_2) + p_2 b_2 b_i(1-a_2))(1-x_2)(1-y_2) \\ \quad = (1-a_2)(1-b_2)(p x_2 x_i(1-y_2) + (1-p) y_2 y_i(1-x_2)), \quad \forall i \geqslant 5, \\ (p_1 a_3 a_i(1-b_3) + p_2 b_3 b_i(1-a_3))(1-x_3)(1-y_3) \\ \quad = (1-a_3)(1-b_3)(p x_3 x_i(1-y_3) + (1-p) y_3 y_i(1-x_3)), \quad \forall i \geqslant 5. \end{cases} \tag{3.50}$$

From the case where $n = 4$, we know there exists a λ_7-measure zero set V_4 such that (3.48) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in Q^7_{PL,p} - V_4$, given by

$$(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}, p) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4}, p_1) \text{ and } (\mathbf{b}_{1:4}, \mathbf{a}_{1:4}, 1-p_1).$$

To proceed, we determine (x_i, y_i) for each $i \geqslant 5$. Let $c_i := p_1 a_i(1-b_i)$ and $d_i := p_2 b_i(1-a_i)$ for $i = 2, 3$.

1. Plugging $(\mathbf{x}_{2:3}, \mathbf{y}_{2:3}, p) = (\mathbf{a}_{2:3}, \mathbf{b}_{2:3}, p_1)$ into (3.50) and simplifying, we obtain a system of linear equations in (x_i, y_i),

$$\begin{cases} c_2 x_i + d_2 y_i = c_2 a_i + d_2 b_i, \\ c_3 x_i + d_3 y_i = c_3 a_i + d_3 b_i. \end{cases} \tag{3.51}$$

   If the coefficient matrix of this system of linear equations is non-zero, i.e. $c_2 d_3 - c_3 d_2 \neq 0$ (this is a condition on a_2, a_3, b_2, b_3), then (3.51) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by $(x_i, y_i) = (a_i, b_i)$.

2. Plugging $(\mathbf{x}_{2:3}, \mathbf{y}_{2:3}, p) = (\mathbf{b}_{2:3}, \mathbf{a}_{2:3}, 1-p_1)$ into (3.50) and simplifying,

we obtain a system of linear equations in (x_i, y_i),

$$\begin{cases} d_2 x_i + c_2 y_i = c_2 a_i + d_2 b_i, \\ d_3 x_i + c_3 y_i = c_3 a_i + d_3 b_i. \end{cases} \tag{3.52}$$

If the coefficient matrix of this system of linear equations is non-zero, i.e. $c_2 d_3 - c_3 d_2 \neq 0$ (this is a condition on a_2, a_3, b_2, b_3), then (3.52) has a unique solution in $\mathbb{C}$ (counted with multiplicity), given by $(x_i, y_i) = (b_i, a_i)$.

Altogether, we can define

$$V_n := \{(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in \mathbb{C}^{2n-1} : (\mathbf{a}_{2:4}, \mathbf{b}_{2:4}, p_1) \in V_4 \text{ or } c_2 d_3 - c_3 d_2 = 0\} \subset \mathbb{C}^{2n-1}$$

which is of λ_{2n-1}-measure zero by Lemma A.39 since it is defined by a non-zero polynomial. From the arguments above, for all $(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}, p_1) \in Q^{2n-1}_{PL,p} - V_n$, (3.47) (and hence (3.13)) has exactly 2 solutions in $\mathbb{C}$ (counted with multiplicity), given by $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{a}, \mathbf{b}, p_1)$ and $(\mathbf{b}, \mathbf{a}, 1 - p_1)$. This finishes the proof. $\square$

Remark 3.25. *For this case, when $n \leqslant 2$, the equations system has less equations than variables so we do not expect the model to be identifiable. When $n = 3$, the number variables of the equation system is less than or equal to the number equations. But we do not have conclusions on generic identifiability of the mixtures of PL models in these cases.*

3.5 Conclusions and Discussions

In this study, we develop a general theory to check generic identifiability of polynomial systems under two assumptions, which can be applied to many machine learning identifiability problems. The techniques are brand new, making use of the machinery from algebraic geometry. In particular, we present how we adopt our main theory to three examples of mixtures of ranking models. Among the examples, an important contribution is to affirm the generic identifiability of two mixtures of BTL models, which was unsolved in the literature. Note we also include the proof taking our method on mixtures of MNL models with 2-slate and 3-slate in Appendix A.2.3. The uniform mixture case has been proved in Chierichetti et al. (2018) and we extend it to non-uniform mixture and to the parameter space including p.

For the three examples discussed in Section 3.4, a stronger conjecture for the cases when p_1 is given is that if there exists a $p_1 \in (0, 1)$ such that Assumption 3.2′ and Assumption 3.3 to are true for some mixtures of ranking models that can be transformed to polynomial systems, then for any $p_1 \in (0, 1)$ the generic identifiability holds for that mixtures of ranking models. As we have tried many experiments for different p_1, the generic identifiability always holds.

A limitation of our work is on checking Assumption 3.3, which depends on whether the software tool can solve a specific polynomial system fast or slow. Specifically, we have also tried verifying Assumption 3.3 for three mixtures of BTL models. However, the Magma procedure did not finish for a week. Compared to the work from Zhao et al. (2016), where it showed the generic identifiability of k-mixture Plackett-Luce model, we did not achieve verifying the generic identifiability for $k \geqslant 3$.

Regarding to the role of our work in machine learning field, often in the literature of algorithms for mixtures models, researchers assume the identifiability at the beginning of the work but without any explanations on why they believe that is true. With such assumption, we can talk about noise perturbation on the observations/data. Otherwise, a small such perturbation can lead to huge/infinite change of the latent parameters. Hence, we believe the guarantee on generic identifiability of mixtures of ranking model is cardiotonic to those researchers to continue their work.

4 DISCUSSIONS

The presented two works show how clustering can be executed during learning. The key idea is to add new variables to differ data points. In the first work, we add γ to represent the data points to be either clean points or outliers. In the second work, we have U to represent a user to be either type 1 or type 2 and add a new parameter vector $\mathbf{b}$ to represent the scores for the type 2 user. We also see that this idea was applied in other works, such as Feng et al. (2014), in which a new vector variable $\mathbf{t} \in \{0, 1\}^n$ is added in a classification setting to split data point into clean ones and corrupted ones.

The goal to do clustering while learning is to amend the model which researchers commonly assumed in the past. The benefit of doing so is to improve the prediction result. The disadvantages is that it increases the complexity for the theoretical analysis and one needs to come out suitable methods for different settings and different models. Other interesting examples of clustering while learning includes robust generalized linear models, high-dimensional linear models, k-mixture regression models and so on. We believe the learning performance can be improved by carefully clustering the data points and designing the models. We hope that more and more approaches via learning along with clustering will be provided for general-purpose use in the future.

A APPENDIX

The supplmentary materials is organized as follows: Chapter A.1.1 presents some additional discussions on β. Chapter A.1.2, Chapter A.1.3, Chapter A.1.4 and Chapter A.1.5 mainly provide proofs respectively for problem reformulation and support recovery, tuning parameter selection and strategy for second pool selection. They may also include additional discussions and formal statements as referred in the main text.

A.1 Appendix for Chapter 2

A.1.1 Additional Discussions

We present more miscellaneous discussions here to readers who may care about β.

Debugging connection to β**.** Throughout this paper, we have focused on estimating γ for the purpose of debugging. A result concerning how the second pool can be used to obtain a better estimate of β is as follows:

Proposition A.1. *Let* $X = USV^\top$ *and* $\widetilde{X} = \widetilde{S}V_0^\top$. *Let* $m < p$. *It holds that*

$$\|V_0(\widehat{\beta} - \beta^*)\|_2 \leqslant \frac{c_1 \sigma \sqrt{m}}{\sqrt{L}\sigma_{\min}(\tilde{S})} + \lambda n \|\tilde{S}^{-2} V_0 V S U z_{\widehat{\gamma}}\|_2, \tag{A.1}$$

where $z_{\widehat{\gamma}}$ *is the subgradient of* $\|\widehat{\gamma}\|_1$.

Proof of Proposition A.1. Recall the objective function (2.3) is

$$(\widehat{\beta}, \widehat{\gamma}) \in \arg \min_{\substack{\beta \in \mathbb{R}^p, \\ \gamma \in \mathbb{R}^n}} \left\{ \frac{1}{2n} \|y - X\beta - \gamma\|_2^2 + \frac{\eta}{2m} \|\widetilde{y} - \widetilde{X}\beta\|_2^2 + \lambda \|\gamma\|_1 \right\}.$$

By KKT conditions of the objective function,

$$\begin{aligned}\nabla_\beta &= -\frac{1}{n}X^\top(y - X\widehat{\beta} - \widehat{\gamma}) - \frac{\eta}{m}\widetilde{X}^\top(\widetilde{y} - \widetilde{X}\widehat{\beta}) = 0;\\ \nabla_\gamma &= -\frac{1}{n}(y - X\widehat{\beta} - \widehat{\gamma}) + \lambda\partial|\widehat{\gamma}| = 0.\end{aligned} \tag{A.2}$$

Plug $y = X\beta^* + \gamma^* + \epsilon$ and $\widetilde{y} = \widetilde{X}\beta^* + \widetilde{\epsilon}$ into (A.2) we obtain

$$-\left(\frac{1}{n}X^\top X + \frac{\eta}{m}\widetilde{X}^\top\widetilde{X}\right)(\beta^* - \widehat{\beta}) - \frac{1}{n}X^\top(\gamma^* - \widehat{\gamma}) - \frac{1}{n}X^\top\epsilon - \frac{\eta}{m}\widetilde{X}^\top\widetilde{\epsilon} = 0; \tag{A.3a}$$

$$-\frac{1}{n}X(\beta^* - \widehat{\beta}) - \frac{1}{n}(\gamma^* - \widehat{\gamma}) - \frac{1}{n}\epsilon + \lambda\partial|\widehat{\gamma}| = 0. \tag{A.3b}$$

Mutiply $X^\top$ on (A.3b) and plug it into (A.3a) we get

$$\widetilde{X}^\top\widetilde{X}(\widehat{\beta} - \beta^*) = \lambda\frac{m}{\eta}X^\top\partial|\widehat{\gamma}| + \widetilde{X}\widetilde{\epsilon}. \tag{A.4}$$

Given that $\widetilde{X} = \widetilde{S}V_0^\top$,

$$\widetilde{S}^\top\widetilde{S}V_0^\top(\widehat{\beta} - \beta^*) = \lambda\frac{m}{\eta}V_0^\top X^\top\partial|\widehat{\gamma}| + V_0^\top V_0\widetilde{S}\widetilde{\epsilon}.$$

Plugging into the SVD of $X = USV^\top$, we have

$$\begin{aligned}\left\|V_0^\top(\widehat{\beta} - \beta^*)\right\|_2 &\leqslant \lambda\frac{m}{\eta}\left\|(\widetilde{S}^\top\widetilde{S})^{-1}V_0^\top X^\top\partial|\widehat{\gamma}|\right\|_2 + \|(\widetilde{S}^\top\widetilde{S})^{-1}\widetilde{S}\|\|\widetilde{\epsilon}\|_2\\ &\leqslant \lambda\frac{m}{\eta}\left\|(\widetilde{S}^\top\widetilde{S})^{-1}V_0^\top VSU^\top\partial|\widehat{\gamma}|\right\|_2 + c_1\frac{\sqrt{m}\sigma}{\sqrt{L}\sigma_{\min}(\widetilde{S})}\\ &\leqslant \lambda\frac{m}{\eta}\left\|(\widetilde{S}^\top\widetilde{S})^{-1}V_0^\top VSU^\top\right\|_2\sqrt{n} + c_1\frac{\sqrt{m}\sigma}{\sqrt{L}\sigma_{\min}(\widetilde{S})}\\ &\leqslant c\sigma\sqrt{\frac{\log n}{n}}\frac{m}{\eta}\left\|(\widetilde{S}^\top\widetilde{S})^{-1}S_0^{1/2}\right\|_2 + c_1\frac{\sqrt{m}\sigma}{\sqrt{L}\sigma_{\min}(\widetilde{S})},\end{aligned}$$

with probability at least $1 - \exp(-cm)$. The second step is because $\widetilde{\sigma}$ has subgaussian parameter σ^2/L. □

Note that when $\widetilde{S}$ is chosen large enough, then $\|V_0(\widehat{\beta} - \beta^*)\|_2$ is controlled to a small number. Besides, if the subspace V_0 contains the buggy subspace of X_T, then $\|y_T - y_T^*\|_2$ is well controlled and we can spot the contaminated points. This, together with the orthogonal design we will discuss in Chapter A.1.3, suggests that a successful debugging strategy may be obtained by producing a carefully chosen interaction between the non-buggy subspace (augmented using a second pool of clean data points) and the buggy subspace.

Related work She and Owen (2011). Without the second pool, She and Owen (2011) demonstrated the equivalence of the solution $\widehat{\beta}$ to the joint optimization of the objective (2.3) over (β, γ) to the optimum of a regression M-estimator in β with the Huber loss. This motivates the question of whether the optimizer $\widehat{\beta}$ of the objective (2.3) may similarly be viewed as the optimum of an M-estimation problem.

Proposition A.2. *The solution $\widehat{\beta}$ of the joint optimization problem* (2.3) *is the unique optimum of the following weighted M-estimation problem:*

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_{n\lambda} \left(y_i - x_i^\top \beta \right) + \frac{\eta}{2m} \|\widetilde{y} - \widetilde{X}\beta\|_2^2 \right\}. \tag{A.5}$$

Proof. Recall the definition of the Huber loss function:

$$\ell_k(u) = \begin{cases} \lambda|u| - \frac{k^2}{2}, & \text{if } |u| > k, \\ \frac{u^2}{2}, & \text{if } |u| < k. \end{cases}$$

We will show the desired equivalence via the KKT conditions for both objective functions. Taking gradients with respect to β and γ for the original objective function (2.3), we obtain the following system of equations:

$$0 = \frac{X^\top X}{n}\beta - \frac{X^\top(y-\gamma)}{n} + \eta\left(\frac{\widetilde{X}^\top\widetilde{X}}{m}\beta - \frac{\widetilde{X}^\top\widetilde{y}}{m}\right), \tag{A.6}$$

$$0 = \frac{\gamma}{n} - \frac{y - X\beta}{n} + \lambda\,\mathrm{sign}(\gamma). \tag{A.7}$$

The second equation (A.7) has a unique solution, given by the soft-thresholding function:

$$\gamma = \mathrm{SoftThresh}_{n\lambda}\left(y - X\beta\right),$$

where for scalars $u, k \in \mathbb{R}$, we have

$$\mathrm{SoftThresh}_k(u) = \begin{cases} u - \lambda\,\mathrm{sign}(u), & \text{if } |u| \geqslant k, \\ 0, & \text{if } |u| < k, \end{cases}$$

and $\mathrm{SoftThresh}_k$ acts on vectors componentwise. Plugging back into equation (A.6), we obtain

$$0 = X^\top\left(\frac{X\beta - y}{n} + \frac{1}{n}\mathrm{SoftThresh}_{n\lambda}\left(y - X\beta\right)\right) + \eta\left(\frac{\widetilde{X}^\top\widetilde{X}}{m}\beta - \frac{\widetilde{X}^\top\widetilde{y}}{m}\right). \tag{A.8}$$

We now consider the KKT conditions for the weighted M-estimator (A.5). Taking a gradient with respect to β, we obtain

$$0 = -\sum_{i=1}^{n} \ell'_{n\lambda}\left(y_i - x_i^\top\beta\right)\frac{x_i}{n} + \eta\left(\frac{\widetilde{X}^\top\widetilde{X}}{m}\beta - \frac{\widetilde{X}^\top\widetilde{y}}{m}\right). \tag{A.9}$$

The key is to note that

$$u - \ell'_{n\lambda}(u) = \mathrm{SoftThresh}_{n\lambda}(u),$$

so

$$-\ell'_{n\lambda}\left(y_i - x_i^\top \beta\right)\frac{1}{n} = \frac{x_i^\top \beta - y_i}{n} + \frac{1}{n}\,\text{SoftThresh}_{n\lambda}\left(y_i - x_i^\top \beta\right),$$

from which we may infer the equivalence of equations (A.8) and (A.9). This concludes the proof. □

The proposition also illustrates that the objective uses Huber loss to get the robust estimation $\widehat{\beta}$, and then imply the estimation $\widehat{\gamma}$. Therefore, estimations of β and γ complement each other. Our reformulation more relies on giving a direct analysis of γ and its support.

A.1.2 Appendix for Chapter 2.2

We show reformulation of the objective function in this section.

Proof of Proposition 2.2. Using the notation (2.4), we can translate (2.3) into

$$(\widehat{\beta}, \widehat{\gamma}) \in \arg\min_{\beta,\gamma} \left\{ \frac{1}{2n} \left\| y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right\|_2^2 + \lambda \|\gamma\|_1 \right\}, \tag{A.10}$$

First note that we can split $y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix}$ into two parts by projecting onto the column space of X' and the perpendicular space:

$$\begin{aligned}
&\left\| y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right\|_2^2 \\
&\quad = \left\| P_{X'} \left(y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 + \left\| P_{X'}^{\perp} \left(y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 \\
&\quad = \left\| P_{X'} \left(y' - X'\beta - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 + \left\| P_{X'}^{\perp} \left(y' - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 .
\end{aligned}$$

For any value of $\widehat{\gamma}$, we can choose $\widehat{\beta}$ such that $\left\| P_{X'} \left(y' - X'\widehat{\beta} - \begin{bmatrix} \gamma \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 = 0$, simply by taking $\widehat{\beta} = (X'^\top X')^{-1} X'^\top \left(y' - \begin{bmatrix} \widehat{\gamma} \\ \vec{0}_m \end{bmatrix} \right)$. Hence, we get

$$\left\| y' - X'\beta - \begin{bmatrix} \widehat{\gamma} \\ \vec{0}_m \end{bmatrix} \right\|_2^2 = \left\| P_{X'}^{\perp} \left(y' - \begin{bmatrix} \widehat{\gamma} \\ \vec{0}_m \end{bmatrix} \right) \right\|_2^2 = \left\| P_{X'}^{\perp} y' - \bar{P}\widehat{\gamma} \right\|_2^2,$$

and (A.10) becomes

$$\begin{aligned} \widehat{\gamma} &\in \frac{1}{2n} \left\| P_{X'}^{\perp} y' - \bar{P}\widehat{\gamma} \right\|_2^2 + \lambda \|\widehat{\gamma}\|_1, \\ \widehat{\beta} &= (X'^\top X')^{-1} X'^\top \left(y' - \begin{bmatrix} \widehat{\gamma} \\ \vec{0}_m \end{bmatrix} \right). \end{aligned}$$

Therefore, the two optimization problems share the same solution for $\widehat{\gamma}$. □

A.1.3 Appendix for Chapter 2.3

Notations in appendix: We write $P_{X',TT}^{\perp}$ to represent the submatrix of $P_{X'}^{\perp}$ with rows and column indexed by T. We write $P_{X',T\cdot}^{\perp}$ to represent the submatrix of $P_{X'}^{\perp}$ with rows indexed by T and $P_{X',\cdot T}^{\perp}$ to represent the submatrix of $P_{X'}^{\perp}$ with columns indexed by T. For simplicity, let $\bar{P} = P_{X'}^{\perp} M_{[n]}$. We slightly abuse notation by using $\bar{P}_T$ and $\bar{P}_{T^c}$ to denote $\bar{P}_{\cdot T}$ and $\bar{P}_{\cdot T^c}$, respectively.

In this appendix, we provide proofs and additional details for the results in Chapter 2.3. The proofs for fixed design are in Chapter A.1.3. We discuss orthogonal design in Chapter A.1.3 and sub-Gaussian design in Chapter A.1.3. In particular, we use the two special designs to better understand the three assumptions and see how having a clean pool helps with the support recovery. We will call one-pool case the setting with only contaminated pool and call two-pool case the setting with both data pools.

Proofs of Theorem 2.6 and Theorem 2.7

Proof of Theorem 2.6. We follow the usual Primal Dual Witness argument for support recovery in linear regression, which contains the following steps Wainwright (2009):

1. Set $\widehat{\gamma}_{T^c} = 0$.

2. Solve the oracle subproblem for $(\widehat{\gamma}_T, \hat{z}_T)$:

$$\widehat{\gamma}_T \in \arg\min_{\gamma \in \mathbb{R}^t} \left\{ \frac{1}{2n} \|Ay' - B\gamma\|_2^2 + \lambda \|\gamma\|_1 \right\}, \tag{A.11}$$

   and choose $\hat{z}_T \in \partial \|\widehat{\gamma}_T\|_1$. In the one data pool case, we have $A = P^{\perp}_{X,\cdot T}$ and $B = P^{\perp}_{X,\cdot T}$; in the two data pool case, we have $A = P^{\perp}_{X',\cdot T}$ and $B = \bar{P}_T$.

3. Solve $\hat{z}_{T^c}$ via the zero-subgradient equation, and check whether the strict dual feasibility condition holds: $\|\widehat{z}_{T^c}\|_\infty < 1$.

As in the usual Lasso analysis Wainwright (2009), under the eigenvalue condition (2.6), $(\widehat{\gamma}_T, 0) \in \mathbb{R}^n$ is the unique optimal solution of the Lasso, where $\widehat{\gamma}_T$ is the solution obtained by solving the oracle subproblem (A.11).

The focus of our current analysis is to verify the conditions under which the strict dual feasibility condition holds. The KKT conditions for equation (2.5) may be rewritten as

$$\bar{P}_T^\top \bar{P}_T(\hat{\gamma}_T - \gamma_T^*) - \bar{P}_T^\top P^{\perp}_{X'} \epsilon' + n\lambda \hat{z}_T = 0, \tag{A.12}$$

$$\bar{P}_{T^c}^\top \bar{P}_T(\hat{\gamma}_T - \gamma_T^*) - \bar{P}_{T^c}^\top P^{\perp}_{X'} \epsilon' + n\lambda \hat{z}_{T^c} = 0, \tag{A.13}$$

where $\hat{z}_T \in \partial \|\widehat{\gamma}_T\|_1, \hat{z}_{T^c} \in \partial \|\widehat{\gamma}_{T^c}\|_1$.

We will use the following equations to simplify terms later:

$$\bar{P}_T^\top \bar{P}_T = (P_{X'}^{\perp\top} P_{X'}^{\perp})_{TT}, \quad \begin{pmatrix} \bar{P}_T^\top P_{X'}^{\perp} \epsilon' \\ \bar{P}_{T^c}^\top P_{X'}^{\perp} \epsilon' \end{pmatrix} = \bar{P}^\top P_{X'}^{\perp} \epsilon' = \bar{P}^\top \epsilon' = \begin{pmatrix} \bar{P}_T^\top \epsilon' \\ \bar{P}_{T^c}^\top \epsilon' \end{pmatrix}.$$

Since $\bar{P}_T^\top \bar{P}_T$ is invertible by condition (2.6), we can multiply equation (A.12) by $\left(\bar{P}_T^\top \bar{P}_T\right)^{-1}$ on the left to obtain

$$\hat{\gamma}_T - \gamma_T^* = (\bar{P}_T^\top \bar{P}_T)^{-1} \bar{P}_T^\top \epsilon' - n\lambda (\bar{P}_T^\top \bar{P}_T)^{-1} \hat{z}_T. \tag{A.14}$$

Plugging this into equation (A.13), we then obtain

$$\hat{z}_{T^c} = -\frac{1}{n\lambda} \bar{P}_{T^c}^\top \bar{P}_T \left[(\bar{P}_T^\top \bar{P}_T)^{-1} \bar{P}_T^\top \epsilon' - n\lambda (\bar{P}_T^\top \bar{P}_T)^{-1} \hat{z}_T \right] + \frac{1}{n\lambda} \bar{P}_{T^c}^\top \epsilon',$$

or

$$\hat{z}_{T^c} = \underbrace{\bar{P}_{T^c}^\top \bar{P}_T (\bar{P}_T^\top \bar{P}_T)^{-1} \hat{z}_T}_{\mu} + \underbrace{\bar{P}_{T^c}^\top \left(I - \bar{P}_T (\bar{P}_T^\top \bar{P}_T)^{-1} \bar{P}_T^\top \right) \frac{\epsilon'}{n\lambda}}_{V_{T^c}}. \tag{A.15}$$

We need to show that $\|\hat{z}_{T^c}\|_\infty < 1$.
Note that condition (2.7) gives us

$$\exists \alpha' \in [0, 1), \ \|\mu\|_\infty = \max_{j \in T^c} \|\bar{P}_j^\top \bar{P}_T (\bar{P}_T^\top \bar{P}_T)^{-1}\|_1 \leqslant \alpha'.$$

Furthermore, since

$$\lambda \geqslant \frac{1}{1-\alpha'} \left\| \bar{P}_{T^c}^\top \left(I - \bar{P}_T (\bar{P}_T^\top \bar{P}_T)^{-1} \bar{P}_T^\top \right) \frac{\epsilon'}{n} \right\|_\infty,$$

we have

$$\|V_{T^c}\|_\infty \leqslant \frac{1-\alpha'}{2}.$$

Combining these inequalities, we obtain strict dual feasibility:

$$\|\hat{z}_{T^c}\|_\infty \leqslant \|\mu\|_\infty + \|V_{T^c}\|_\infty < 1.$$

In addition, applying the triangle inequality to the RHS of equation (A.14), we obtain

$$G' = \|(\bar{P}_T^\top \bar{P}_T)^{-1}\bar{P}_T^\top \epsilon'\|_\infty + n\lambda\|(\bar{P}_T^\top \bar{P}_T)^{-1}\hat{z}_T\|_\infty \geqslant \|\hat{\gamma}_T - \gamma_T^*\|_\infty.$$

This concludes the proof. □

Proof of Theorem 2.7. Note that

$$\forall i \in T, \quad |\gamma_i^*| - |\hat{\gamma}_i| \leqslant \|\hat{\gamma}_T - \gamma_T^*\|_\infty \leqslant G',$$

where the last inequality uses Theorem 2.6. Thus, if condition (2.8) also holds, we have

$$\forall i \in T, \quad |\hat{\gamma}_i| \geqslant \min_{i \in T} |\gamma_i^*| - \|\hat{\gamma}_T - \gamma_T^*\|_\infty \geqslant \min_{i \in T} |\gamma_i^*| - G' > 0,$$

concluding the proof. □

Orthogonal design

Main results for orthogonal design We now focus on a special case, where our data have an orthogonal property. Let $X = \begin{bmatrix} RQ^\top \\ FQ^\top \end{bmatrix} \in \mathbb{R}^{(t+p)\times p}$, $\widetilde{X} = WQ^\top \in \mathbb{R}^{p\times p}$, where Q is an orthogonal matrix with columns $q_1, q_2, \cdots, q_p$, F, W are diagonal matrices with diagonals f_i's and w_i's separately ($i \in [p]$), and $R = \left[\begin{array}{cccc|c} r_1 & 0 & 0 & 0 & \\ 0 & r_2 & 0 & 0 & \\ 0 & 0 & \cdots & 0 & \mathbf{0}_{t\times(p-t)} \\ 0 & 0 & 0 & r_t & \end{array}\right]$. We assume for all $i \in [p]$, $r_i \neq 0, f_i \neq 0$. Consider the first t points are buggy and the rest p points are nonbuggy, i.e., $X_T = RQ^\top \in \mathbb{R}^{t\times p}, X_{T^c} = FQ^\top \in \mathbb{R}^{p\times p}$.

Applying Theorems 2.6 and 2.7, we obtain Propositions A.3 and A.4.

Proposition A.3. *In the one-pool case, suppose we choose*

$$\lambda \geqslant \frac{2\sigma}{n(1-\alpha)}\left(\sqrt{\log 2(n-t)} + C\right), \tag{A.16}$$

for some constant $C > 0$, *and*

$$\alpha = \max_{1\leqslant i\leqslant t}\left|\frac{r_i}{f_i}\right| < 1. \tag{A.17}$$

Then the contaminated pool is capable of achieving subset support recovery with probability at least $1 - e^{-\frac{C^2}{2}}$.

In the two-pool case, suppose we choose

$$\lambda \geqslant \frac{2\sigma}{n(1-\alpha')}\max\left\{1, \sqrt{\frac{\eta n}{mL}}\right\}\left(\sqrt{\log 2(n-t)} + C'\right), \tag{A.18}$$

for some constant $C' > 0$, *and*

$$\alpha' = \max_{1\leqslant i\leqslant t}\left|\frac{r_i f_i}{f_i^2 + \eta\frac{n}{m}w_i^2}\right| < 1. \tag{A.19}$$

Then adding clean points will achieve subset support recovery with probability at least $1 - e^{-\frac{C'^2}{2}}$.

As stated in Theorems 2.6 and 2.7, to ensure exact recovery, we also need to impose a gamma-min condition. This leads to the following proposition:

Proposition A.4. *In the one-pool case, suppose inequality* (A.17) *holds. If also*

$$\min_{1\leqslant i\leqslant t}|\gamma_i^*| > \sigma(\sqrt{2\log t}+c)\max_{1\leqslant i\leqslant t}\sqrt{1+\frac{r_i^2}{f_i^2}} + \frac{2\sigma}{1-\alpha}\left(\sqrt{\log 2(n-t)}+C\right)\left(1+\max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2}\right), \quad \text{(A.20)}$$

then there exists a λ *to achieve exact recovery, with probability at least* $1-2e^{-\frac{c^2}{2}}-e^{-\frac{C^2}{2}}$.

In the two-pool case, suppose $\eta \leqslant \frac{mL}{n}$*, and inequality* (A.19) *holds. If also*

$$\min_{1\leqslant i\leqslant t}|\gamma_i^*| \geqslant \sigma(\sqrt{2\log t}+c)\sqrt{1+\max_{1\leqslant i\leqslant t}\frac{r_i^2(Lf_i^2+\frac{\eta n}{m}w_i^2)}{L(f_i^2+\frac{\eta n}{m}w_i^2)^2}} + \frac{2\sigma}{1-\alpha'}\left(\sqrt{\log 2(n-t)}+C\right)\left(1+\max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2+\frac{\eta n}{m}w_i^2}\right), \quad \text{(A.21)}$$

then there exists a λ *to achieve exact recovery, with probability at least* $1-2e^{-\frac{c^2}{2}}-e^{-\frac{C^2}{2}}$.

Compare (A.17) and (A.19). Mutual incoherence is decreased from $\frac{r_i^2}{f_i^2}$ to $\frac{r_i^2}{f_i^2+\frac{\eta n}{m}w_i^2}$. Compare (A.20) and (A.21). The second max term, $\max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2} \geqslant \max_{1\leqslant i\leqslant t}\frac{r_i^2(Lf_i^2+\frac{\eta n}{m}w_i^2)}{L(f_i^2+\frac{\eta n}{m}w_i^2)^2}$, because

$$\max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2} \geqslant \max_{1\leqslant i\leqslant t}\frac{r_i^2(f_i^2+\frac{\eta n}{m}w_i^2)}{(f_i^2+\frac{\eta n}{m}w_i^2)^2} \geqslant \max_{1\leqslant i\leqslant t}\frac{r_i^2(Lf_i^2+\frac{\eta n}{m}w_i^2)}{L(f_i^2+\frac{\eta n}{m}w_i^2)^2}$$

when $L \geqslant 1$. Also note that $\frac{1}{1-\alpha} > \frac{1}{1-\alpha'}$. Altogether, the requirement of $\min_{i\in[t]}|\gamma_i^*|$ is weakened by introducing clean points. Thus, we see that the mutual incoherence improves in two-pool setting. The gamma-min condition imposes a lower bound of $\Omega\left(\sqrt{\log(n-t)}\right)$ on the signal-to-noise ratio, $\frac{\min_{i\in[t]}|\gamma_i^*|}{\sigma}$, and including second pool reduces the prefactor.

As can be seen, we want $|w_i|$ to be sufficiently large compared to $|f_i|$.

However, if $|w_i|$ is bounded, we may instead ensure support recovery by repeating points. In this section, we discuss the effect of repeating points and determine the number of points needed to guarantee correct support recovery. Suppose

$$W = \begin{bmatrix} \vec{w}_1 & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{w}_2 & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \cdots & \vec{w}_p \end{bmatrix},$$

where $\vec{w}_i = [w_{i1}, \dots, w_{il_i}]^\top$. For the i^{th} direction q_i, we have k_i repeated points with respective weights $w_{i1}, w_{i2}, \dots, w_{il_i}$.

Proposition A.5. *Suppose the scale of clean data points is bounded by* w_B*. Using* $w_{i1}, \dots, w_{il_i}$*, where* $l_i = \left\lceil \left(\frac{|w_i|}{w_B}\right)^2 \right\rceil$ *and* $|w_{ij}| = w_B$, $\forall j \in [l_i]$*, achieves the same effect on Conditions 2.3, 2.4, and 2.5 as adding a single point with scale* w_i*.*

From Proposition A.5, we see that to correctly identify the bugs, we can also query multiple points in the same direction if the leverage of a single additional point is not large enough.

Proofs for orthogonal design We will first simplify the three conditions, and then provide the proofs of Propositions A.3, A.4, and A.5.

In the one-pool case, we have

$$\begin{aligned} P^\perp_{X,TT} &= I_{t\times t} - X_T(X^\top X)^{-1}X_T^\top \\ &= I_{t\times t} - R(R^\top R + F^\top F)^{-1}R^\top \\ &= \mathrm{diag}\left(\frac{f_1^2}{r_1^2 + f_1^2}, \cdots, \frac{f_t^2}{r_t^2 + f_t^2}\right). \end{aligned}$$

Note that $P^\perp_{X,TT}$ is a diagonal matrix. Thus, the eigenvalues are immediately obtained and

$$\lambda_{\min}(P^{\perp}_{X,TT}) = \min_{1\leqslant i\leqslant t} \frac{f_i^2}{r_i^2+f_i^2} = \min_{1\leqslant i\leqslant t} \frac{1}{\left(\frac{r_i}{f_i}\right)^2+1} = \frac{1}{\max_{1\leqslant i\leqslant t}\left(\frac{r_i}{f_i}\right)^2+1}.$$

The condition that $P^{\perp}_{X,TT}$ is invertible is therefore equivalent to the condition that $f_i \neq 0$ for all i. Assuming this is true, we have

$$\begin{aligned} P^{\perp}_{X,T^cT}(P^{\perp}_{X,TT})^{-1} &= -F(R^\top R + F^\top F)^{-1}R^\top \cdot (I_{t\times t} - R(R^\top R + F^\top F)^{-1}R^\top)^{-1} \\ &= \begin{bmatrix} diag\left(-\frac{r_1}{f_1}, \cdots, -\frac{r_t}{f_t}\right)_{t\times t} \\ \mathbf{0}_{(p-t)\times t} \end{bmatrix}. \end{aligned}$$

The mutual incoherence condition can then be written in terms of the quantity

$$\left\|P^{\perp}_{X,T^cT}(P^{\perp}_{X,TT})^{-1}\right\|_\infty = \max_{1\leqslant i\leqslant t}\left|\frac{r_i}{f_i}\right| = \max_{1\leqslant i\leqslant t}\left|\frac{r_i f_i}{f_i^2}\right|.$$

Note that the mutual incoherence condition also implies that $f_i \neq 0$, $\forall i$, since the mutual incoherence parameter will otherwise go to infinity.
The remaining condition is the gamma-min condition. Note that the upper bound on the ℓ_∞-error of γ consists of two parts:

$$\|\widehat{\gamma} - \gamma^*\|_\infty \leqslant \|(P^{\perp}_{X,TT})^{-1}(P^{\perp}_{X,T\cdot})\epsilon\|_\infty + n\lambda\left\|(P^{\perp}_{X,TT})^{-1}\right\|_\infty.$$

Regarding $P^{\perp}_{X,T\cdot}$ as two blocks, $\left(P^{\perp}_{X,TT},\ P^{\perp}_{X,TT^c}\right)$, we have

$$\|(P^{\perp}_{X,TT})^{-1}(P^{\perp}_{X,T\cdot})\epsilon\|_\infty = \left\|\left(I \quad (P^{\perp}_{X,TT})^{-1}P^{\perp}_{X,TT^c}\right)\epsilon\right\|_\infty.$$

Altogether, we see that

$$G = \max_{1\leqslant i\leqslant t}\left|\epsilon_i - \frac{r_i}{f_i}\epsilon_{i+t}\right| + n\lambda\left(\max_{1\leqslant i\leqslant t}\left\{\frac{r_i^2}{f_i^2}\right\}+1\right).$$

To summarize, the minimum eigenvalue condition becomes

$$\lambda_{\min}(P^{\perp}_{X,TT}) = \frac{1}{\max_{1\leqslant i\leqslant t}\left(\frac{r_i}{f_i}\right)^2 + 1} > 0; \tag{A.22a}$$

the mutual incoherence condition becomes

$$\left\|P^{\perp}_{X,T^cT}(P^{\perp}_{X,TT})^{-1}\right\|_\infty = \max_{1\leqslant i\leqslant t}\left|\frac{r_i}{f_i}\right| = \alpha \in [0,1); \tag{A.22b}$$

and the gamma-min condition becomes

$$\min_{1\leqslant i\leqslant t}|\gamma_i^*| \geqslant G = \max_{1\leqslant i\leqslant t}|\epsilon_i - \frac{r_i}{f_i}\epsilon_{i+t}| + n\lambda\left(\max_{1\leqslant i\leqslant t}\left\{\frac{r_i^2}{f_i^2}\right\} + 1\right). \tag{A.22c}$$

Similar calculations show that in the two-pool case, the minimum eigenvalue condition becomes

$$\lambda_{\min}(P^{\perp}_{X',TT}) = \min_{1\leqslant i\leqslant t}\frac{f_i^2 + \frac{\eta n}{m}w_i^2}{r_i^2 + f_i^2 + \frac{\eta n}{m}w_i^2} = \frac{1}{\max_{i\in[t]}\frac{r_i^2}{f_i^2+\frac{\eta n}{m}w_i^2} + 1} > 0; \tag{A.23a}$$

the mutual incoherence condition becomes

$$\left\|P^{\perp}_{X',T^cT}(P^{\perp}_{X',TT})^{-1}\right\|_\infty = \max_{1\leqslant i\leqslant t}\left|\frac{r_i f_i}{f_i^2 + \frac{\eta n}{m}w_i^2}\right| = \alpha' \in [0,1); \tag{A.23b}$$

and the gamma-min condition becomes

$$\min_{1\leqslant i\leqslant t}|\gamma_i^*| \geqslant G', \tag{A.23c}$$

where

$$G' = \max_{1\leqslant i\leqslant t}\left|\epsilon_i - \frac{r_i f_i}{f_i^2 + \frac{\eta n}{m}w_i^2}\epsilon_{i+t} - \frac{\sqrt{\frac{\eta n}{m}}r_i w_i}{f_i^2 + \frac{\eta n}{m}w_i^2}\widetilde{\epsilon}_i\right| \\ + n\lambda\left(\max_{1\leqslant i\leqslant t}\left\{\frac{r_i^2}{f_i^2 + \frac{\eta n}{m}w_i^2}\right\} + 1\right).$$

Here is the proof of Proposition A.3.

Proof of Proposition A.3. According to Theorem 2.6, the subset support recovery result relies on two conditions: the minimum eigenvalue condition and the mutual incoherence condition. In the orthogonal design case, we will argue that both inequalities (A.22a) and (A.23a) hold in the one-pool case, and inequlaity (A.19) is sufficient for both inequalities (A.23a) and (A.23b) in the two-pool case.

For the one-pool case, the assumption (A.17) implies that $f_i \neq 0, \ ,\forall i \in [t]$. Note that the minimum eigenvalue condition (A.22a) is equivalent to $f_i \neq 0, \ ,\forall i \in [t]$. Hence, the minimum eigenvalue condition holds. Furthermore, the mutual incoherence condition (A.23a) clearly holds.

For the two-pool case, if $f_i = 0$ for some $i \in [t]$, then plugging into (A.19) implies that $w_i^2 > 0$. Thus, f_i and w_i cannot be zero at the same time, implying that the eigenvalue condition (A.23a) holds. Note that inequality (A.19) is equivalent to inequlaity (A.23b).

The remaining of the argument concerns the choice of λ. Note that Theorem 2.6 requires λ to be lower-bounded for subset recovery (see inequality (2.9)). Taking the two-pool case as an example, we will show that when inequality (A.18) holds, inequality (2.9) holds with high probability. Define

$$Z_j = \bar{P}_{\cdot j}^\top\left(I - \bar{P}_T(\bar{P}_T^\top\bar{P}_T)^{-1}\bar{P}_T^\top\right)\frac{\epsilon'}{n}, \quad j \in T^c.$$

Note that $\left\|\bar{P}_{\cdot j}^{\top}\left(I - \bar{P}_{T}(\bar{P}_{T}^{\top}\bar{P}_{T})^{-1}\bar{P}_{T}^{\top}\right)\right\|_2 \leqslant 1$ for all $j \in T^c$, and $\epsilon' = \begin{pmatrix} \epsilon \\ \sqrt{\frac{\eta n}{m}}\widetilde{\epsilon} \end{pmatrix}$ has i.i.d. sub-Gaussian entries with parameter at most $\max\{1, \frac{\eta n}{mL}\}\sigma^2$. Thus, Z_j is sub-Gaussian with parameter at most $\max\{1, \frac{\eta n}{mL}\}\frac{\sigma^2}{n^2}$. By a sub-Gaussian tail bound (cf. Lemma A.6), we then have

$$\mathbb{P}\left(\max_{j \in T^c} |Z_j| \geqslant \delta_0\right) \leqslant 2(n-t)\exp\left(-\frac{n^2\delta_0^2}{2\max\{1, \frac{\eta n}{mL}\}\sigma^2}\right).$$

Let C' be a constant such that

$$2(n-t)\exp\left(-\frac{n^2\delta_0^2}{2\max\{1, \frac{\eta n}{mL}\}\sigma^2}\right) = \exp\left(-\frac{C'^2}{2}\right),$$

and define

$$\delta_0 := \frac{\sigma}{n}\max\{1, \sqrt{\frac{\eta n}{mL}}\}\sqrt{\log 2(n-t) + C'^2}.$$

Note that we want

$$\frac{2\max_{j \in T^c}|Z_j|}{1-\alpha'} \leqslant \lambda,$$

which therefore occurs with probability at least $1 - e^{-\frac{C'^2}{2}}$ when

$$\lambda \geqslant \frac{2\sigma}{n(1-\alpha')}\max\{1, \sqrt{\frac{\eta n}{mL}}\}\left(\sqrt{\log 2(n-t)} + C'\right) \geqslant \frac{2\delta_0}{1-\alpha'}.$$

The proof for the one-pool case is similar, so we omit the details. □

Here is the proof of Proposition A.4.

Proof of Proposition A.4. To simplify notation, define

$$u_i := \epsilon_i - \frac{r_i}{f_i}\epsilon_{i+t},$$
$$v_i := \epsilon_i - \frac{r_i f_i}{f_i^2 + \frac{\eta n}{m} w_i^2}\epsilon_{i+t} - \frac{\sqrt{\frac{\eta n}{m}} r_i w_i}{f_i^2 + \frac{\eta n}{m} w_i^2}\widetilde{\epsilon}_i.$$

Note that u_i is σ_{u_i}-sub-Gaussian and v_i is σ_{v_i}-sub-Gaussian, with variance parameters

$$\sigma_{u_i} = \sqrt{1 + \frac{r_i^2}{f_i^2}}\sigma, \quad \sigma_{v_i} = \sqrt{1 + \frac{r_i^2(L^2 f_i^2 + \frac{\eta n}{m} w_i^2)}{L^2(f_i^2 + \frac{\eta n}{m} w_i^2)^2}}\sigma.$$

We now prove two technical lemmas:

Lemma A.6 (Concentration for non-identical sub-Gaussian random variables)**.** *Suppose* $\{u_i\}_{i=1}^t$ *are* σ_{u_i}*-sub-Gaussian random variables and* $\{v_i\}_{i=1}^t$ *are* σ_{v_i}*-sub-Gaussian random variables. Then the following inequalities hold:*

$$P\left(\max_{1 \leqslant i \leqslant t} |u_i| > \delta_1\right) \leqslant 2t \exp\left(-\frac{\delta_1^2}{2\max_{1 \leqslant i \leqslant t} \sigma_{u_i}^2}\right), \tag{A.24}$$

$$P\left(\max_{1 \leqslant i \leqslant t} |v_i| > \delta_1\right) \leqslant 2t \exp\left(-\frac{\delta_1^2}{2\max_{1 \leqslant i \leqslant t} \sigma_{v_i}^2}\right). \tag{A.25}$$

Proof. Note that

$$\max_{1 \leqslant i \leqslant t} |u_i| = \max_{1 \leqslant i \leqslant 2t} u_i,$$

where $u_{t+i} \coloneqq -u_i$, for $1 \leqslant i \leqslant t$. By a union bound, we have

$$\begin{aligned} P\left(\max_{1\leqslant i\leqslant t} |u_i| > \delta_1\right) &= P\left(\bigcup_{1\leqslant i\leqslant 2t} \{u_i > \delta_1\}\right) \\ &\leqslant \sum_{1\leqslant i\leqslant 2t} P\left(u_i \geqslant \delta_1\right) \\ &= \sum_{1\leqslant i\leqslant t} P\left(u_i \geqslant \delta_1\right) + \sum_{1\leqslant i\leqslant t} P\left(u_{t+i} \geqslant \delta_1\right) \\ &= \sum_{1\leqslant i\leqslant t} P\left(u_i \geqslant \delta_1\right) + \sum_{1\leqslant i\leqslant t} P\left(u_i \leqslant -\delta_1\right). \end{aligned}$$

For each u_i, we have the tail bounds

$$P(u_i > \delta_1) \leqslant \exp\left(-\frac{\delta_1^2}{2\sigma_{u_i}^2}\right), \quad P(u_i < -\delta_1) \leqslant \exp\left(-\frac{\delta_1^2}{2\sigma_{u_i}^2}\right).$$

Altogether, we see that

$$P\left(\max_{1\leqslant i\leqslant t} |u_i| > \delta_1\right) \leqslant 2\sum_{1\leqslant i\leqslant t} \exp\left(-\frac{\delta_1^2}{2\sigma_{u_i}^2}\right) \leqslant 2t\exp\left(-\frac{\delta_1^2}{2\max_{1\leqslant i\leqslant t}\sigma_{u_i}^2}\right).$$

Similarly, we may obtain the desired concentration inequality for the ν_i's:

$$P\left(\max_{1\leqslant i\leqslant t} |\nu_i| > \delta_1\right) \leqslant 2t\exp\left(-\frac{\delta_1^2}{2\max_{1\leqslant i\leqslant t}\sigma_{\nu_i}^2}\right).$$

□

Lemma A.7. *In the one-pool case, under the orthogonal design setting, suppose*

$$\min_{1\leqslant i\leqslant t} |\gamma_i^*| > (\sqrt{2}\sqrt{\log t} + c_1)\max_{1\leqslant i\leqslant t}\sigma_{u_i} + n\lambda\left(1 + \max_{1\leqslant i\leqslant t}\frac{r_i^2}{f_i^2}\right), \qquad \text{(A.26)}$$

where $\sigma_{u_i} = \sqrt{1 + \frac{r_i^2}{f_i^2}}\sigma$. *Then the gamma-min condition holds with probability* *at least* $1-2e^{-c_1^2/2}$.

In the two-pool case, suppose

$$\min_{1\leqslant i\leqslant t}|\gamma_i^*| > (\sqrt{2}\sqrt{\log t}+c_2)\max_{1\leqslant i\leqslant t}\sigma_{\nu_i}+n\lambda\left(1+\max_{i\in[t]}\frac{r_i^2}{f_i^2+\frac{\eta n}{m}w_i^2}\right), \tag{A.27}$$

where $\sigma_{\nu_i}=\sqrt{1+\frac{r_i^2(L^2f_i^2+\frac{\eta n}{m}w_i^2)}{L^2(f_i^2+\frac{\eta n}{m}w_i^2)^2}}\sigma$. *Then the gamma-min condition holds with probability at least* $1-2e^{-c_2^2/2}$.

We use inequality (A.24) in Lemma A.6. Let

$$\delta_1=\sqrt{2\log t+c_1^2}\max_{1\leqslant i\leqslant t}\sigma_{u_i},$$

where $c_1\in(0,+\infty)$. Then with probability $1-2e^{-\frac{c_1^2}{2}}$, the following holds:

$$\max_{1\leqslant i\leqslant t}|u_i|\leqslant\sqrt{2\log t+c_1^2}\max_{1\leqslant i\leqslant t}\sigma_{u_i}\leqslant(\sqrt{2\log t}+c_1)\max_{1\leqslant i\leqslant t}\sigma_{u_i}.$$

In inequality (A.25), take $\delta_2=\sqrt{2\log t+c_2^2}\max_{1\leqslant i\leqslant t}\sigma_{u_i}$ where $c_2\in(0,+\infty)$. Then with probability $1-2e^{-\frac{c_2^2}{2}}$, the following holds:

$$\max_{1\leqslant i\leqslant t}|\nu_i|\leqslant\sqrt{2\log t+c_2^2}\max_{1\leqslant i\leqslant t}\sigma_{\nu_i}\leqslant(\sqrt{2\log t}+c_2)\max_{1\leqslant i\leqslant t}\sigma_{\nu_i}.$$

Combining these inequalities with conditions (A.22c) and (A.23c), we obtain $G\leqslant\min_{i\in[t]}|\gamma_i^*|$ with probability at least $1-2e^{-\frac{c_1^2}{2}}$ or at least $1-2e^{-\frac{c_2^2}{2}}$. Specifically, when we choose $c_1=c_2=2.72$, we can achieve a probability guarantee of at least 95% for the two statements.

Therefore, Proposition A.4 is proved by plugging the results from Lemma A.6 into Lemma A.7. □

Here is the proof of Proposition A.5.

Proof of Proposition A.5. We will prove the proposition by comparing the three conditions in the two situations: adding one clean point and repeating multiple clean points. The conditions for adding one clean point are already provided in inequalities (A.23a), (A.23b) and (A.23c) above.

We now provide the conditions for repeating multiple clean points. The minimum eigenvalue condition becomes

$$\lambda_{\min}(P^{\perp}_{X',TT}) = \min_{1\leqslant i\leqslant t} \frac{f_i^2 + \sum_{j=1}^{l_i} w_{ij}^2}{r_i^2 + f_i^2 + \frac{\eta n}{m}\sum_{j=1}^{l_i} w_{ij}^2} = \frac{1}{\max_{1\leqslant i\leqslant t} \frac{r_i^2}{f_i^2+\sum_{j=1}^{l_i} w_{ij}^2} + 1}; \tag{A.28a}$$

the mutual incoherence condition becomes

$$\left\| P^{\perp}_{X',T^cT}(P^{\perp}_{X',TT})^{-1} \right\|_\infty = \max_{1\leqslant i\leqslant t} \left| \frac{r_i f_i}{f_i^2 + \frac{\eta n}{m}\sum_{j=1}^{l_i} w_{ij}^2} \right|; \tag{A.28b}$$

and the gamma-min condition becomes

$$\begin{aligned} \|\widehat{\gamma} - \gamma^*\|_\infty \leqslant \max_{1\leqslant i\leqslant t} \Bigg| \epsilon_i &+ \frac{r_i f_i}{f_i^2 + \frac{\eta n}{m}\sum_{j=1}^{l_i} w_{ij}^2}\epsilon_{i+t} \\ &+ \sum_{j=1}^{k_i} \frac{r_i w_{ij}}{f_i^2 + \frac{\eta n}{m}\sum_{j=1}^{l_i} w_{ij}^2} \frac{\epsilon_{i+t+p+j}}{L} \Bigg| \\ &+ n\lambda \left(\max_{1\leqslant i\leqslant t}\left\{ \frac{r_i^2}{f_i^2 + \frac{\eta n}{m}\sum_{j=1}^{l_i} w_{ij}^2}\right\} + 1 \right). \end{aligned} \tag{A.28c}$$

Compared with inequlaities (A.23a), (A.23b) and (A.23c), conditions (A.28a), (A.28b) and (A.28c) replace w_i^2 by $\sum_{j=1}^{l_i} w_{ij}^2$. Suppose the scale of the clean data points is bounded by w_B. Then adding one data point may not be enough to satisfy the three conditions. Thus, to achieve the same effect of a large scaled $|w_i|$ in inequalities (A.23a), (A.23b) and (A.23c), we need the number of repeated clean points to be at least $\left(\frac{|w_i|}{w_B}\right)^2$. □

Sub-Gaussian design

In this section, we will present the support recovery results for sub-Gaussian design in Proposition A.8 and Proposition A.9, and the comparisons of the three conditions in the one- and two-pool cases in Table A.1. Later, we will provide the proofs of the propositions.

Main results for sub-Gausian design

Proposition A.8. *Suppose* $\{x_j\}_{j\in T^c}$ *and* $\{\widetilde{x}_i\}_{i\in[m]}$*, are i.i.d. sub-Gaussian with parameter* σ_x^2 *and covariance matrix* $\Sigma \succ 0$*. Further assume that* $\|X_T\|_2 \leqslant B_T$*. For the one-pool case, suppose we choose* λ *to satisfy inequality* (A.16) *and the sample size satisfies*

$$n > t + \max\Big\{p + C_1, \frac{4c_1^2\sigma_x^4(p+C_1)\|\Sigma\|_2^2}{\lambda_{\min}^2(\Sigma)}, \sqrt{t}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_2^2(\log n + \sqrt{p\log n})\right)\left(1 + \frac{2c_1\sigma_x^2\|\Sigma\|_2}{\lambda_{\min}(\Sigma)}\right)\frac{B_T}{\lambda_{\min}(\Sigma)}\Big\}, \tag{A.29}$$

then the contaminated pool achieves subset support recovery with probability at least $1 - e^{-\frac{C^2}{2}} - 2e^{-C_1} - n^{-(c_2-1)}$.

For the two-pool case, assume we choose λ *to satisfy* (A.18) *and the sample sizes satisfy*

$$n > \max\Big\{t + m, \frac{t}{1+\eta} + \frac{\sqrt{t}}{1+\eta}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_2^2(\log n + \sqrt{p\log n})\right)\left(1 + \frac{2c_1\sigma_x^2\|\Sigma\|_2}{\lambda_{\min}(\Sigma)}\right)\frac{B_T}{\lambda_{\min}(\Sigma)}\Big\} \tag{A.30}$$

and

$$m \geqslant \max\{1, 4c_1^2\sigma_x^4\|\Sigma\|_2^2\}(p + C_1').$$

Then adding clean points achieves subset support recovery with probability at least $1 - e^{-\frac{C'^2}{2}} - 2e^{-C_1'} - n^{-(c_2-1)}$.

As seen in Proposition A.8, the number of data points n may be reduced by $1+\eta$ with the introduction of a second data pool. Note that when T is randomly chosen from $[n]$, we have $B_T = O(\sqrt{t}\|\Sigma\|_2)$, so inequalities (A.29) and (A.30) require $\frac{t}{n}$ to be upper-bounded, and adding a second pool may weaker the upper bound to be $(1+\eta)$ than the upper bound for one-pool case.

We now present a result concerning exact support recovery:

Proposition A.9. *In the one-pool case, suppose inequality* (A.29) *holds. If*

$$\min_{i \in T} |\gamma_i^*| \geqslant \frac{1}{b_{\min}} \left(2\sigma\sqrt{\log t + c} + \frac{2\sigma\sqrt{t}}{(1-\alpha)} \left(\sqrt{\log 2(n-t)} + C \right) \right), \tag{A.31}$$

then there exists a λ *to achieve exact recovery with probability at least* $1 - 2e^{-c} - e^{-\frac{C^2}{2}} - 2e^{-C_1} - n^{-C_2}$.

For the two-pool case, suppose the assumptions in Proposition A.8 hold, and

$$\min_{i \in T} |\gamma_i^*| \geqslant \frac{1}{b'_{\min}} \Big(2\sigma\sqrt{\log t + c} + \frac{2\sigma\sqrt{t}}{(1-\alpha')} \max\{1, \sqrt{\frac{\eta n}{mL}}\} \left(\sqrt{\log 2(n-t)} + C' \right) \Big). \tag{A.32}$$

Then there exists a λ *to achieve exact recovery with probability at least* $1 - 2e^{-c} - e^{\frac{-C'^2}{2}} - 2e^{-C_1'} - n^{-C_2}$.

Compared to Proposition A.8, Proposition A.9 additionally requires the “signal-to-noise" ratio to be large enough. We can show that $b_{\min} \leqslant b'_{\min}$;

thus, for an appropriate choice of η, the lower bound (A.31) is smaller than the bound (A.32), so the gamma-min condition is improved.

We now briefly compare the three conditions for the one- and two-pool cases in the random design setting.

Table A.1: Comparison between the two cases

Condition	One-pool case	Two-pool case
Eigenvalue	$\lambda_{\min}\left(P^{\perp}_{X,TT}\right) = b_{\min}$	$\lambda_{\min}\left(P^{\perp}_{X',TT}\right) = b'_{\min} \geqslant b_{\min}$
Mutual incoherence	$\\| - X_{T^c}((n-t)\Sigma)^{-1}X_T^\top\\|_\infty$	$\frac{\\|-X_{T^c}((n-t)\Sigma)^{-1}X_T^\top\\|_\infty}{1+\eta\frac{n}{n-t}}$
Gamma-min	$\min_i \lvert\gamma_i^*\rvert \geqslant \frac{2\sigma\sqrt{\log t}+n\lambda\sqrt{t}}{b_{\min}}$	$\min_i \lvert\gamma_i^*\rvert \geqslant \frac{2\sigma\sqrt{\log t}+n\lambda\sqrt{t}}{b'_{\min}}$

In general, the eigenvalue condition is improved by adding a second pool. The mutual incoherence condition is improved in the two-pool case with large m by a constant multiplier $\frac{1}{1+\eta\frac{n}{m}}$ $(\leqslant 1)$, and the gamma-min condition lower bound is improved by a constant $\frac{b_{\min}}{b'_{\min}}$ $(\leqslant 1)$.

For the **eigenvalue condition**, the key result is that adding clean data points will not hurt, i.e., it makes the minimum eigenvalue smaller. A formal statement is provided in Proposition A.10. Recall that

$$P^{\perp}_{X',TT} = I - X'_T(X'^\top X')^{-1}X'_T,$$
$$P^{\perp}_{X,TT} = I - X_T(X^\top X)^{-1}X_T^\top,$$

where $X' = \begin{pmatrix} X \\ \sqrt{\frac{\eta n}{m}}\widetilde{X} \end{pmatrix}$, and we assume that $X^\top X$ is invertible.

Proposition A.10 (Comparison of minimum eigenvalue conditions)**.** *We have*

$$\lambda_{\min}(P^{\perp}_{X',TT}) \geqslant \lambda_{\min}(P^{\perp}_{X,TT}).$$

Note that the result of Proposition A.10 does not require any assumptions on $\widetilde{X}$ or η. However, the degree of improvement depends on η, as seen in the proof. Usually when n is small, increasing η leads to a big jump of the minimum eigenvalue; when n is large, increasing η does not change the minimum eigenvalue much. A typical relationship between η and $\lambda_{\min}\left(P^{\perp}_{X',TT}\right)$ can be seen in Figure A.1.

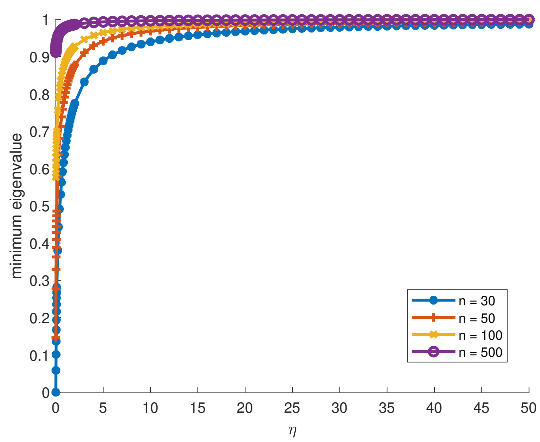


Figure A.1: Influence of η on the Minimum Eigenvalue Condition. The x-axis is the weight parameter η and the y-axis is $\lambda_{\min}(P^{\perp}_{X',TT})$. We take $t = 15, p = 20$, and $m = 5$, and vary n from 30 to 500. Both pools are drawn randomly from $\mathcal{N}(\mathbf{0}, \mathbf{I}_p)$.

For **mutual incoherence** condition, it is possible to find settings for small m that make the mutual incoherence condition worse. Consider the following example:

Example A.11 (Example where the mutual incoherence condition worsens). *Suppose*

$$X_T = \begin{bmatrix} -1.8271 & -1.6954 & -1.1000 \\ 0.3020 & -1.4817 & -0.2284 \end{bmatrix},$$

$$X_{T^c} = \begin{bmatrix} -1.7680 & -0.0863 & 1.6822 \\ -0.5750 & -1.1013 & 0.4749 \\ -0.6693 & -0.6413 & 0.6126 \\ -0.3271 & 0.3060 & -1.0068 \\ 0.6177 & 0.3941 & -2.6407 \\ -0.7001 & 2.3465 & 0.4309 \end{bmatrix},$$

$$\widetilde{X} = \begin{bmatrix} -1.8722 & 0.5154 & 0.1560 \\ -0.9036 & 0.6064 & -0.2540 \end{bmatrix}.$$

Then

$$\|P^{\perp}_{X,T^cT}(P^{\perp}_{X,TT})^{-1}\|_{\infty} = 0.96 < 1 < \|P^{\perp}_{X',T^cT}(P^{\perp}_{X',TT})^{-1}\|_{\infty} = 1.28.$$

Despite this negative example, we can show that including a second pool helps when m is large compared to p. Recalling the assumption that $X_{T^c}^{\top} X_{T^c}$ is invertible, we can write

$$\begin{aligned}
&P^{\perp}_{X,T^cT}(P^{\perp}_{X,TT})^{-1} \\
&= -X_{T^c}\left(X_T^{\top}X_T + X_{T^c}^{\top}X_{T^c}\right)^{-1} X_T^{\top}\left(I - X_T\left(X_T^{\top}X_T + X_{T^c}^{\top}X_{T^c}\right)^{-1} X_T^{\top}\right)^{-1} \\
&= -X_{T^c}\left(X_T^{\top}X_T + X_{T^c}^{\top}X_{T^c}\right)^{-1} X_T^{\top}\left(I + X_T\left(X_{T^c}^{\top}X_{T^c}\right)^{-1} X_T^{\top}\right) \\
&= -X_{T^c}\left(X_{T^c}^{\top}X_{T^c}\right)^{-1}\left(X_T^{\top}X_T(X_{T^c}^{\top}X_{T^c})^{-1} + I\right)^{-1}\left(I + X_T^{\top}X_T\left(X_{T^c}^{\top}X_{T^c}\right)^{-1}\right) X_T^{\top} \\
&= -X_{T^c}(X_{T^c}^{\top}X_{T^c})^{-1}X_T^{\top}.
\end{aligned} \tag{A.33}$$

The first equality uses the definitions of $P^{\perp}_{X,T^cT}$ and $P^{\perp}_{X,TT}$, the second equality uses the Woodbury matrix identity Henderson and Searle (1981), and the third equality follows from simple linear algebraic manipulations.

Similarly, we can simplify the mutual incoherence condition for the two-pool case, by replacing $X_{T^c}^\top X_{T^c}$ with $X_{T^c}^\top X_{T^c} + \eta \frac{n}{m} \widetilde{X}^\top \widetilde{X}$ in the inverse:

$$P^{\perp}_{X',T^cT}(P^{\perp}_{X',TT})^{-1} = -X_{T^c}\left(X_{T^c}^\top X_{T^c} + \eta \frac{n}{m}\widetilde{X}^\top \widetilde{X}\right)^{-1} X_T^\top, \tag{A.34}$$

where we know that $X_{T^c}^\top X_{T^c} + \eta \frac{n}{m} \widetilde{X}^\top \widetilde{X}$ must be invertible since $X_{T^c}^\top X_{T^c}$ is invertible.

Given these simplifications, it is easy to see that the difference between these two terms lies in the middle inverses. When m is large, we have $(X_{T^c}^\top X_{T^c})^{-1} \approx ((n-t)\Sigma)^{-1}$ and $\left(X_{T^c}^\top X_{T^c} + \eta \frac{n}{m} \widetilde{X}^\top \widetilde{X}\right)^{-1} \approx ((n - t + \eta n)\,\Sigma)^{-1}$, where Σ is the covariance matrix for the common distribution of X_{T^c} and $\widetilde{X}$. Therefore, the mutual incoherence parameter in the one-pool case is approximately equal to the mutual incoherence in the two-pool case scaled by $\left(1 + \eta \frac{n}{n-t}\right)^{-1}$, which immediately implies that adding a second data pool improves the mutual incoherence condition. This is stated formally in the following proposition:

Proposition A.12 (Comparison of mutual incoherence conditions)**.** *Let* $B_T = O(\sqrt{t})$. *In the one-pool case, if* $n \geqslant t + \frac{c_1^2 \sigma_x^4 (p + C_1)\|\Sigma\|^2}{\lambda^2_{\min}(\Sigma)}$, *then*

$$\left| \left\| X_{T^c} \frac{\Theta}{n-t} X_T^\top \right\|_\infty - \left\| X_{T^c}\left(X_{T^c}^\top X_{T^c}\right)^{-1} X_T^\top \right\|_\infty \right| = O\left(t(n-t)^{-1}(\sqrt{p} + \sqrt{\log n})\right),$$

with high probability.

In the two-pool case, if

$$n \geqslant t + \max\left\{\frac{c_1^2 \sigma_x^4 \|\Sigma\|^2}{\lambda^2_{\min}(\Sigma)}, 1\right\} m$$

and

$$m \geq \max\{1, c_1^2 \sigma_x^4 (p + C_1') \|\Sigma\|_2^2\},$$

then

$$\left| \left\| X_{T^c} \frac{\Theta}{n - t + \eta n} X_T^\top \right\|_\infty - \left\| X_{T^c} \left(X_{T^c}^\top X_{T^c} + \frac{\eta n}{m} \widetilde{X}^\top \widetilde{X} \right)^{-1} X_T^\top \right\|_\infty \right|$$
$$= O\left(t(n - t + \eta n)^{-1} (\sqrt{p} + \sqrt{\log n}) \right),$$

with high probability.

Proposition A.12 states that when m and n are sufficiently large, the one-pool mutual incoherence parameter is close to $\frac{\left\| X_{T^c} \Theta X_T^\top \right\|_\infty}{n-t}$ and the two-pool mutual incoherence parameter is close to $\frac{\left\| X_{T^c} \Theta X_T^\top \right\|_\infty}{n-t+\eta n}$. Since the second expression has a larger denominator, the mutual incoherence condition improves with the introduction of a second data pool with parameter $\eta > 0$.

For **gamma-min** condition, we need to compare the terms G and G'. Note that inequalities (A.31) and (A.32) are equivalent to lower-bounding the "signal-to-noise" ratio. The order of the lower bound for two-pool case is as same as the one-pool case, i.e., $\frac{\min_i |\gamma_i^*|}{\sigma} \geq O(\sqrt{t \log n})$. However, adding a second pool improves the constant by having a factor of $\frac{1}{b'_{\min}}$ instead of $\frac{1}{b_{\min}}$. As established in Proposition A.10, we have $b_{\min} \leq b'_{\min}$. Therefore, the lower bound in the two-pool case is smaller than the lower bound in the one-pool case.

Note that the **weight parameter** η shows up in all the three conditions. However, recall that the mutual incoherence condition is not always improved by adding a second pool, unless m is sufficiently large. Therefore, an appropriate conclusion is that once we have a large clean data pool, it is reasonable to place arbitrarily large weight on the second pool. On the other hand, if we have fewer clean data points, we cannot be as confident about the estimator obtained using the second pool alone. For example, in the orthogonal design, if we obtain clean points in the non-buggy subspace, the mutual incoherence condition is not improved no matter how large we make η. In addition, the gamma-min condition involves the randomness from noise, and in order to control the sparsity of γ, we need the regularizer λ to match large η (cf. inequality (A.18)). Based on inequality (A.32), we need the "signal-to-noise" ratio, i.e., $\frac{n\lambda\sqrt{t}}{\sigma}$, to be sufficient large. If η is too large, we cannot estimate relatively small components of γ^*. In summary, selecting η too large or too small is not wise: If η is too small, we do not improve the three conditions, whereas if η is too large, the range of controllable "signal-to-noise" ratios decays.

Proofs for sub-Gaussian design Now we provide proofs of sub-Gaussian design. Here is the proof of Proposition A.8.

Proof of Proposition A.8. We prove the results for the one- and two-pool cases sequentially. In each case, we begin with background calculations, and then analyze the eigenvalue condition followed by the mutual incoherence condition.

For the one-pool case, we know that λ satisfies inequality (A.16) with probability at least $1 - e^{-\frac{C^2}{2}}$.

Note that x_j, $j \in T^c$ are sub-Gaussian random vectors with parameter σ_x. By Theorem 4.7.1 and Exercise 4.7.3 in Vershynin Vershynin (2018) and our assumption of n, we have

$$\left\| \Sigma - \frac{X_{T^c}^\top X_{T^c}}{n-t} \right\|_2 \leqslant c_1 \sigma_x^2 \sqrt{\frac{p + C_1}{n-t}} \|\Sigma\|_2, \tag{A.35}$$

with probability at least $1 - e^{-C_1}$. We will later use this bound multiple times to establish the eigenvalue condition and the mutual incoherence condition.

We first consider the eigenvalue condition. By the dual Weyl's inequality Horn and Johnson (1994), we have $\lambda_{\min}(A + B) \geqslant \lambda_{\min}(A) + \lambda_{\min}(B)$

for any square matrices A and B. Then

$$\begin{aligned}\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right) &= \lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t} - \Sigma + \Sigma\right) \\ &\geqslant \lambda_{\min}(\Sigma) + \lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t} - \Sigma\right) \\ &\geqslant \lambda_{\min}(\Sigma) - \left\|\frac{X_{T^c}^\top X_{T^c}}{n-t} - \Sigma\right\|_2,\end{aligned}$$

where the second inequality follows from the fact that $\lambda_{\min}(A) \leqslant \lambda_{\max}(A)$ for any square matrix A. Combining this with inequality (A.35) and taking $n \geqslant t + 4\frac{c_1^2\sigma_x^4(p+C_1)\|\Sigma\|_2^2}{\lambda_{\min}^2(\Sigma)}$ by assumption (A.29), we have that

$$\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right) \geqslant \lambda_{\min}(\Sigma) - c_1\sigma_x^2\sqrt{\frac{p+C_1}{n-t}}\|\Sigma\|_2 \geqslant \frac{1}{2}\lambda_{\min}(\Sigma) > 0, \quad \text{(A.36)}$$

with probability $1 - e^{-C_1}$. We now derive the following result:

Lemma A.13. *Suppose $X_{T^c}^\top X_{T^c}$ is invertible, where $X_T \in \mathbb{R}^{t\times p}$ and $X_{T^c} \in \mathbb{R}^{(n-t)\times p}$. Then*

$$\lambda_{\min}\left(P_{X,TT}^\perp\right) \geqslant 1 - \frac{\lambda_{\max}(X_T^\top X_T)}{\lambda_{\max}(X_T^\top X_T) + \lambda_{\min}(X_{T^c}^\top X_{T^c})} > 0,$$

implying that the eigenvalue condition for the one-pool case holds.

Proof. Define $C = Q(I + Q^\top Q)^{-1}Q^\top$ and $Q \in \mathbb{R}^{s\times p}$, and suppose rank$(Q)$ $= r$. Let $Q = USV^\top$ be the SVD, where $U \in \mathbb{R}^{t\times p}, V \in \mathbb{R}^{p\times p}$, and $S = \begin{bmatrix} J_{r\times r} & 0_{r\times(p-r)} \\ 0_{(t-r)\times r} & 0_{(t-r)\times(p-r)}\end{bmatrix}$. Here, J is a diagonal matrix of positive singular values. Then

$$\begin{aligned} C &= USV^\top(I + VS^\top SV^\top)^{-1}VS^\top U^\top \\ &= US(I + S^\top S)^{-1}S^\top U^\top \\ &= U\begin{bmatrix} J_{r\times r} & 0_{r\times(p-r)} \\ 0_{(t-r)\times r} & 0_{(t-r)\times(p-r)} \end{bmatrix} \cdot \begin{bmatrix} (I+J^2)^{-1}_{r\times r} & 0_{r\times(p-r)} \\ 0_{(t-r)\times r} & I_{(p-r)\times(p-r)} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} J_{r\times r} & 0_{r\times(p-r)} \\ 0_{(t-r)\times r} & 0_{(t-r)\times(p-r)} \end{bmatrix} U^\top \\ &= U\begin{bmatrix} (J(I+J^2)^{-1}J)_{r\times r} & 0_{r\times(p-r)} \\ 0_{(t-r)\times r} & 0_{(p-r)\times(p-r)} \end{bmatrix} U^\top. \end{aligned} \tag{A.37}$$

Therefore, $\lambda_{\max}(C) = \frac{a^2_{\max}}{1+a^2_{\max}}$, where $a_{\max}$ is the maximum singular value appearing in J. Also note that $a^2_{\max}$ is the maximum eigenvalue of $Q^\top Q$.

Following (16.51) in Seber Seber (2008), given $X^\top_{T^c}X_{T^c}$ is invertible, there exists a non-singular matrix A such that $AX^\top_{T^c}X_{T^c}A^\top = I$ and $AX^\top_T X_T A^\top = D$, where D is diagonal matrix.

Note that

$$\begin{aligned} X_T(X_T^\top X_T + X_{T^c}X_{T^c})^{-1}X_T^\top &= X_T A^\top(A(X_T^\top X_T + X_{T^c}X_{T^c})A^\top)^{-1}AX_T^\top \\ &= X_T A^\top(AX_T^\top X_T A^\top + I)AX_T^\top \\ &= Q(Q^\top Q + I)^{-1}Q^\top, \end{aligned}$$

where $Q := X_T A^\top$.

Based on our earlier arguments, we know that the matrix under consideration has maximum eigenvalue $\frac{\lambda_{\max}(AX_T^\top X_T A^\top)}{1+\lambda_{\max}(AX_T^\top X_T A^\top)}$. Since $AX_T^\top X_T A^\top$ is similar to $X_T^\top X_T A^\top A$, we have$\lambda_{\max}(AX_T^\top X_T A^\top) = \lambda_{\max}(X_T^\top X_T A^\top A)$. Furthermore, we have $A^\top A = (X_{T^c}^\top X_{T^c})^{-1}$, implying that

$$\begin{aligned}\lambda_{\max}(AX_T^\top X_T A^\top) &= \lambda_{\max}(X_T^\top X_T (X_{T^c}^\top X_{T^c})^{-1}) \\ &\leqslant \max_v \frac{\left\|X_T^\top X_T (X_{T^c}^\top X_{T^c})^{-1} v\right\|_2^2}{\|(X_{T^c}^\top X_{T^c})^{-1} v\|_2^2} \cdot \max_v \frac{\left\|(X_{T^c}^\top X_{T^c})^{-1} v\right\|_2^2}{\|v\|_2^2} \\ &\leqslant \frac{\lambda_{\max}(X_T^\top X_T)}{\lambda_{\min}(X_{T^c}^\top X_{T^c})}.\end{aligned}$$

Altogether, we have

$$\begin{aligned}\lambda_{\max}\left(X_T \left(X_T^\top X_T + X_{T^c}^\top X_{T^c}\right)^{-1} X_T^\top\right) &\leqslant \frac{1}{1 + \lambda_{\max}^{-1}\left(X_T^\top X_T (X_{T^c}^\top X_{T^c})^{-1}\right)} \\ &\leqslant \frac{1}{1 + \frac{\lambda_{\min}(X_{T^c}^\top X_{T^c})}{\lambda_{\max}(X_T^\top X_T)}}.\end{aligned} \tag{A.38}$$

Finally, we may conclude that

$$\begin{aligned}\lambda_{\min}\left(P_{X,TT}^\perp\right) &= \lambda_{\min}\left(I - X_T \left(X_T^\top X_T + X_{T^c}^\top X_{T^c}\right)^{-1} X_T^\top\right) \\ &= 1 - \lambda_{\max}\left(X_T \left(X_T^\top X_T + X_{T^c}^\top X_{T^c}\right)^{-1} X_T^\top\right) \\ &\geqslant 1 - \frac{1}{1 + \frac{\lambda_{\min}(X_{T^c}^\top X_{T^c})}{\lambda_{\max}(X_T^\top X_T)}} \\ &= 1 - \frac{\lambda_{\max}(X_T^\top X_T)}{\lambda_{\max}(X_T^\top X_T) + \lambda_{\min}(X_{T^c}^\top X_{T^c})}.\end{aligned}$$

Since $\lambda_{\min}(X_{T^c}^\top X_{T^c}) > 0$, we have $\lambda_{\min}\left(P_{X,TT}^\perp\right) < 1$, implying the desired result. □

We now consider the mutual incoherence condition. By the triangle inequality, we have

$$\frac{1}{n-t}\left\|X_{T^c}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}X_T\right\|_\infty \leqslant \underbrace{\frac{1}{n-t}\left\|X_{T^c}\Theta X_T^\top - X_{T^c}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}X_T^\top\right\|_\infty}_{①} + \underbrace{\frac{1}{n-t}\left\|X_{T^c}\Theta X_T^\top\right\|_\infty}_{②}.$$

We bound ① and ② separately. Note that

$$\begin{aligned} ① &= \frac{\max_{j\in T^c}\left\|x_j^\top\left(\Theta - \left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}\right)X_T^\top\right\|_1}{n-t} \\ &\leqslant \frac{\sqrt{t}}{n-t}\max_{j\in T^c}\|x_j\|_2\left\|\Theta-\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}\right\|_2\left\|X_T^\top\right\|_2. \end{aligned}$$

In order to bound ①, we bound three parts separately. By assumption, we have $\left\|X_T^\top\right\|_2 \leqslant B_T$. For $\max_{j\in T^c}\|x_j\|_2$, we leverage the Hanson-Wright inequality (Theorem 6.2.1 in Vershynin (2018)) and a union bound. By the Hanson-Wright inequality, we see that for $t > 0$,

$$P\left(\|x_j\|_2^2 - \mathbb{E}[\|x_j\|_2^2] \geqslant t\right) \leqslant \exp\left\{-c\min\left(\frac{t^2}{\sigma_x^4 p}, \frac{t}{\sigma_x^2}\right)\right\},$$

where c is an absolute constant.

By a union bound, we then have

$$P\left(\max_{j\in T^c}\|x_j\|_2 \geqslant \sqrt{\mathbb{E}[\|x_j\|_2^2] + \Delta}\right) = P\left(\max_{j\in T^c}\|x_j\|_2^2 \geqslant \mathbb{E}[\|x_j\|_2^2] + \Delta\right)$$
$$\leqslant \sum_{j\in T^c} P\left(\|x_j\|_2^2 \geqslant \mathbb{E}[\|x_j\|_2^2] + \Delta\right)$$
$$\leqslant (n-t)\exp\left\{-c\min\left(\frac{\Delta^2}{\sigma_x^4 p}, \frac{\Delta}{\sigma_x^2}\right)\right\}.$$

Setting $\Delta = c_2\sigma_x^2\max\{\sqrt{p\log n}, \log n\}$ with $c_2 \geqslant 1$ so that we have $\min\left\{\frac{\Delta^2}{\sigma_x^4 p}, \frac{\Delta}{\sigma_x^2}\right\} \geqslant c_2 \log n$, we conclude that

$$\begin{aligned}\max_{j\in T^c}\|x_j\|_2 &\leqslant \sqrt{\mathbb{E}[\|x_j\|_2^2] + \Delta}\\ &\leqslant \sqrt{\mathrm{trace}(\Sigma)} + \Delta\\ &\leqslant \sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p\log n}),\end{aligned} \tag{A.39}$$

with probability at least $1 - n^{-(c_2-1)}$, where $c_2 \geqslant \max\{2, 2/c\}$.

To bound $\left\|\Theta - \left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}\right\|_2$, note that for two matrices A and B, we have

$$\left\|A^{-1} - B^{-1}\right\|_2 \leqslant \frac{\|A-B\|_2}{\lambda_{\min}(A)\lambda_{\min}(B)}.$$

Combining this fact with inequalities (A.35) and (A.36), we obtain

$$\begin{aligned}\left\|\Theta - \left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)^{-1}\right\|_2 &\leqslant \frac{\left\|\Sigma - \frac{X_{T^c}^\top X_{T^c}}{n-t}\right\|_2}{\lambda_{\min}(\Sigma)\,\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}}{n-t}\right)} \leqslant \frac{2\left\|\Sigma - \frac{X_{T^c}^\top X_{T^c}}{n-t}\right\|_2}{\lambda_{\min}^2(\Sigma)}\\ &\leqslant \frac{2c_1\sigma_x^2\sqrt{\frac{p+C_1}{n-t}}\|\Sigma\|_2}{\lambda_{\min}^2(\Sigma)}.\end{aligned} \tag{A.40}$$

Altogether, we obtain the bound

$$\textcircled{1} \leqslant \frac{\sqrt{t}}{n-t}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p\log n})\right)\cdot \frac{2c_1\sigma_x^2\sqrt{\frac{p+C_1}{n-t}}\|\Sigma\|}{\lambda_{\min}^2(\Sigma)}B_T. \tag{A.41}$$

We now consider ②. Note that

$$\begin{aligned}
\frac{\left\|X_{T^c}\Theta X_T^\top\right\|_\infty}{n-t} &= \frac{1}{n-t}\max_{j\in T^c}\|x_j^\top\Theta X_T^\top\|_1 \\
&\leqslant \frac{\sqrt{t}}{n-t}\max_{j\in T^c}\|x_j^\top\|_2\|\Theta\|_2\|X_T^\top\|_2 \\
&= \frac{\sqrt{t}}{n-t}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p\log n})\right)\cdot\frac{1}{\lambda_{\min}(\Sigma)}B_T.
\end{aligned} \tag{A.42}$$

Therefore,

$$\begin{aligned}
\textcircled{1} + \textcircled{2} \leqslant \frac{\sqrt{t}}{n-t}&\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p\log n})\right) \\
&\cdot\left(1 + \frac{2c_1\sigma_x^2\sqrt{\frac{p+C_1}{n-t}}\|\Sigma\|_2}{\lambda_{\min}(\Sigma)}\right)\frac{B_T}{\lambda_{\min}(\Sigma)}.
\end{aligned}$$

Finally, assuming n satisfies the bound (A.29), and taking a union bound over all the probabilistic statements appearing above, we conclude that the mutual incoherence condition holds with probability at least $1 - e^{-\frac{C^2}{2}} - 2e^{-C_1} - n^{-(c_2-1)}$. This concludes the proof.

For the two-pool case, we will use the following inequalities:

$$\begin{aligned}
\left\|\Sigma - \frac{X_{T^c}^\top X_{T^c}}{n-t}\right\|_2 &\leqslant c_1\sigma_x^2\sqrt{\frac{p+C_1'}{n-t}}\|\Sigma\|_2, \\
\left\|\Sigma - \frac{\widetilde{X}^\top\widetilde{X}}{m}\right\|_2 &\leqslant c_1\sigma_x^2\sqrt{\frac{p+C_1'}{m}}\|\Sigma\|_2,
\end{aligned}$$

with probablity at least $1-2e^{-C_1'}$. Combining these inequalities and using the triangle inequality, we obtain

$$\begin{aligned}
&\left\| \Sigma - \frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n} \right\|_2 \\
&\leqslant \frac{n-t}{n-t+\eta n}\left\| \Sigma - \frac{X_{T^c}^\top X_{T^c}}{n-t} \right\|_2 + \frac{\eta n}{n-t+\eta n}\left\| \Sigma - \frac{\widetilde{X}^\top \widetilde{X}}{m} \right\|_2 \\
&\leqslant c_1 \sigma_x^2 \|\Sigma\|_2 \frac{n-t}{n-t+\eta n}\sqrt{\frac{p+C_1'}{n-t}} + c_1 \sigma_x^2 \|\Sigma\|_2 \frac{\eta n}{n-t+\eta n}\sqrt{\frac{p+C_1'}{m}} \\
&\overset{n \geqslant t+m}{\leqslant} 2c_1 \sigma_x^2 \|\Sigma\|_2 \sqrt{\frac{p+C_1'}{m}},
\end{aligned} \tag{A.43}$$

with probability at least $1-2e^{-C_1'}$.

Analogous to Lemma A.13, we can conclude that if $X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}$ is invertible, the eigenvalue condition satisfies

$$\lambda_{\min}(P_{X',TT}^{\perp}) \geqslant 1 - \frac{\lambda_{\max}(X_T^\top X_T)}{\lambda_{\max}(X_T^\top X_T) + \lambda_{\min}\left(X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}\right)} > 0.$$

(This can be proved just by replacing $X_{T^c}^\top X_{T^c}$ with $X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}$ in the proof of Lemma A.13.) However, since we further wish to bound the minimum eigenvalue from below by $\lambda_{\min}(\Sigma)/2$, to match the one-pool case and to be used in the proof for the mutual incoherence condition later, we will consider $X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}$ directly.

Note that

$$\begin{aligned}
\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}\right) &= \lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}-\Sigma+\Sigma\right)\\
&\geqslant \lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}-\Sigma\right)+\lambda_{\min}(\Sigma)\\
&\geqslant \lambda_{\min}(\Sigma)-\left\|\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}-\Sigma\right\|_2.
\end{aligned}$$

Thus, if we choose $m \geqslant 4c_1^2\sigma_x^4(p+C_1')\|\Sigma\|_2^2$, we have

$$\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}\right) \geqslant \frac{1}{2}\lambda_{\min}(\Sigma) > 0,$$

with probability at least $1-2e^{-C_1'}$.

We now consider the mutual incoherence condition. Similar to the derivation of inequality (A.40), we have that

$$\begin{aligned}
\left\|\Theta-\left(\frac{X_{T^c}^\top X_{T^c}+\frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}}{n-t+\eta n}\right)^{-1}\right\|_2 &\leqslant \frac{\left\|\Sigma-\frac{X_{T^c}^\top X_{T^c}+\eta\frac{n}{m}\widetilde{X}^\top\widetilde{X}}{(1+\eta)n-t}\right\|_2}{\lambda_{\min}(\Sigma)\,\lambda_{\min}\left(\frac{X_{T^c}^\top X_{T^c}+\eta\frac{n}{m}\widetilde{X}^\top\widetilde{X}}{(1+\eta)n-t}\right)}\\
&\leqslant 2c_1\sigma_x^2\frac{\|\Sigma\|_2}{\lambda_{\min}^2(\Sigma)}\sqrt{\frac{p+C_1'}{m}}.
\end{aligned}$$

Combining this with inequality (A.39), we obtain

$$\begin{aligned}
&\frac{\left\| X_{T^c}\Theta X_T^\top - X_{T^c}\left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n}\right)^{-1} X_T^\top \right\|_\infty}{n-t+\eta n} \\
&= \frac{\max_{j \in T^c} \left\| x_j^\top \left(\Theta - \left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n}\right)^{-1}\right) X_T^\top \right\|_1}{n-t+\eta n} \\
&\leqslant \frac{\sqrt{t}}{n-t+\eta n} \max_{j \in T^c} \|x_j\|_2 \cdot \left\| \Theta - \left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n}\right)^{-1} \right\|_2 \|X_T^\top\|_2 \\
&\leqslant \frac{\sqrt{t}}{n-t+\eta n}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p \log n})\right) \\
&\qquad \cdot 2c_1\sigma_x^2 \frac{\|\Sigma\|_2}{\lambda_{\min}^2(\Sigma)} \sqrt{\frac{p + C_1'}{m}} B_T.
\end{aligned}$$

Therefore, together with the triangle inequality and inequality (A.42), we can bound the mutual incoherence parameter as follows:

$$\begin{aligned}
&\frac{\left\| X_{T^c}\left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n}\right)^{-1} X_T^\top \right\|_\infty}{n-t+\eta n} \\
&\leqslant \frac{\left\| X_{T^c}\Theta X_T^\top - X_{T^c}\left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m}\widetilde{X}^\top \widetilde{X}}{n-t+\eta n}\right)^{-1} X_T^\top \right\|_\infty}{n-t+\eta n} + \frac{\|X_{T^c}\Theta X_T^\top\|_\infty}{n-t+\eta n} \\
&\leqslant \frac{\sqrt{t}}{n-t+\eta n}\left(\sqrt{p\|\Sigma\|_2} + c_2\sigma_x^2(\log n + \sqrt{p \log n})\right) \\
&\qquad \cdot \left(1 + 2c_1\sigma_x^2 \frac{\|\Sigma\|_2}{\lambda_{\min}(\Sigma)} \sqrt{\frac{p + C_1'}{m}}\right) \frac{B_T}{\lambda_{\min}(\Sigma)}.
\end{aligned}$$

By the assumption on n in inequality (A.30), the mutual incoherence condition therefore holds with probability $1 - e^{-\frac{C'^2}{2}} - 2e^{-C_1'} - n^{-(c_2-1)}$. □

Here is the proof of Proposition A.9.

Proof of Proposition A.9. To achieve exact support recovery, we need all the three conditions to hold. The eigenvalue condition and the mutual incoherence condition have already been discussed in the analysis of subset support recovery in Appendix A.8, so it remains to analyze the gamma-min condition.

Recall that

$$G' = \|(P^{\perp}_{X',TT})^{-1}P^{\perp}_{X',T\cdot}\epsilon'\|_{\infty} + n\lambda \left\|(P^{\perp}_{X',TT})^{-1}\right\|_{\infty}.$$

To simplify notation, we define

$$A := \|(P^{\perp}_{X',TT})^{-1}P^{\perp}_{X',T\cdot}P^{\perp}_{X'}\epsilon'\|_{\infty}, \quad B := n\lambda \left\|(P^{\perp}_{X',TT})^{-1}\right\|_{\infty}.$$

We also define the random variables

$$Z_i := e_i^{\top}(P^{\perp}_{X',TT})^{-1}P^{\perp}_{X',T\cdot}P^{\perp}_{X'}\epsilon'.$$

Since $P^{\perp}_{X'}$ is a projection matrix and the maximum singular value of $P^{\perp}_{X',T\cdot}$ is smaller than the maximum singular value of $P^{\perp}_{X'}$'s, we have

$$\left\|(P^{\perp}_{X',TT})^{-1}P^{\perp}_{X',T\cdot}P^{\perp}_{X'}\right\|_2 \leqslant \left\|(P^{\perp}_{X',TT})^{-1}\right\|_2 \leqslant \left\|(P^{\perp}_{X',TT})^{-1}\right\|_2 \leqslant \frac{1}{b'_{\min}},$$

for all $i \in T$. Note that Z_i is a zero-mean sub-Gaussian random variable with parameter at most $\frac{\sigma}{b'_{\min}}$. By a sub-Gaussian tail bound, we then have

$$P\left(\max_{1\leqslant i\leqslant t} |Z_i| > \frac{\sigma}{b'_{\min}}\left(\sqrt{2\log t} + \Delta\right)\right) \leqslant 2e^{-\frac{\Delta^2}{2}}.$$

Therefore, with probability at least $1 - 2e^{-c}$, we have $A \leqslant \frac{2\sigma\sqrt{\log t + c}}{b'_{\min}}$. Note that $\|(P^{\perp}_{X',TT})^{-1}\|_{\infty} \leqslant \sqrt{t}\|(P^{\perp}_{X',TT})^{-1}\|_2 = \frac{\sqrt{t}}{b'_{\min}}$. We can then immediately obtain the bound $B \leqslant \frac{2n\lambda\sqrt{t}}{b'_{\min}}$.

Combined with the fact that

$$\lambda \geqslant \frac{2\sigma}{n(1-\alpha')} \max\left\{1, \sqrt{\frac{\eta n}{mL}}\right\}\left(\sqrt{\log 2(n-t)} + C'\right),$$

we then obtain

$$G' \leqslant \frac{1}{b'_{\min}}\left[2\sigma\sqrt{\log t + c} + \frac{2\sigma\sqrt{t}}{1-\alpha'}\max\{1, \sqrt{\frac{\eta n}{mL}}\}(\sqrt{\log 2(n-t)} + C')\right].$$

Thus, as long as $\min_{i\in T}|\gamma_i^*|$ is greater than or equal to the RHS of the inequality above, the gamma-min condition holds with probability at least $1 - 2e^{-c} - e^{-\frac{C'^2}{2}}$. Consequently, the exact support recovery is achieved.

The proof of the one-pool case is similar as the proof of the two-pool case provided above, so we omit the details here. □

Here is the proof of Proposition A.10

Proof. Proof of Proposition A.10

By the Sherman-Morrison-Woodbury formula Henderson and Searle (1981), we have

$$\begin{aligned} &X_T\left(X^\top X + \frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}\right)^{-1} X_T^\top \\ &= X_T\left(X^\top X\right)^{-1} X_T^\top \\ &\quad - \frac{\eta n}{m} X_T\left(X^\top X\right)^{-1}\widetilde{X}^\top (I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top)^{-1}\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top. \end{aligned} \tag{A.44}$$

We now state and prove two useful lemmas:

Lemma A.14. *Assume $X^\top X$ is invertible. Define*

$$A \coloneqq X_T\left(X^\top X\right)^{-1}\widetilde{X}^\top (I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top)^{-1}\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top.$$

Then $\lambda_{\min}(A) \geqslant 0$. Equality holds when $\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top$ is not full-rank.

Proof. First note that since $X^\top X$ is invertible and $\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top \succ 0$, the matrix $I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top$ is invertible. Note that

$$\forall y \in \mathbb{R}^t \neq 0, \quad y^\top A y \geqslant 0,$$

so the minimum eigenvalue of A is nonnegative.

In order to study when the $\lambda_{\min} = 0$, let $z = \widetilde{X}\left(X^\top X\right)^{-1} X_T^\top y$. When $y \neq 0$ and $\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top$ is full-rank, we have $z \neq 0$. Thus, if $\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top$ is full-rank, we have $\lambda_{\min}(A) > 0$. When $y \neq 0$ and $\widetilde{X}\left(X^\top X\right)^{-1} X_T^\top$ is not full-rank, there exists $y \neq 0$ such that $z = 0$, which causes $y^\top A y = 0$ and $\lambda_{\min}(A) = 0$. □

Lemma A.15. *The following equations holds:*

$$\lambda_{\min}(P^{\perp}_{X,TT}) = 1 - \lambda_{\max}(X_T\left(X^\top X\right)^{-1} X_T^\top),$$

$$\lambda_{\min}(P^{\perp}_{X',TT}) = 1 - \lambda_{\max}(X_T\left(X^\top X + \frac{\eta n}{m}\widetilde{X}\widetilde{X}^\top\right)^{-1} X_T^\top).$$

Proof. Since $X_T\left(X^\top X\right)^{-1} X_T^\top$ is symmetric positive semidefinite, we can write $X_T\left(X^\top X\right)^{-1} X_T^\top = Q\Lambda Q^\top$, where Q is an orthogonal matrix and Λ is a diagonal matrix with nonnegative diagonals. Then

$$I - X_T\left(X^\top X\right)^{-1} X_T^\top = Q(I - \Lambda)Q^\top.$$

Furthermore, we have shown in inequality (A.38) that

$$\lambda_{\max}\left(X_T(X^\top X)^{-1}X_T^\top\right) \leqslant \frac{1}{1 + \frac{\lambda_{\min}(X_{T^c}^\top X_{T^c})}{\lambda_{\max}(X_T^\top X_T)}}.$$

Hence, the maximum diagonal in Λ is upper-bounded by 1, and $I - \Lambda$ has all diagonal entries in the range $[0, 1]$. Thus, we have shown that

$\min \operatorname{diag}(I - \Lambda) = \max(\operatorname{diag}(\Lambda))$, implying the conclusion of the lemma. $\square$

Returning to the proof of the proposition, we have

$$\begin{aligned}
&\lambda_{\max}\left(X_T\left(X^\top X + \frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}\right)^{-1}X_T^\top\right)\\
&\quad\leqslant \lambda_{\max}\left(X_T\left(X^\top X\right)^{-1}X_T^\top\right)\\
&\qquad - \frac{\eta n}{m}\lambda_{\min}\left((X_T\left(X^\top X\right)^{-1}\widetilde{X}^\top(I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top)^{-1}\widetilde{X}\left(X^\top X\right)^{-1}X_T^\top\right)\\
&\quad\overset{(i)}{\leqslant} \lambda_{\max}\left(X_T\left(X^\top X\right)^{-1}X_T^\top\right),
\end{aligned}$$

Here, (i) comes from the fact that

$$\lambda_{\min}\left(X_T\left(X^\top X\right)^{-1}\widetilde{X}^\top \cdot (I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top)^{-1}\widetilde{X}\left(X^\top X\right)^{-1}X_T^\top\right) \geqslant 0,$$

which follows from Lemma A.14. Furthermore, by Lemma A.15, we have

$$\lambda_{\min}\left(P^{\perp}_{X',TT}\right) = 1 - \lambda_{\max}\left(X_T\left(X^\top X + \frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}\right)^{-1}X_T^\top\right)$$

and

$$\lambda_{\min}\left(P^{\perp}_{X',TT}\right) = 1 - \lambda_{\max}\left(X_T\left(X^\top X + \frac{\eta n}{m}\widetilde{X}^\top\widetilde{X}\right)^{-1}X_T^\top\right).$$

Altogether, we conclude that the minimum eigenvalue is at least improved by
$\frac{\eta n}{m}\lambda_{\min}\left(X_T\left(X^\top X\right)^{-1}\widetilde{X}^\top(I + \frac{\eta n}{m}\widetilde{X}(X^\top X)^{-1}\widetilde{X}^\top)^{-1}\widetilde{X}\left(X^\top X\right)^{-1}X_T^\top\right)$. $\square$

Here is the proof of Proposition A.12.

Proof of Proposition A.12. The proof leverages arguments from the proof of Proposition A.8. The goal is to argue that when n and m are sufficiently large, the empirical quantities are close to their population-level versions. We will use Big-O notation to simplify our discussion.

As already stated in inequality (A.41), if $n \geqslant t + \frac{c_1^2 \sigma_x^4 \|\Sigma\|^2}{\lambda_{\min}^2(\Sigma)}(p + C_1)$, then

$$\frac{\left\| X_{T^c} \Theta X_T^\top - X_{T^c} \left(\frac{X_{T^c}^\top X_{T^c}}{n-t} \right)^{-1} X_T^\top \right\|_\infty}{n-t} \leqslant \frac{\sqrt{t}}{n-t} \left(\sqrt{p\|\Sigma\|_2} + c_2 \sigma_x^2 (\log n + \sqrt{p \log n}) \right) \cdot \frac{2c_1 \sigma_x^2 \sqrt{\frac{p+C_1}{n-t}} \|\Sigma\|}{\lambda_{\min}^2(\Sigma)} B_T.,$$

with probability at least $1 - e^{-C_1} - n^{-1}$, where $c_2 > \max\{2, 2/c\}$.

Also for the two-pool case, if $n \geqslant t + \max\left\{ \frac{c_1^2 \sigma_x^4 \|\Sigma\|^2}{\lambda_{\min}^2(\Sigma)}, 1 \right\} m$ and $m \geqslant \max\{1, c_1^2 \sigma_x^4 (p + C_1') \|\Sigma\|_2^2\}$, we have

$$\begin{aligned} &\frac{\left\| X_{T^c} \Theta X_T^\top - X_{T^c} \left(\frac{X_{T^c}^\top X_{T^c} + \frac{\eta n}{m} \widetilde{X}^\top \widetilde{X}}{n-t+\eta n} \right)^{-1} X_T^\top \right\|_\infty}{n-t+\eta n} \\ &\leqslant \frac{\sqrt{t}}{n-t+\eta n} \left(\sqrt{p\|\Sigma\|_2} + c_2 \sigma_x^2 (\log n + \sqrt{p \log n}) \right) \\ &\quad \cdot \left(1 + 2c_1 \sigma_x^2 \frac{\|\Sigma\|_2}{\lambda_{\min}(\Sigma)} \sqrt{\frac{p + C_1'}{m}} \right) \frac{B_T}{\lambda_{\min}(\Sigma)}, \end{aligned}$$

with probability at least $1 - 2e^{-C_1'} - n^{-1}$, where c_2 is defined in the same way as above. Noting that $B_T \propto \sqrt{t}$ and using the triangle inequality, we conclude the proof. □

A.1.4 Proofs for Chapter 2.4

In this section, we provide proofs and additional details for the results in Chapter 2.4. We will establish several auxiliary results in the process, which are stated and proved in Appendix A.1.4. The flow of logic is outlined below:

Theorem 2.12 $\Leftarrow$ (Lemma A.18, Lemma A.24);
Lemma A.18 $\Leftarrow$ Theorem 2.7;
Lemma A.24 $\Leftarrow$ (Lemma A.19, Lemma A.23);
Lemma A.23 $\Leftarrow$ (Lemma A.20, Lemma A.21);
Lemma A.21 $\Leftarrow$ Lemma A.19.
Corollary 2.13 $\Leftarrow$ (Theorem 2.12, Corollary A.17).

We sometimes write $\widehat{\gamma}(\lambda)$ to represent the estimator from Lasso-based debugging with tuning parameter λ.

Proof of Theorem 2.12

We will first argue that the algorithm will stop, and then argue that all bugs are identified correctly when the algorithm stops. Finally, we will take a union bound over all the iterations in the while loop to obtain a probabilistic conclusion.

Algorithm 1 stops: Note that if we have an iteration k such that $\widehat{\lambda}^k > 2\lambda^*$ and $C = 0$, then the algorithm must stop after at most $\lfloor \log_2 \frac{\lambda^u}{\lambda^*} \rfloor$ iterations. Otherwise, we know that $C = 1$ for all iterations k such that $\widehat{\lambda}^k \geqslant \lambda^*$. Thus, after $k = \lfloor \log_2 \frac{\lambda^u}{\lambda^*} \rfloor$ iterations, we have

$$\lambda^k = \frac{\lambda^u}{2^{\lfloor \log_2 \frac{\lambda^u}{\lambda^*} \rfloor}} \in \left[\frac{\lambda^u}{2^{\log_2 \frac{\lambda^u}{\lambda^*}}}, \frac{\lambda^u}{2^{\log_2 \frac{\lambda^u}{\lambda^*} - 1}} \right] = [\lambda^*, 2\lambda^*].$$

As established in Lemma A.18, we know that all true bugs will be identified with such a value of λ^k, so the remaining points are $(X^{(k)}, y^{(k)}) = (X_{T^c}, y_{T^c})$. Also note that

$$\|P^{\perp}_{X_{T^c}} y_{T^c}\|_\infty = \|P^{\perp}_{X_{T^c}} (X_{T^c}\beta^* + \epsilon_{T^c})\|_\infty = \|P^{\perp}_{X_{T^c}} \epsilon_{T^c}\|_\infty.$$

Hence, by Lemma A.24, we have

$$\|P^{\perp}_{X_{T^c}} \epsilon_{T^c}\|_\infty < \frac{5}{2}\frac{1}{\bar{c}}\sqrt{\log 2n}\,\widehat{\sigma}.$$

Therefore, the stopping criteria takes effect and the algorithm stops.

Algorithm 1 correctly identifies all bugs: A byproduct of the preceding argument is that $\widehat{\lambda} > \lambda^*$. By Theorem 2.6, we have $\text{supp}(\widehat{\gamma}^k) \subseteq \text{supp}(\gamma^*)$. Now suppose we are at a stage where l of the t bugs are flagged, where $l \in \{0, 1, \ldots, t\}$.

If $l = t$, then $\bar{X} = X_{T^c}$. As argued preveiously, the algorithm stops with high probability. Hence, we output all of the bugs.

Otherwise, we have $l \leqslant t-1$. Suppose this happens at the k^{th} iteration. Then at least one bug remains in $(X^{(k)}, y^{(k)})$, and all the clean points are included. Let S denote the corresponding row indices of X and let γ_S^* denote the following subvector of γ^*. Since bugs still remain, we must have $\min_{i \in S} |\gamma_{S,i}^*| \geqslant \min_{i \in T} |\gamma_i^*|$. Furthermore,

$$\|P_{X^{(k)}}^{\perp} y^{(k)}\|_\infty = \|P_{X^{(k)}}^{\perp}(X^{(k)}\beta^* + \gamma_S^* + \epsilon_S)\|_\infty = \|P_{X^{(k)}}^{\perp}(\gamma_S^* + \epsilon_S)\|_\infty.$$

By Lemma A.24, we have

$$\|P_{X^{(k)}}^{\perp}(\gamma_S^* + \epsilon_S)\|_\infty > \frac{5}{2}\frac{1}{\bar{c}}\sqrt{\log 2n}\,\widehat{\sigma},$$

implying that $C = 0$. Thus, the procedure proceeds to the $(k+1)^{\text{st}}$ iteration. If for all k such that $\widehat{\lambda}^k \geqslant 2\lambda^*$, bugs still remain, then $\widehat{\lambda}^k$ keeps shrinking until the $\lfloor \log_2 \frac{\lambda^u}{\lambda^*} \rfloor^{\text{th}}$ iteration. Then the tuning parameter must lie in the interval $(\lambda^*, 2\lambda^*]$, resulting in a value of $\widehat{\gamma}$ such that $\text{supp}(\widehat{\gamma}) = \text{supp}(\gamma^*)$.

Probability by union bound: Now we study the probability for this algorithm to output a value of $\widehat{\gamma}$ that achieves exact recovery. Firstly, the algorithm stops as long as Lemma A.18 and Lemma A.24 hold, which holds with probability at least $1 - \frac{3}{n-t} - 2\exp\left(-2\left(\frac{1}{2} - c_t - \nu\right)^2 n\right)$.

Secondly, consider the argument that the algorithm correctly identifies all bugs. For each iteration, the events $\{C = 0 \text{ if a bug still exists}\}$ and

$\{C = 1$ if no bugs exist$\}$ hold as long as Lemma A.18 and Lemma A.24 hold, which happens with probability at least $1-\frac{3}{n-t}-2\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right)$. If the algorithm has K iterations, the probability that the algorithm flags all bugs is therefore at least $1-\frac{3K}{n-t}-2K\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right)$ by a union bound. Since we have argued that $K \leqslant \log_2 \frac{\lambda^u}{\lambda(\sigma^*)}$, the desired statement follows.

Proof of Corollary 2.13

According to the PDW procedure, we can set $\widehat{\gamma} = \vec{0}$, solve for $\hat{z}$ via the zero-subgradient equation, and check whether $\|\hat{z}\|_\infty < 1$, where $\widehat{z}$ is a subgradient of $\|\widehat{\gamma}\|_1$. The gradient of the loss function is equal to zero, which implies that

$$\widehat{z} = \frac{1}{\lambda n}\|\bar{P}^\top P_{X'}^\perp y'\|_\infty.$$

Therefore, we see that $\|\widehat{z}\|_\infty < 1$ for $\lambda > \frac{\|\bar{P}^\top P_{X'}^\perp y'\|_\infty}{n}$, which means the optimizer satisfies $\widehat{\gamma} = \vec{0}$. Since $\lambda_u = \frac{2\|\bar{P}^\top P_{X'}^\perp y'\|_\infty}{n}$, the output with tuning parameter λ_u gives $\widehat{\gamma}(\lambda_u) = 0$.

Note that

$$\|\bar{P}^\top P_{X'}^\perp y'\|_\infty = \|\bar{P}^\top \bar{P}\gamma^* + \bar{P}^\top P_{X'}^\perp \epsilon'\|_\infty \leqslant \|\bar{P}^\top \bar{P}\gamma^*\|_\infty + \|P_{X'}^\perp \epsilon'\|_\infty$$

by the triangle inequality. The second term is bounded by $2\max\{1, \sqrt{\frac{\eta n}{mL}}\}\cdot \sqrt{\log 2n}\,\sigma^*$ with probability at least $1-\frac{1}{n}$, since $e_j^\top P_{X'}^\perp \epsilon'$ is Gaussian with variance at most $\max\{1, \sqrt{\frac{\eta n}{mL}}\}\sigma^*$. For the first term, we have

$$\begin{aligned}
\|\bar{P}^\top \bar{P}\gamma^*\|_\infty &= \left\|\bar{P}^\top \bar{P}\gamma^*\right\|_\infty \\
&\overset{(i)}{\leqslant} t\left\|\bar{P}^\top \bar{P}\right\|_{\max}\|\gamma_T^*\|_\infty \\
&\overset{(ii)}{\leqslant} t\|\gamma^*\|_\infty \\
&\leqslant \frac{Cc_\nu}{2}\sqrt{1-c_t}\sqrt{\log 2n}\, n^{c_n+\frac{1}{2}}\sigma^*,
\end{aligned}$$

where (i) holds because $\|v^\top \gamma^*\|_1 = \sum_{i \in T} |v_i \gamma_i^*| \leqslant t\|v\|_\infty \|\gamma^*\|_\infty$ for any row v of the matrix $\bar{P}^\top \bar{P}$, and (ii) holds because $\bar{P}^\top \bar{P}$ is a submatrix of the projection matrix $P_{X'}^\perp$ and each entry of a projection matrix is upper-bounded by 1. Altogether, we obtain

$$\lambda_u \leqslant \left[\max\left\{1, \sqrt{\frac{\eta n}{mL}}\right\} \frac{2\sqrt{\log 2n}}{n} + \frac{Cc_v}{2}\sqrt{1-c_t}\sqrt{\log 2n}\, n^{c_n + \frac{1}{2}}\right] \sigma^*.$$

By a similar argument as in Theorem 2.12 and Corollary A.17, we know that Algorithm 1 stops with at most $\log_2 \frac{\lambda_u}{\lambda(\sigma^*)}$ with probability at least $1 - \frac{1}{n-t}$. Hence,

$$\begin{aligned}
\log_2 \frac{\lambda_u}{\lambda(\sigma^*)} &= \log_2 \frac{\left[\max\{1, \sqrt{\frac{\eta n}{mL}}\} + \frac{Cc_v}{4}\sqrt{1-c_t}\, n^{c_n+\frac{3}{2}}\right] \frac{2\sqrt{\log 2n}}{n}\sigma^*}{\frac{4}{1-\alpha'}\sqrt{2\log 2n(1-c_t)}\frac{\left\|\bar{P}_{T^c}^\perp\right\|_2}{n}\sigma^*} \\
&\overset{(1)}{\leqslant} \log_2 \frac{\left[\max\{1, \sqrt{\frac{\eta n}{mL}}\} + \frac{C}{4} n^{c_n+\frac{3}{2}}\right] 2\sqrt{\log n}}{\frac{4}{1-\alpha'}\sqrt{2\log 2n}} \\
&\overset{(2)}{\leqslant} \log_2 \frac{\left[\max\{1, \sqrt{\frac{\eta n}{mL}}\} + \frac{C}{4} n^{c_n+\frac{3}{2}}\right]}{2} \\
&\leqslant c\left(\frac{3}{2} + c_n\right)\log_2 n + \max\left\{0, \frac{1}{2}\log_2 \frac{\eta n}{mL} - 1\right\},
\end{aligned}$$

where (1) comes from the fact that $\bar{P}_{T^c}^\perp$ is a submatrix of $P_{X'}^\perp$, which has spectral norm 1 when $n \geqslant t + p + 1$; and (2) holds because $1 - \alpha' < 1$. To illustrate that $\|\bar{P}_{T^c}^\perp\|_2 = 1$, note that it is sufficient to show $\|P_{X',T^cT^c}^\perp\|_2 = 1$ $P_{X',T^cT^c}^\perp$ is a principal matrix of $P_{X'}^\perp$. By interlacing theorem (Hwang (2004)), we know that $\lambda_{\max}(P_{X',T^cT^c}^\perp)$ is no less than the $(t+1)^{\text{st}}$ largest eigenvalue of $P_{X'}^\perp$, which is a projection matrix and therefore has $n - p$ eigenvalues equal to 1. Thus, if $t + 1 \leqslant n - p$, i.e., $n \geqslant t + p + 1$, then $\|P_{X',T^cT^c}^\perp\|_2 = 1$.

Now that we have bounded the number of iterations, we consider probability that the statement holds. Note that ϵ' is sub-Gaussian and all the statements based on $\lambda(\sigma^*)$ hold with probability $1-\frac{1}{n-t}$. Compared to Theorem 2.12, note that on each iteration, we have subset support recovery with probability $1-\frac{1}{n-t}$; and on iteration $\log_2 \frac{\lambda_u}{\lambda(\sigma^*)}$, we have exact support recovery with probability $1-\frac{1}{n-t}$. Thus, we conclude that Algorithm 1 outputs a value of $\widehat{\lambda}$ that achieves exact recovery with probability at least

$$1 - \frac{5\left(c\log_2 n + \max\left\{0, \frac{1}{2}\log_2 \frac{\eta n}{mL}\right\}\right)}{n-t} - 2\left(c\log_2 n + \max\left\{0, \frac{1}{2}\log_2 \frac{\eta n}{mL}\right\}\right) e^{-2\left(\frac{1}{2}-c_t-\nu\right)^2 n}.$$

Proof of Proposition 2.11

We consider the three cases of Gaussian, sub-Gaussian and convex-concentration settings.

Let $\Sigma = \mathbb{E}[x_i x_i^\top]$ and $\Theta = \Sigma^{-1}$, and assume that $X^{(k)}$ corresponds to some X_S with rows indexed by S. Our goal is to prove that

$$\left\|\frac{X_S \Sigma^{-1} X_S^\top}{p} - I\right\|_{\max} \leqslant c \max\left\{\sqrt{\frac{\log|S|}{p}}, \frac{\log|S|}{p}\right\}, \tag{A.45}$$

$$\left\|\frac{X_S \top X_S}{|S|} - \Sigma\right\|_2 \leqslant \frac{\lambda_{\min}(\Sigma)}{2}, \tag{A.46}$$

for at most $\log_2 \frac{\lambda_u}{\lambda^*}$ of such sets S. Note that $T^c \subseteq S \subseteq [n]$ holds with probability at least $1 - \frac{\log_2 \frac{\lambda_u}{\lambda^*}}{n-t}$.

Proof of Proposition 2.11 for Gaussian case The spectral norm bound follows from standard results Vershynin (2010), which holds for a fixed set S with probability at least $1 - e^{-|S|} \geqslant 1 - e^{-(n-t)}$. Note that Algorithm 1 runs for at most $\log_2 \frac{\lambda_u}{\lambda^*}$ iterations by Theorem 2.12. Taking a union bound over all sets S, we obtain an overall probability of $1 - \log_2 \frac{\lambda_u}{\lambda^*} e^{-(1-c_t)n} \geqslant 1 - e^{-\frac{n}{2} + \log\log_2 \frac{\lambda_u}{\lambda^*}}$.

We now consider (A.45). Define $z_i = \Theta^{1/2} x_i$ for $1 \leqslant i \leqslant n$, so that

$$X\Theta^{1/2} = \begin{bmatrix} -z_1^\top- \\ \dots \\ -z_n^\top- \end{bmatrix}.$$

We know the $\Theta^{1/2} x_i$'s are i.i.d. isotropic Gaussian random vectors. Hence, $z_i^\top z_i \sim \chi^2(p)$ satisfies

$$\frac{\|z_i\|_2^2}{p} - 1 \leqslant 4\sqrt{\frac{\log \frac{1}{\delta}}{p}},$$

with probability at least $1 - \delta$. Similarly, we can bound $z_k^\top z_k$ and $(z_i + z_k)^\top (z_i + z_k)$. Since $z_i^\top z_k = \frac{1}{2}[(z_i + z_k)^\top (z_i + z_k) - z_i^\top z_i - z_k^\top z_k]$, we then have

$$\frac{\langle z_i, z_k \rangle}{p} \leqslant 8\sqrt{\frac{\log \frac{1}{\delta}}{p}}, \quad \forall i \neq k,$$

with probability at least $1 - \delta$.

We now choose $\delta = \frac{1}{n^c}$ for some $c > 2$ and take a union bound over all n^2 entries of the matrix $X\Theta X^\top$, to obtain

$$\left\| \frac{X\Theta X^\top}{p} - I \right\|_{\max} \leqslant c \max \left\{ \sqrt{\frac{\log n}{p}}, \frac{\log n}{p}, \right\}$$

with probability at least $1 - \frac{1}{n^{c'-2}}$, where $c' > 2$.

Finally, note that for all $S \subseteq [n]$, we have

$$\left\| \frac{X_S \Theta X_S}{p} - I \right\|_{\max} \leqslant \left\| \frac{X\Theta X}{p} - I \right\|_{\max}.$$

Proof of Proposition 2.11 for sub-Gaussian case By Lemma A.26, inequality (A.46) holds for a fixed set S, with probability at least $1-e^{-c|S|} \geqslant 1-e^{-c(n-t)}$ for some $c > 0$. Note that Algorithm 1 runs for at most $\log_2 \frac{\lambda_u}{\lambda^*}$ iterations. We then take a union bound over the possible subsets $T^c \subseteq S \subseteq [n]$ to reach a probability of at least $1 - \log_2 \frac{\lambda_u}{\lambda^*} e^{-c(1-c_t)n} \geqslant 1 - e^{-\frac{cn}{2}+\log\log_2 \frac{\lambda_u}{\lambda^*}}$.

Next, we focus on verifying inequality (A.45). Assuming that the x_i's are independent random vectors and the components of the x_i's are independent of each other, our goal is to prove that

$$\left\| \frac{X\Theta X^\top}{p} - I \right\|_{\max} \lesssim \max\left\{ \sqrt{\frac{\log n}{p}}, \frac{\log n}{p} \right\},$$

w.h.p., where $\Sigma = \mathrm{Cov}(x_i) = \Theta^{-1} =: D^2$ is a diagonal matrix.

Define $z_i = D^{-1}x_i$. Since the z_i's are mutually independent with independent components, we know that the vector $g_{ij} = (z_{i1}, ..., z_{ip}, z_{j1}, ..., z_{jp})^\top$, for $i \neq j$, also has independent components. Furthermore, the sub-Gaussian parameter of g_{ij} is bounded by $l_{\max} = \max_{q=1}^p \frac{K}{d_q^2}$, where K is the sub-Gaussian variance parameter of the x_i's. This is because for a unit vector u, we have

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda u^\top g_{ij}}\right] &= \Pi_{q=1}^p \mathbb{E}\left[e^{\lambda u_q z_{iq}}\right] \mathbb{E}\left[e^{\lambda u_{p+q} z_{jq}}\right] \\
&= \Pi_{q=1}^p \mathbb{E}\left[e^{\lambda \frac{u_q}{d_q} x_{iq}}\right] \mathbb{E}\left[e^{\lambda \frac{u_{p+q}}{d_q} x_{jq}}\right] \\
&\leqslant \Pi_{q=1}^p \mathbb{E}\left[e^{\lambda^2 \frac{u_q^2}{2d_q^2} K}\right] \mathbb{E}\left[e^{\lambda^2 \frac{u_{p+q}^2}{2d_q^2} K}\right] \\
&= \mathbb{E}\left[e^{\sum_{q=1}^p \lambda^2 \frac{u_q^2+u_{p+q}^2}{2d_q^2} K}\right] \\
&\leqslant \mathbb{E}\left[e^{\sum_{q=1}^p (u_q^2+u_{p+q}^2) \frac{\lambda^2}{2} l_{\max}}\right] \\
&= \mathbb{E}\left[e^{\frac{\lambda^2}{2} l_{\max}}\right].
\end{aligned}$$

Since we have assumed that $\|\Sigma\|_2$ is bounded, the d_q's are all bounded for each q, so l_{max} is bounded, as well.

Now let $A = \begin{bmatrix} 0_{p\times p} & I_{p\times p} \\ 0_{p\times p} & 0_{p\times p} \end{bmatrix}$. By the Hanson-Wright inequality, with probability at least $1-\delta$, we have

$$\left|\frac{\langle z_i, z_j\rangle}{p}\right| = \frac{g_{ij}^\top A g_{ij}}{p} \leqslant c_1\sqrt{\frac{\log\frac{2}{\delta}}{p}}, \tag{A.47}$$

where c_1 is a constant related to l_{max}.

Now applying the Hanson-Wright inequality to the vector z_i, we have

$$\left|\frac{\|z_i\|_2^2}{p} - \frac{\mathbb{E}[\|z_i\|_2^2]}{p}\right| \leqslant c_2 \max\left\{\sqrt{\frac{\log\frac{2}{\delta}}{p}}, \frac{\log\frac{2}{\delta}}{p}\right\}, \tag{A.48}$$

with probability at least $1-\delta$. Noting that $\mathbb{E}[\|z_i\|_2^2] = \mathrm{tr}(\Theta\Sigma) = p$, we will finally have

$$\left|\frac{\|z_i\|_2^2}{p} - 1\right| \leqslant c_2 \max\left\{\sqrt{\frac{\log\frac{2}{\delta}}{p}}, \frac{\log\frac{2}{\delta}}{p}\right\}.$$

Plugging in $\delta = \frac{2}{n^3}$ and taking a union bound, we then conclude that

$$\left\|\frac{X\Theta X^\top}{p} - I\right\|_{max} \leqslant 2\max\{c_1, c_2\}\max\left\{\sqrt{\frac{\log n}{p}}, \frac{\log n}{p}\right\},$$

with probability at least $1-\frac{2}{n}$.

Proof of Proposition 2.11 for convex concentration case Recall the following definition:

Definition A.16 (Convex concentration property)**.** *Let* X *be a random vector* *in $\mathbb{R}^d$. If for every 1-Lipschitz convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}[\varphi(X)] < \infty$ and for every $t > 0$, we have*

$$\mathbb{P}\left(|\varphi(X) - \mathbb{E}[\varphi(X)]| \geqslant t\right) \leqslant 2\exp(-t^2/K^2),$$

then X satisfies the convex concentration property with constant K.

Suppose x_i has the convex concentration property with parameter K. Note that

$$\begin{aligned}\left\|\frac{X\Theta X^\top}{p} - I\right\|_{\max} &= \max_{i,j}\left|e_i^\top\left(\frac{X\Theta X^\top}{p} - I\right)e_j\right| \\ &= \max_{i,j}\left|\frac{x_i^\top \Theta x_j}{p} - e_i^\top e_j\right|.\end{aligned}$$

By Lemma A.25, we thus have the exponential tail bound

$$\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_i}{p} - 1\right| \geqslant w\right) \leqslant 2\exp\left(-\frac{1}{C}\min\left\{\frac{w^2p^2}{2K^4\|\Theta\|_F}, \frac{wp}{K^2\|\Theta\|_2}\right\}\right),$$

for all $1 \leqslant i \leqslant p$, which implies that

$$\left|\frac{x_i^\top \Theta x_i}{p} - 1\right| \leqslant cK^2 \max\left\{\sqrt{\frac{\log\frac{2}{\delta}}{p}}, \frac{\log\frac{2}{\delta}}{p}\right\},$$

with probability at least $1 - \delta$. Taking $\delta = 2/n^3$, we then obtain

$$\left|\frac{x_i^\top \Theta x_i}{p} - 1\right| \leqslant cK^2 \max\left\{\sqrt{\frac{\log n}{p}}, \frac{\log n}{p}\right\}, \tag{A.49}$$

with probability at least $1 - \frac{2}{n^3}$.

Now we consider the off-diagonals $\frac{x_i \Theta x_j}{p}$, for $i \neq j$. We first rewrite

$$\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_j}{p}\right| \geqslant \Delta\right) = \mathbb{P}\left(\left|x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}\right| \geqslant \frac{\Delta p}{\|\Theta x_j\|_2}\right).$$

Conditioning on $\|\Theta x_j\|_2$ for some $w > 0$, we obtain

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_j}{p}\right| \geqslant \Delta\right) = \\
&\quad \mathbb{P}\left(\left|x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}\right| \geqslant \frac{\Delta p}{\|\Theta x_j\|_2}\middle| \|\Theta x_j\|_2 \geqslant w\right) \mathbb{P}\left(\|\Theta x_j\|_2 \geqslant w\right) \\
&\qquad + \mathbb{P}\left(\left|x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}\right| \geqslant \frac{\Delta p}{\|\Theta x_j\|_2}\middle| \|\Theta x_j\|_2 < w\right) \mathbb{P}\left(\|\Theta x_j\|_2 < w\right).
\end{aligned}$$

Since we have a convex 1-Lipschitz function mapping from x_i to $x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}$, we can further upper-bound the probability using the convex concentration property:

$$
\begin{aligned}
&\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_j}{p}\right| \geqslant \Delta\right) \\
&\leqslant \mathbb{P}\left(\|\Theta x_j\|_2 \geqslant w\right) + \mathbb{P}\left(\left|x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}\right| \geqslant \frac{\Delta p}{\|\Theta x_j\|_2} \middle| \|\Theta x_j\|_2 < w\right) \\
&\leqslant \mathbb{P}\left(\|x_j\|_2 \geqslant \frac{w}{\|\Theta\|_2}\right) + \mathbb{P}\left(\left|x_i^\top \frac{\Theta x_j}{\|\Theta x_j\|_2}\right| \geqslant \frac{\Delta p}{w}\right) \\
&\overset{(1)}{\leqslant} \mathbb{P}\left(\|x_j\|_2 - \mathbb{E}[\|x_j\|_2] \geqslant \frac{w}{\|\Theta\|_2} - \mathbb{E}[\|x_j\|_2]\right) + 2\exp\left(-\frac{\Delta^2 p^2}{w^2 K^2}\right) \\
&\overset{(2)}{\leqslant} \mathbb{P}\left(\|x_j\|_2 - \mathbb{E}[\|x_j\|_2] \geqslant \frac{w}{\|\Theta\|_2} - \sqrt{\mathbb{E}[\|x_j\|_2^2]}\right) + 2\exp\left(-\frac{\Delta^2 p^2}{w^2 K^2}\right) \\
&\overset{(3)}{\leqslant} 2\exp\left(-\frac{\left(\frac{w}{\|\Theta\|_2} - \sqrt{\mathrm{tr}(\Sigma)}\right)^2}{K^2}\right) + 2\exp\left(-\frac{\Delta^2 p^2}{w^2 K^2}\right) \\
&\leqslant 2\exp\left(-\frac{\left(\frac{w}{\|\Theta\|_2} - \sqrt{p\|\Sigma\|_2}\right)^2}{K^2}\right) + 2\exp\left(-\frac{\Delta^2 p^2}{w^2 K^2}\right),
\end{aligned}
$$

where (1) and (3) use the convex concentration property and (2) uses Jensen's inequality. The last inequality assumes that $w \geqslant \sqrt{p\|\Sigma\|_2}$, can be guaranteed if we choose w sufficiently large.

Plugging $\Delta = c\max\left\{\frac{\log n}{p}, \sqrt{\frac{\log n}{p}}\right\}$ and $w = c'\left(\sqrt{p} + \sqrt{\log n}\right)$ into the above derivations, we then obtain

$$
\begin{aligned}
\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_j}{p}\right| \geqslant \Delta\right) \leqslant\ & 2\exp\left(-\frac{c'' \log n}{K^2}\right) \\
&+ 2\exp\left(-c''' \frac{\max\{(\log n)^2, p\log n\}}{(p+\log n)K^2}\right).
\end{aligned}
$$

If $p > \log n$, then $2\exp\left(-\frac{\max\{(\log n)^2, p\log n\}}{(p+\log n)K^2}\right) \leqslant 2\exp\left(-\frac{c'''' \log n}{K^2}\right)$; If $p \leqslant \log n$, then $2\exp\left(-\frac{\max\{(\log n)^2, p\log n\}}{(p+\log n)K^2}\right) \leqslant 2\exp\left(-\frac{c''''' \log n}{K^2}\right)$. Hence, we have

$$\mathbb{P}\left(\left|\frac{x_i^\top \Theta x_j}{p}\right| \geqslant \Delta\right) \leqslant 2\exp\left(-C\log n\right).$$

We can choose c and c' sufficiently large to ensure that $C > 2$. Combining this with inequality (A.49) using a union bound, we finally obtain the desired result.

Auxiliary lemmas

By Theorem 2.6, we have the following corollary:

Corollary A.17. *For two data pools, suppose the eigenvalue and mutual incoherence conditions hold. Let $\lambda \geqslant \lambda(\sigma^*)$. Then with probability $1 - \frac{1}{n-t}$, we have* $\text{supp}(\widehat{\gamma}) \subseteq \text{supp}(\gamma^*)$, *and*

$$\|\widehat{\gamma}(\lambda) - \gamma^*\|_\infty \leqslant G'(\lambda). \tag{A.50}$$

Proof. Recall that the rule for regularizer selection in Theorem 2.6 is

$$\lambda \geqslant \frac{2}{1-\alpha'}\left\|\bar{P}_{T^c}^\top\left(I - \bar{P}_T(\bar{P}_T^\top \bar{P}_T)^{-1}\bar{P}_T^\top\right)\frac{\epsilon'}{n}\right\|_\infty.$$

Note that $e_j^\top \bar{P}_{T^c}^\top\left(I - \bar{P}_T(\bar{P}_T^\top \bar{P}_T)^{-1}\bar{P}_T^\top\right)\frac{\epsilon'}{n}$ is sub-Gaussian with variance parameter $\max\{1, \frac{\eta n}{mL}\}\frac{\|\bar{P}_{T^c}^\perp\|_2^2\sigma^{*2}}{n^2}$. We have

$$\max_{j\in T^c}\left|e_j^\top \bar{P}_{T^c}^\top\left(I - \bar{P}_T(\bar{P}_T^\top \bar{P}_T)^{-1}\bar{P}_T^\top\right)\frac{\epsilon'}{n}\right| \leqslant 4\max\left\{1, \frac{\eta n}{mL}\right\}\sqrt{\log 2(n-t)\frac{\|\bar{P}_{T^c}^\perp\|_2}{n}\sigma^{*2}},$$

with probability at least $1 - \frac{1}{n-t}$. According to the definition of $\lambda(\sigma^*)$, we can further derive the bound for $\widehat{\gamma}$, since

$$\|\widehat{\gamma} - \gamma^*\|_\infty \leqslant \|(P_{X',TT}^\perp)^{-1}P_{X',T\cdot}^\perp\epsilon'\|_\infty + 2n\lambda(\sigma^*)\left\|(P_{X',TT}^\perp)^{-1}\right\|_\infty.$$

□

The following lemma suggests that if $\min_{i \in T} |\gamma_i^*| \geqslant G'(2\lambda^*)$, then $\mathrm{supp}(\widehat{\gamma}(\lambda)) = \mathrm{supp}(\gamma^*)$ if we take $\lambda \in [\lambda^*, 2\lambda^*]$.

Lemma A.18. *If* $\min_{i \in T} |\gamma_i^*| \geqslant G'(2\lambda^*)$, *then taking* $\lambda \in [\lambda^*, 2\lambda^*]$ *yields an estimator* $\widehat{\gamma}(\lambda)$ *that satisfies* $\mathrm{supp}(\widehat{\gamma}(\lambda)) = \mathrm{supp}(\gamma^*)$.

Proof. According to Theorem 2.6, for a regularizer $\lambda \in [\lambda^*, 2\lambda^*]$, we have $\widehat{\gamma}_{T^c} = 0$ and $\|\widehat{\gamma}(\lambda) - \gamma^*\|_\infty \leqslant G'(\lambda)$. If $\min_{i \in T} |\gamma_i^*| \geqslant G'(2\lambda^*)$, then by the triangle inequality, we have

$$|\widehat{\gamma}_i| > \min_{i \in T} |\gamma_i^*| - G'(\lambda) \geqslant G'(2\lambda^*) - G'(\lambda) \geqslant 0,$$

for all $i \in T$. □

We use X_S to represent some $X^{(k)}$ for $S \subseteq [n]$, as shown in Algorithm 2.12. In each loop of the algorithm, we know that the points in S^c all lie in T by the subset recovery result. Thus, $S \supseteq T^c$. Let $l = n - |S|$, and note that $0 \leqslant l \leqslant t$.

Lemma A.19. *Suppose Assumption 2.10 holds. If* $\lambda_{\min}(\Sigma)$ *and* $\lambda_{\max}(\Sigma)$ *are bounded, then*

$$\left\| P_{X_S}^{\perp} - \left(1 - \frac{p}{n-l}\right) I \right\|_{\max} \leqslant C \frac{\max\{p, \sqrt{p \log(n-l)}, \log(n-l)\}}{n-l}.$$

Proof. Using the notation $\Theta = \Sigma^{-1}$ and $\widehat{\Sigma} = \frac{X_S^\top X_S}{|S|}$, we have

$$\begin{aligned}
\left\| P_{X_S}^{\perp} - \left(1 - \frac{p}{|S|}\right) I_{|S| \times |S|} \right\|_{\max} &= \left\| X_S (X_S^\top X_S)^{-1} X_S^\top - \frac{p}{|S|} I \right\|_{\max} \\
&\leqslant \left\| \frac{X_S (\widehat{\Sigma})^{-1} X_S^\top}{|S|} - \frac{X_S \Theta X_S^\top}{|S|} \right\|_{\max} + \left\| \frac{X_S \Theta X_S^\top}{|S|} - \frac{p}{|S|} I \right\|_{\max}.
\end{aligned}$$

By assumption, we may bound the second term by

$$\left\|\frac{X_S\Theta X_S^\top}{|S|} - \frac{p}{|S|}I\right\|_{\max} \leqslant \frac{p}{|S|}\cdot c\max\left\{\sqrt{\frac{\log|S|}{p}}, \frac{\log|S|}{p}\right\} = \frac{c\max\{\sqrt{p\log|S|}, \log|S|\}}{|S|}.$$

For the first term, we have

$$\begin{aligned}\left\|\frac{X_S(\widehat{\Sigma})^{-1}X_S^\top}{|S|} - \frac{X_S\Theta X_S^\top}{|S|}\right\|_{\max} &= \frac{1}{|S|}\left\|X_S\left((\widehat{\Sigma})^{-1} - \Theta\right)X_S^\top\right\|_{\max} \\ &\leqslant \left\|(\widehat{\Sigma})^{-1} - \Theta\right\|_2 \cdot \max_{1\leqslant i\leqslant |S|}\frac{1}{|S|}\|X_S^\top e_i\|_2^2.\end{aligned}$$

We now have the bound

$$\begin{aligned}\left\|(\widehat{\Sigma})^{-1} - \Theta\right\|_2 &\leqslant \frac{\frac{1}{2}\lambda_{\min}(\Sigma)}{\lambda_{\min}(\Sigma)\lambda_{\min}(\widehat{\Sigma})} \\ &\leqslant \frac{\frac{1}{2}\lambda_{\min}(\Sigma)}{\lambda_{\min}(\Sigma)(\lambda_{\min}(\Sigma) - \frac{1}{2}\lambda_{\min}(\Sigma))} = \frac{1}{\lambda_{\min}(\Sigma)},\end{aligned}$$

as well, where the second inequality holds by Weyl's Theorem (Horn and Johnson (1994)): $\lambda(\widehat{\Sigma}) \geqslant \lambda(\Sigma) - \|\Sigma - \widehat{\Sigma}\|_2$. The basic idea for the first inequality is to use the multiplicativity of matrix norms to conclude that

$$\begin{aligned}\left\|A^{-1} - B^{-1}\right\|_2 &\leqslant \left\|A^{-1}(A-B)B^{-1}\right\|_2 \\ &\leqslant \left\|A^{-1}\right\|_2\|A-B\|_2\left\|B^{-1}\right\|_2 \\ &= \frac{\|A-B\|_2}{\lambda_{\min}(A)\cdot\lambda_{\min}(B)}.\end{aligned} \tag{A.51}$$

Hence, an upper bound on $\|A - B\|_2$—which we obtain from our assumptions—together with minimum eigenvalue bounds on A and B, implies an upper bound on $\left\|A^{-1} - B^{-1}\right\|_2$.

Finally, we have

$$\begin{aligned}
\max_{1\leqslant i\leqslant |S|} \frac{1}{|S|}\|X_S^\top e_i\|_2^2 &\leqslant \max_{1\leqslant i\leqslant |S|} \frac{1}{|S|} \cdot \frac{\|\Theta^{1/2}X_S^\top e_i\|_2^2}{\lambda_{\min}^2(\Theta^{1/2})} \\
&= \frac{1}{\lambda_{\min}(\Theta)} \cdot \max_{1\leqslant i\leqslant |S|} \frac{\|\Theta^{1/2}X_S^\top e_i\|_2^2}{|S|} \\
&= \lambda_{\max}(\Sigma) \cdot \max_{1\leqslant i\leqslant |S|} \frac{e_i^\top X_S \Theta X_S^\top e_i}{|S|} \\
&\leqslant \lambda_{\max}(\Sigma) \cdot \left\| \frac{X_S\Theta X_S^\top}{|S|} \right\|_{\max}.
\end{aligned}$$

By assumption, we have

$$\left\| \frac{X_S\Theta X_S^\top}{p} - I \right\|_{\max} \leqslant c \max\left\{ \sqrt{\frac{\log|S|}{p}}, \frac{\log|S|}{p} \right\}.$$

Hence, rescaling and using the triangle inequality, we have

$$\begin{aligned}
\left\| \frac{X_S\Theta X_S^\top}{|S|} \right\|_{\max} &\leqslant \frac{p}{|S|}\left(\left\| \frac{X_S\Theta X_S^\top}{p} - I \right\|_{\max} + 1 \right) \\
&\leqslant \frac{p}{|S|} + \frac{p}{|S|} \max\left\{ \sqrt{\frac{\log|S|}{p}}, \frac{\log|S|}{p} \right\}.
\end{aligned}$$

Altogether, we have the bound

$$\left\| \frac{X_S(\widehat{\Sigma})^{-1}X_S^\top}{|S|} - \frac{X_S\Theta X_S^\top}{|S|} \right\|_{\max} \leqslant \frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)} \cdot \frac{p}{|S|} \left(1 + \max\left\{ \sqrt{\frac{\log|S|}{p}}, \frac{\log|S|}{p} \right\} \right).$$

Finally, we have

$$\frac{c\max\{\sqrt{p\log|S|},\log|S|\}}{|S|}+c''\frac{p}{|S|}\left(1+\max\left\{\sqrt{\frac{\log|S|}{p}},\frac{\log|S|}{p}\right\}\right)\\ \leqslant C\frac{\max\{p,\sqrt{p\log|S|},\log|S|\}}{|S|}.$$

This finishes the proof. □

We use $\alpha(k)$ to represent the k^{th} order statistics of $|\epsilon_i|$, for $i\in T^c$, where $\alpha_{(1)}\leqslant\alpha_{(2)}\leqslant\cdots\leqslant\alpha_{(n-t)}$.

Lemma A.20. *For i.i.d. random variables* $\{|\epsilon_i|\}_{i\in T^c}$, *the* k^{th} *order statistics, for any* $k\in\{\frac{n-t}{2},\ldots,\frac{n}{2}\}$ *satisfy*

$$c_\nu\sigma^*\leqslant\alpha(k)\leqslant C_\nu\sigma^*,$$

with probability at least $1-2\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2n\right)$, *for* $\nu\in(0,\frac{1}{2})$ *such that* $\nu<\frac{1}{2}-c_t$.

Proof. By the assumptions on the noise distribution, we have

$$\nu=\mathbb{P}\left[|\epsilon_i|\leqslant c_\nu\sigma^*\right]\ \text{and}\ \nu=\mathbb{P}\left[|\epsilon_i|\geqslant C_\nu\sigma^*\right].$$

Let ξ_i's be i.i.d. Bernoulli variables such that

$$\xi_i=\begin{cases}1 & \text{if } |\epsilon_i|\leqslant c_\nu\sigma^*,\\ 0 & \text{otherwise.}\end{cases}$$

Note that $t=c_tn$ for some positive constant $c_t\in(0,\frac{1}{2})$. We have

$$k-\nu(n-t)\geqslant\frac{n-t}{2}-\nu(n-t)=\frac{(1-c_t)(1-2\nu)}{2}n>0$$

and

$$\left(\frac{k}{n-t}-\nu\right)^2(1-c_t) \geqslant \left(\frac{1}{2}-\nu\right)^2(1-c_t) \geqslant \left(\frac{1-2\nu}{2}\right)\left(\frac{1-c_t-2\nu}{2}\right).$$

By Hoeffding's inequality (Hoeffding (1994)), we then obtain

$$\begin{aligned}\mathbb{P}\left[\sum_{i=1}^{n-t}\xi_i \geqslant k\right] &= \mathbb{P}\left[\sum_{i=1}^{n-t}\xi_i - \nu(n-t) \geqslant k-\nu(n-t)\right] \\ &\leqslant \exp\left(-2\left(\frac{k}{n-t}-\nu\right)^2(n-t)\right) \\ &\leqslant \exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right),\end{aligned}$$

implying that

$$\mathbb{P}\left[\alpha(k) \leqslant c_\nu\sigma^*\right] = \mathbb{P}\left[\sum_{i=1}^{n}\xi_i \geqslant k\right] \leqslant \exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right).$$

Similarly, let η_i's be i.i.d. Bernoulli variables such that

$$\eta_i = \begin{cases} 1 & \text{if } |\epsilon_i| \geqslant C_\nu\sigma^*, \\ 0 & \text{otherwise.}\end{cases}$$

Note that the assumption that $c_t < \frac{1}{2}-\nu$ gives us

$$n-t-k-\nu(n-t) > n-c_t n-\frac{n}{2}-\nu(1-c_t)n \geqslant \left(\frac{1}{2}-c_t-\nu\right)n > 0,$$

and

$$\left(1-\frac{k}{n-t}-\nu\right)^2(1-c_t) \geqslant \left(\frac{1}{2}-c_t-\nu\right)^2\frac{n}{n-t} \geqslant \left(\frac{1}{2}-c_t-\nu\right)^2.$$

Then by Hoeffding inequality, we obtain

$$\mathbb{P}\left[\sum_{i=1}^{n-t}\eta_i \geqslant n-t-k\right] = \mathbb{P}\left[\sum_{i=1}^{n-t}\eta_i - \nu(n-t) \geqslant n-t-k-\nu(n-t)\right]$$
$$\leqslant \exp\left(-2\left(1-\frac{k}{n-t}-\nu\right)^2(n-t)\right)$$
$$\leqslant \exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right),$$

so that

$$\mathbb{P}\left[\alpha(k) \geqslant C_\nu \sigma^*\right] \leqslant \exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right).$$

□

Lemma A.21. *Suppose the assumptions of Lemma A.19 hold and*

$$n^{1-2c_n} \geqslant \max\left\{\frac{32C^2}{1-c_t}\log(2n)\left(p^2+\log^2 n\right),\quad \left(\frac{24}{c_\nu}\right)^{\frac{1}{c_n}}\right\},$$

and

$$\max_{i\in S}|\gamma_S^*| \leqslant \frac{c_\nu C}{2}\sqrt{1-c_t}\sqrt{\log 2n}\,\frac{n^{1/2+c_n}}{t}\sigma^*,$$

for some constant $c_n \in (0, \frac{1}{2})$*. Then the* k^{th} *order statistic of* $|P_{X_S}^{\perp}(\gamma_S^* + \epsilon_S)|$ *and the* k^{th} *order statistic of* $\left|\left(1-\frac{p}{|S|}\right)(\gamma_S^*+\epsilon_S)\right|$ *have differences of at most* $\frac{\bar{c}}{4}\sigma^*$*, for any* $k \in [|S|]$*, with probability at least* $1-\frac{1}{n-t}$*.*

Proof. Recall that $l = n - |S|$. Now consider the sequences $\{z_i = |e_i^\top P_{X_S}^{\perp}(\gamma_S^* + \epsilon_S)|\}_{i=1}^{n-l}$ and $\left\{w_i = \left|\left(1-\frac{p}{n-l}\right)(\gamma_{S,i}^*+\epsilon_{S,i})\right|\right\}_{i=1}^{n-l}$. By the triangle inequality, we have

$$|z_i - w_i| \leqslant \left| e_i^\top \left(P_{X_S}^\perp - \left(1 - \frac{p}{n-l}\right) I \right) (\gamma_S^* + \epsilon_S) \right|$$
$$\leqslant \left| \underbrace{e_i^\top \left(P_{X_S}^\perp - \left(1 - \frac{p}{n-l}\right) I \right) \gamma_S^*}_{\nu_i} \right| + \left| \underbrace{e_i^\top \left(P_{X_S}^\perp - \left(1 - \frac{p}{n-l}\right) I \right) \epsilon_S}_{u_i} \right|,$$

for $i = 1, \ldots, n - l$.

Since u_i is sub-Gaussian with parameter at most

$$\left\| (P_{X_S}^\perp)_{i\cdot} - e_i^\top \left(1 - \frac{p}{n-l}\right) \right\|_2^2 \sigma^{*2},$$

we can upper-bound the maximum of $\{|u_i|\}$. With probability at least $1 - \frac{1}{n-t}$, we have

$$\max_{i \in S} |u_i| \leqslant 2\sqrt{\log 2(n-l)}\sigma^* \left\| (P_{X_S}^\perp)_{i\cdot} - e_i^\top \left(1 - \frac{p}{n-l}\right) \right\|_2$$
$$\leqslant 2\sqrt{\log 2(n-l)}\sigma^* \sqrt{n-l} \left\| P_{X_S}^\perp - \left(1 - \frac{p}{n-l}\right) \right\|_{\max}$$
$$\leqslant 2C\sqrt{\log 2(n-l)} \frac{(\sqrt{p} + \sqrt{\log(n-l)})^2}{\sqrt{n-l}} \sigma^*,$$

where the last inequality follows by Lemma A.19. Further note that since $n^{1-2c_n} \geqslant \frac{32C^2}{1-c_t} \log(2n)\,(p^2 + \log^2 n)$ for some $c_n \in (0, \frac{1}{2})$, we have $\max_{i \in S} |u_i| \leqslant \frac{1}{n^{c_n}} \sigma^*$.

For the ν_i's, we have

$$\begin{aligned}\max_{i\in S}|\nu_i| &\overset{(i)}{\leqslant} t\left\|P^{\perp}_{X_S}-\left(1-\frac{p}{n-l}\right)\right\|_{\max}\max_{i\in S}|\gamma^*_S| \\ &\overset{(ii)}{\leqslant}\sqrt{\frac{t^2}{n(1-c_t)}}\frac{(\sqrt{p}+\sqrt{\log(n-l)})^2}{\sqrt{n-l}}\max_{i\in S}|\gamma^*_S| \\ &\overset{(iii)}{\leqslant}\frac{1}{2C}\sqrt{\frac{1}{1-c_t}}\frac{t}{n^{1/2+c_n}}\frac{1}{\sqrt{\log 2n}}\max_{i\in S}|\gamma^*_S|,\end{aligned} \tag{A.52}$$

where (i) holds because $|a^\top\gamma^*_S| \leqslant \|a\|_\infty\|\gamma^*_S\|_\infty|\text{supp}(\gamma^*_S)|$ for any vector a, (ii) holds by Lemma A.19, and (iii) holds by our assumption on n. Combining this with the assumption that

$$\max_{i\in S}|\gamma^*_S| \leqslant \frac{c_\nu C}{4}\sqrt{1-c_t}\sqrt{\log 2n}\,\frac{n^{1/2+c_n}}{t}\sigma^*,$$

we obtain $\max_{i\in S}|\nu_i| \leqslant \frac{c_\nu}{8}\sigma^*$. Finally, using the fact that $n \geqslant \left(\frac{24}{c_\nu}\right)^{\frac{1}{c_n}}$, we obtain

$$|z_i - w_i| \leqslant \frac{c_\nu}{6}\sigma^*,$$

with probability at least $1-\frac{1}{n-t}$.

We then use the following lemma:

Lemma A.22. *For two sequences* $a_1,\dots,a_n$ *and* $b_1,\dots,b_n$ *such that* $|a_i-b_i| \leqslant c$ *for some positive number* c*, the* j^{th} *order statistics of* $\{a_i\}$ *and* $\{b_i\}$*, denoted by* $\alpha_a(j)$ *and* $\alpha_b(j)$*, satisfy*

$$|\alpha_a(j)-\alpha_b(j)| \leqslant c. \tag{A.53}$$

Proof. Without loss of generality, suppose $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$. If there exists $j \in [n]$ such that inequality (A.53) does not hold, then we have either $a_j > c + \alpha_b(j)$ or $a_j < \alpha_b(j) - c$. If the first case occurs, we have

$$a_n \geqslant \cdots \geqslant a_j > c + \alpha_b(j) \geqslant c + \alpha_b(j-1) \geqslant \cdots c + \alpha_b(1).$$

Pick a number z between $c + \alpha_b(j)$ and a_j. We see that at least j of the b_i's, denoted by $\vec{b}_\downarrow$, are smaller than $z - c$; and at least $n - j + 1$ of a_i's, denoted by $\vec{a}_\uparrow$, are greater than z. This means that at most $j - 1$ of a_i's are no larger than z. Note that for the $\vec{b}_\downarrow$, the components of the corresponding vector $\vec{a}_\downarrow$ are within a distance of c, so the elements of $\vec{a}_\downarrow$ must be at most z. However, this contradicts the fact that at most $j - 1$ of the a_i's are at most z. This concludes the proof. □

From Lemma A.22, we can compare the order statistics of sequences $\{z_i\}_{i=1}^n$ and $\{w_i\}_{i=1}^n$ and conclude that they have differences of at most $\frac{\bar{c}}{6}\sigma^*$, with probability at least $1 - \frac{1}{n-t}$. □

Lemma A.23. *Suppose the conditions of Lemma A.20 and Lemma A.21 hold, and also* $\min_{i \in T} |\gamma_i^*| > 4\sqrt{\log(2n)}\sigma^*$. *Then*

$$\left(c_\nu - \frac{|S|}{|S| - p}\frac{c_\nu}{6}\right)\sigma^* \leqslant \widehat{\sigma} \leqslant \left(\frac{|S|}{|S| - p}\frac{c_\nu}{6} + C_\nu\right)\sigma^*,$$

with probability at least $1 - 2\exp\left(-2\left(\frac{1}{2} - c_t - \nu\right)^2 n\right) - \frac{2}{n-t}$.

Proof. Let $M_P(S)$ denote the median of $|P_{X_S}^\perp(\gamma_S^* + \epsilon_S)|$. By Lemma A.21, we know that $M_P(S)$ is close to the median of $\left|\left(1 - \frac{p}{|S|}\right)(\gamma_S^* + \epsilon_S)\right|$. Thus, it remains to analyze the median of $\{|\gamma_i^* + \epsilon_i|\}_{i \in S}$.

Note that for $j \in T^c$, we have $|\gamma_j^* + \epsilon_j| = |\epsilon_j|$. Therefore, for all $j \in S \cap T^c = T^c$, we have $|\gamma_j^* + \epsilon_i|_\infty \leqslant 2\sqrt{\log 2n}\,\sigma^*$, with probability at least $1 - \frac{1}{n}$.

For $i \in T \cap S$, by the assumption that $\min_{i \in T} |\gamma_i^*| > 4\sqrt{\log 2n}\,\sigma^*$, we have $|\gamma_i^* + \epsilon_i| \geqslant |\gamma_i^*| - |\epsilon_i| > 2\sqrt{\log 2n}\,\sigma^*$. Therefore, the median of $|\gamma_S^* + \epsilon_S|$ is actually the k^{th} order statistics of $|\epsilon_{T^c}|$ for some $\{k \in \frac{n-t}{2}, \ldots, \frac{n}{2}\}$.

By Lemma A.21, we have

$$\left(1-\frac{p}{|S|}\right)\alpha(k)-\frac{c_\nu}{6}\sigma^* \leqslant M_P(S) \leqslant \left(1-\frac{p}{|S|}\right)\alpha(k)+\frac{c_\nu}{6}\sigma^*.$$

In Algorithm 1, at some iteration k, we have $\widehat{\sigma}=\frac{|S|}{|S|-p}M_P(S)$, where S is the corresponding set of indices of $\left(\mathrm{supp}(\widehat{\gamma}^{(k)})\right)^c$. Thus,

$$\alpha(k)-\frac{|S|}{|S|-p}\frac{c_\nu}{6}\sigma^* \leqslant \widehat{\sigma} \leqslant \alpha(k)+\frac{|S|}{|S|-p}\frac{c_\nu}{6}\sigma^*.$$

Combining this with Lemma A.20, we have

$$\left(c_\nu-\frac{|S|}{|S|-p}\frac{c_\nu}{6}\right)\sigma^* \leqslant \widehat{\sigma} \leqslant \left(\frac{|S|}{|S|-p}\frac{c_\nu}{6}+C_\nu\right)\sigma^*,$$

with probability at least $1-2\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right)-\frac{2}{n-t}$. □

Lemma A.24. *Suppose* $n \geqslant 12p$,

$$\min_{i\in T}|\gamma_i^*| \geqslant \frac{5}{4}\left(\frac{c_\nu+5C_\nu}{\bar{c}}\right)\sqrt{\log 2n}\,\sigma^*,$$

and inequality (A.52) *holds. Then*

$$\|P^{\perp}_{X_{T^c}}\epsilon_{T^c}\|_\infty < \frac{5}{2\bar{c}}\sqrt{\log 2n}\widehat{\sigma}, \tag{A.54}$$

and for any γ_S^* *such that* $S\cap T \neq \emptyset$, *we have*

$$\|P^{\perp}_{X_S}(\gamma_S^*+\epsilon_S)\|_\infty > \frac{5}{2\bar{c}}\sqrt{\log 2n}\widehat{\sigma}, \tag{A.55}$$

with probability at least $1-\frac{3}{n-t}-2\exp\left(-2\left(\frac{1}{2}-c_t-\nu\right)^2 n\right)$.

Proof. We first establish the bound on $\|P^{\perp}_{X_{T^c}}\epsilon_{T^c}\|_\infty$. Note that $e_j^\top P^{\perp}_{X_{T^c}}\epsilon_{T^c}$ is

Gaussian with variance at most $\max_{j \in T^c}(P^{\perp}_{X_{T^c}})_{jj}$, so

$$\begin{aligned}\|P^{\perp}_{X_{T^c}} \epsilon_{T^c}\|_\infty &= \max_{j \in T^c} |e_j^\top P^{\perp}_{X_{T^c}} \epsilon_{T^c}| \\ &\leqslant \max_j (P^{\perp}_{X_{T^c}})_{jj} 2\sqrt{\log 2(n-l)}\sigma^* \leqslant 2\sqrt{\log 2n}\, \sigma^*,\end{aligned}$$

with probability at least $1 - \frac{1}{n-t}$. In addition, Lemma A.23 implies that

$$\begin{aligned}\|P^{\perp}_{X_{T^c}} \epsilon_{T^c}\|_\infty &\leqslant 2\sqrt{\log 2n} \frac{1}{\left(-\frac{c_\nu}{6}\frac{|S|}{|S|-p} + c_\nu\right)} \widehat{\sigma} \\ &\leqslant 2\sqrt{\log 2n} \frac{1}{\left(-\frac{1}{6}\frac{|S|}{|S|-p} + 1\right)\bar{c}} \widehat{\sigma}.\end{aligned}$$

For $n \geqslant 12p$, we therefore conclude the bound (A.54).

Now consider γ^*_S with nonzero elements, i.e., $S \supset T^c$. We have

$$\begin{aligned}\|P^{\perp}_{X_S}(\gamma^*_S + \epsilon_S)\|_\infty &\geqslant \max_{i \in S} |e_i^\top P^{\perp}_{X_S} \gamma^*_S| - \|P^{\perp}_{X_S} \epsilon_S\|_\infty \\ &\geqslant \max_{i \in S} |e_i^\top P^{\perp}_{X_S} \gamma^*_S| - 2\sqrt{\log 2n}\, \sigma^*,\end{aligned}$$

with probability at least $1 - \frac{1}{n-t}$. We now split $P^{\perp}_{X_S}$ into $P^{\perp}_{X_S} - (1 - \frac{p}{n-l})I$ and $(1 - \frac{p}{n-l})I$. By the triangle inequality, we have

$$\begin{aligned}&\max_{i \in [n-l]} \left|e_i^\top P^{\perp}_{X_S} \gamma^*_S\right| \\ &\quad \geqslant \max_{i \in [n-l]} \left|e_i^\top \left(1 - \frac{p}{n-l}\right) I \gamma^*_S\right| - \max_{i \in [n-l]} \left|e_i^\top \left(P^{\perp}_{X_S} - \left(1 - \frac{p}{n-l}\right) I\right) \gamma^*_S\right| \\ &\quad \geqslant \left(1 - \frac{p}{n-l}\right) \|\gamma^*_S\|_\infty - \max_{i \in [n-l]} \left|\underbrace{e_i^\top \left(P^{\perp}_{X_S} - \left(1 - \frac{p}{n-l}\right) I\right) \gamma^*_S}_{\nu_i}\right|.\end{aligned}$$

Plugging this into the result from inequality (A.52), we then obtain

$$\max_{i\in[n-l]} \left|e_i^\top P_{X_S}^{\perp}\gamma_S^*\right| \geqslant \left(1-\frac{p}{n-l}\right)\|\gamma_S^*\|_\infty - \frac{c_\nu}{8}\sigma^*.$$

Therefore, we have

$$\|P_{X_S}^{\perp}(\gamma_S^*+\epsilon_S)\|_\infty \geqslant \left(1-\frac{p}{n-t}\right)\min_{i\in T}|\gamma_i^*| - (2\sqrt{\log 2n}+c_\nu/8)\sigma^*.$$

By the assumption that $n \geqslant 12p$ and Lemma A.23, we then obtain

$$\begin{aligned}
\|P_{X_S}^{\perp}(\gamma_S^*+\epsilon_S)\|_\infty &\geqslant \frac{5}{6}\min_{i\in T}|\gamma_i^*| - \frac{(2\sqrt{\log 2n}+c_\nu/8)}{c_\nu - \frac{|S|}{|S|-p}\frac{c_\nu}{6}}\widehat{\sigma} \\
&\geqslant \frac{5}{6}\min_{i\in T}|\gamma_i^*| - \frac{(2\sqrt{\log 2n}+c_\nu/8)}{c_\nu - \frac{c_\nu}{5}}\widehat{\sigma} \\
&\geqslant \frac{5}{6}\min_{i\in T}|\gamma_i^*| - \frac{13}{6}\frac{\sqrt{\log 2n}}{\frac{4c_\nu}{5}}\widehat{\sigma}.
\end{aligned}$$

Thus, $\|P_{X_S}^{\perp}(\gamma_S^*+\epsilon_S)\|_\infty \geqslant \frac{5}{2\bar{c}}\sqrt{\log 2n}\,\widehat{\sigma}$ if $\min\limits_{i\in T}|\gamma_i^*|$ satisfies

$$\min_{i\in T}|\gamma_i^*| \geqslant \sqrt{\log 2n}\,\widehat{\sigma}\left(\frac{3}{\bar{c}}+\frac{13}{4c_\nu}\right).$$

This can be further achieved according to Lemma A.23 if

$$\min_{i\in T}|\gamma_i^*| \geqslant \sqrt{\log 2n}\,\sigma^*\left(\frac{3}{\bar{c}}+\frac{13}{4c_\nu}\right)\left(C_\nu+\frac{c_\nu}{6}\frac{|S|}{|S|-p}\right).$$

Also note that by the assumption of $\min_{i\in T}|\gamma_i|$, we have

$$\begin{aligned}
\min_{i\in T}|\gamma_i^*| &\geqslant \frac{5}{4}\left(\frac{c_\nu+5C_\nu}{\bar{c}}\right)\sqrt{\log 2n}\,\sigma^* \\
&\geqslant \sqrt{\log 2n}\,\sigma^*\left(\frac{3}{\bar{c}}+\frac{13}{5c_\nu-\bar{c}}\right)\left(C_\nu+\frac{c_\nu}{6}\frac{|S|}{|S|-p}\right).
\end{aligned}$$

This conclues the proof. □

Lemma A.25 (Theorem 2.5 in Adamczak Adamczak (2015))**.** *Suppose* X *is a zero-mean random vector in* $\mathbb{R}^n$ *satisfying the convex concentration property with constant* K*. Then for any fixed matrix* $A \in \mathbb{R}^{n\times n}$ *and any* $w > 0$*, we have*

$$\mathbb{P}\left(|X^\top AX - \mathbb{E}[X^\top AX]| \geqslant w\right) \leqslant 2\exp\left(-\frac{1}{C}\min\left\{\frac{w^2}{2K^4\|A\|_F^2}, \frac{w}{K^2\|A\|_2}\right\}\right).$$

Lemma A.26. *Suppose* $X \in \mathbb{R}^{n\times p}$ *has i.i.d. rows from a zero-mean distribution satisfying the convex concentration property with constant* K*. Then*

$$\left\|\frac{X^\top X}{n} - \mathbb{E}\left[\frac{X^\top X}{n}\right]\right\|_2 \leqslant c\frac{\lambda_{\min}(\Sigma)}{2},$$

with probability at least $1 - \exp(-n)$*.*

Proof. Note that for any fixed unit vector $u \in \mathbb{R}^p$, the map $\varphi : x \mapsto \langle x, u\rangle$ is convex and 1-Lipschitz. Hence, by the definition of the convex concentration property, each $x_i^\top u$ is sub-Gaussian with parameter proportional to K. In fact, this is enough to show the desired matrix concentration result (cf. Vershynin Vershynin (2010)). We omit the details. □

A.1.5 Appendix for Chapter 2.5

In this sectopm, we provide proofs and additional details for the results in Chapter 2.5.

Proof of Theorem 2.14

We will prove a stronger results here, which implies Theorem 2.14. This is actually mentioned by Remark 2.16.

Theorem A.27. *With respect to* D*, the bug generator, who has attacking budgets no more than* t*, cannot fail the sign support recovery if only if* (2.16) *holds. That* *failure of sign support recovery,* $\text{sign}(\widehat{\gamma}) \neq \text{sign}(\gamma^*)$*, means either* $\widehat{\gamma}_j \neq 0$ *for some* $j \in T^c$ *or* $\widehat{\gamma}_i \gamma_i^* \leqslant 0$ *for some* $i \in T$.

Proof of Theorem 2.14. We will use the following lemma to prove Theorem 2.14.

Lemma A.28. *The following two properties are equivalent:*

(*a*) *For any vector* $\gamma^* \in \mathbb{R}^d$ *with support* K*, the constraint-based optimization has all solutions* $\widehat{\gamma}$ *satisfying* $\text{sign}(\widehat{\gamma}) = \text{sign}(\gamma^*)$.

(*b*) *The matrix* $\bar{P}(D)$ *satisfies the restricted nullspace property with respect to* K.

Proof of Lemma A.28. We first prove $(b) \implies (a)$. This immediately follows Theorem 7.8 in Wainwright (2019) since $(b) \implies \gamma^* = \widehat{\gamma}$ for any vector γ^* with $\text{supp}(\gamma^*) = K$, it thus implies $(b) \implies \text{sign}(\widehat{\gamma}) = \text{sign}(\gamma^*)$. Or we can show it directly as follow. Suppose (a) doesn't hold. Then, we have $\Delta := \gamma^* - \widehat{\gamma} \neq 0$. By the constraint and the objective, it also needs to satisfy that $\Delta \in \mathrm{Null}(\bar{P}(D))$ and

$$\|\gamma^* - \Delta\|_1 = \|\widehat{\gamma}\|_1 \leqslant \|\gamma^*\|_1 = \|\gamma_K^*\|_1.$$

Therefore, we have

$$\|\gamma_K^*\|_1 - \|\Delta_K\|_1 + \|\Delta_{K^c}\|_1 \leqslant \|\gamma_K^* - \Delta_K\|_1 + \|\Delta_{K^c}\|_1 \leqslant \|\gamma_K^*\|_1,$$

which means a nonzero $\Delta \in \mathrm{Null}(\bar{P}) \cap \mathbb{C}^A$ and causes a contradiction. Thus when (b) is true, (a) holds as well.

From now on to the end of the proof, we will abuse notation by using $\bar{P}$ to represent $\bar{P}(D)$. The remaining thing is to prove $(a) \implies (b)$. We will prove by contradiction. If (b) doesn't hold, then there exists a nonzero Δ such that $\bar{P}\Delta = 0$ and $\|\Delta_{K^c}\|_1 \leqslant \|\Delta_K\|_1$. We consider a γ^* with $\gamma_K^* = \Delta_K$ and $\gamma_{K^c}^* = \vec{0}$. Let $\widehat{\gamma}$ be the optimizer given this γ^*. By (a), we shall have

$\text{sign}(\widehat{\gamma}) = \text{sign}(\gamma^*) = \text{sign}\left(\begin{bmatrix} \Delta_K \\ \vec{0}_{(n-t)\times 1} \end{bmatrix}\right)$. The idea is to construct a γ' that has no larger ℓ_1 norm than $\widehat{\gamma}$ and has support not equal to K, which contradicts with (a), and therefore, (b) must hold.

Consider $\gamma' = \widehat{\gamma} - c \cdot \Delta$ where $c = \frac{\widehat{\gamma}_i}{\Delta_i}$ for $i = \arg\min_{j \in K} \frac{\widehat{\gamma}_j}{\Delta_j}$. Since Δ is a nonzero vector, we must have $\Delta_l \neq 0$ for some $l \in K$. Therefore, we have c being positive finite, $\gamma'_i = 0$ and $|\widehat{\gamma}_j| \geqslant c|\Delta_j|$ for all $j \in K$. Therefore, we further get

$$\bar{P}(\gamma^* - \gamma') = \bar{P}(\gamma^* - \widehat{\gamma} + c\Delta) = \bar{P}(\gamma^* - \widehat{\gamma}) = 0,$$

as well as

$$\begin{aligned} \|\gamma'\|_1 &= \|\widehat{\gamma}_K - c \cdot \Delta_K\|_1 + \|\widehat{\gamma}_{K^c} - c \cdot \Delta_{K^c}\|_1 \\ &\overset{(i)}{=} \|\widehat{\gamma}_K\|_1 - c\|\Delta_K\|_1 + c\|\Delta_{K^c}\|_1 \\ &\overset{(ii)}{\leqslant} \|\widehat{\gamma}\|_1, \end{aligned}$$

where (i) is because $\text{sign}(\widehat{\gamma}_K) = \text{sign}(\Delta_K), c > 0, |\widehat{\gamma}_K| \geqslant c|\Delta_K|$ and $\widehat{\gamma}_{K^c} = 0$, (ii) is because $\Delta \in \mathbb{C}(K)$. Hence, we find a γ' to have smaller or equal ℓ_1 norm than $\widehat{\gamma}$. This contradicts with the fact that all the solutions have support K or $\widehat{\gamma}$ is the optimal solution. Therefore, (b) must hold and $(a) \implies (b)$. □

We first prove that (2.16) is sufficient. For any $|K| \leqslant t$ and $K \subseteq [n]$, we know that $\text{Null}(\bar{P}(D)) \cap \mathbb{C}(K) = \{0\}$. Then by Proposition A.28, we conclude that $sign(\widehat{\gamma}) = sign(\gamma^*)$ with $\text{supp}(\gamma^*) = K$ for any subset K of size no more than t.

We second prove that (2.16) is necessary. Note that for any subset K of size less equal to t, we have $\text{sign}(\widehat{\gamma}) = \text{sign}(\gamma^*)$ with $\text{supp}(\gamma^*) = K$. By Proposition A.28, it means $\bar{P}(D)$ satisfies the restricted nullspace property for any such K. Therefore $\text{Null}(\bar{P}(D)) \cap \mathbb{C}^A = \{\vec{0}\}$. □

Theorem 2.14 immediately holds from Theorem A.27.

Proof of Remark 2.17

We will prove the statement in Remark 2.17 here.

Proposition A.29. *The subspace* $\mathrm{Null}(\bar{P}(D))$ *is equivalent to* $\{u \in \mathbb{R}^n \mid \exists v \in \mathbb{R}^p,$ *such that* $u = Xv, X_D v = 0\}$.

Proof of Proposition A.29. We first prove $\mathrm{Null}(\bar{P}(D)) \supseteq \{u \in \mathbb{R}^n \mid \exists v \in \mathbb{R}^p,$ such that $u = Xv, X_D v = 0\}$. Let $u = \left(X + M^\top X_D\right) v$ for some $v \in \mathbb{R}^p$, where $M \in \mathbb{R}^{m \times p}$ contains m rows stacked with the canonical vectors indexed by D so that $MX = X_D$. We have

$$\begin{aligned}
&\left(I - X\left(X^\top X + X_D^\top X_D\right)^{-1} X^\top\right) u \\
&\quad = u - X\left(X^\top X + \frac{\eta n}{m} X_D^\top X_D\right)^{-1} X^\top \left(X + \frac{\eta n}{m} M^\top X_D\right) v \\
&\quad = \frac{\eta n}{m} M^\top X_D v.
\end{aligned}$$

Besides, we have

$$\begin{aligned}
X_D \left(X^\top X + \frac{\eta n}{m} X_D^\top X_D\right)^{-1} X^\top u &= X_D \left(X^\top X + X_D^\top X_D\right)^{-1} X^\top \left(X + M^\top X_D\right) v \\
&= X_D v.
\end{aligned}$$

Therefore $X_D v = 0, u = Xv \implies u \in \mathrm{Null}(\bar{P}(D))$.

Secondly we prove $\mathrm{Null}(\bar{P}(D)) \subseteq \{u \mid \exists v \in \mathbb{R}^d,$ such that $u = Xv, X_D v = 0\}$. Let u be some vector in $\mathbb{N}(X_D)$. Then we have

$$u = X\left(X^\top X + X_D^\top X_D\right)^{-1} X^\top u, \tag{A.56}$$

and

$$X_D \left(X^\top X + X_D^\top X_D\right)^{-1} X^\top u = 0. \tag{A.57}$$

By (A.57), we have $\left(X^\top X + X_D^\top X_D\right)^{-1} X^\top u = v$ for some $v \in \mathrm{Null}(X_D)$. Plugging this back to (A.56), we have $u = Xv$. Hence, we have $u \in \{u \mid \exists v \in \mathbb{R}^d, \text{ such that } u = Xv, X_D v = 0\}$.

□

Proof of Theorem 2.18

Here we prove the proof of Theorem 2.18. We write the minimax MILP here again.

$$\min_{\xi \in \{0,1\}^n} \quad \max_{\substack{a, a^+, a^- \in \mathbb{R}^n, \\ u, u^+, u^- \in \mathbb{R}^n, \\ v \in \mathbb{R}^d \\ z, w \in \{0,1\}^n}} \sum_{j=1}^n a_j^+ - a_j^-, \tag{A.58}$$

$$\text{subject to } u = Xv, \tag{A.59}$$

$$u = u^+ - u^-, a = u^+ + u^-, \tag{A.60}$$

$$u^+, u^- \geqslant 0, u^+ \leqslant z, \; u^- \leqslant (\mathbb{1}_n - z), \tag{A.61}$$

$$\sum_{i=1}^n w_i \leqslant t, \tag{A.62}$$

$$a^+ \leqslant w, \; a^- \leqslant \mathbb{1}_n - w, a = a^+ + a^-, a^+ \geqslant 0, a^- \geqslant 0, \tag{A.63}$$

$$\sum_{i=1}^n \xi_i \leqslant m \; i = 1, \ldots, n, \tag{A.64}$$

$$u \leqslant \mathbb{1}_n - \xi, u \geqslant -(\mathbb{1}_n - \xi). \tag{A.65}$$

Proof of Theorem 2.18. We first argue that if (A.66) has the unique solution of $(u, v) = (\vec{0}, \vec{0})$, then (2.16) holds and thus the debugger can add m

points indexed by D to achieve support recovery.

$$\min_{\substack{D\in[n],\\ |D|\leqslant m}} \max_{K\subseteq[n],|K|\leqslant t,u\in\mathbb{R}^n,v\in\mathbb{R}^d} \|u_K\|_1 - \|u_{K^c}\|_1, \quad \text{subject to } u = Xv, X_D v = 0, \|u\|_\infty \leqslant 1. \tag{A.66}$$

Suppose (2.16) doesn't hold. Then there exists $K \subseteq [n], |K| \leqslant t$ and a nonzero vector u' such that $u' = Xv, X_D v = 0$ and $\|u'_K\|_1 \geqslant \|u'_{K^c}\|_1$. And $\frac{u'}{\|u'\|_2}$ satisfies $\|u'\|_\infty \leqslant 1$. This contradicts with that (A.66) has the unique solution of $(u, v) = (\vec{0}, \vec{0})$, then (2.16) holds. This concludes our first part of the proof.

Now we argue that the MILP is equivalent to (A.66). Equation (A.59) is inherited from original constraint. Equations in (A.60) and (A.61) are equivalent to $a = |u|$. Note that u^+, u^- respectively correspond to the positive and negative parts of u. If $z_i = 0$, then $u_i^+ = 0$, $u_i^- \leqslant 1$ and $u_i^- = -u_i$. If $z_i = 1$, then $u_i^- = 0$, $u_i^+ \leqslant 1$ and $u_i^+ = u_i$. The vector w indicates K in (A.66). If $w_i = 1$, then $i \in K$ otherwise $i \in K^c$. Therefore, equation (A.62) restricts the attacking budget to t. Then, equations in (A.63) are equivalent to $a_i^+ = |u_i|, a_i^- = 0$ for $i \in K$ and $a_i^- = |u_i|, a_i^+ = 0$ for $i \in K^c$. Therefore, the objective function corresponds to $\|u_K\|_1 - \|u_{K^c}\|_1$.

Note that the variable in the first layer is ξ. If $\xi_i = 1$, it means the debugger queries the point x_i. And the constraint $X_D v = 0$ is replaced by (A.65). This is because $x_i^\top v = 0 \Leftrightarrow u_i = 0$. If $\xi_j = 0$, then u_j just needs to satisfy $|u_j| \leqslant 1$.

Therefore, we have shown that the MILP is equivalent to (A.66) and thus conclude Theorem 2.18. □

A.2 Appendix for Chapter 3

A.2.1 Reminder from Algebraic Geometry

In this appendix section, we will introduce some definitions and results from abstract algebra, commutative algebra and general topology, along with writing some examples of these definitions. The are used for proofs of the results in Section 3.3. Everything in this section can be found in any of the standard textbooks to algebraic geometry, such as (Hartshorne, 1977), (Görtz and Wedhorn, 2020) or (Liu, 2002).

Definitions

Let $\mathbb{C}[x_1, \dots, x_n]$ be the polynomial ring in n variables over $\mathbb{C}$.

Definition A.30 (Zero set). *For any subset $S \subset \mathbb{C}[x_1, \dots, x_n]$, we define the zero set of S to be the common zeros of all elements in S, namely*

$$\mathbb{V}(S) := \{x \in \mathbb{C}^n : f(x) = 0 \textit{ for all } f \in S\}$$

Clearly, if $\mathfrak{a}$ is the ideal of $\mathbb{C}[x_1, \dots, x_n]$ generated by S, then $\mathbb{V}(S) = \mathbb{V}(\mathfrak{a})$. Furthermore, Hilbert's Basis Theorem ((Atiyah and Macdonald, 1969, Corollary 7.6)) implies that the polynomial ring $\mathbb{C}[x_1, \dots, x_n]$ is noetherian, that is, all its ideals are finitely generated. So there exist finitely many elements $f_1, \dots, f_r \in S$ such that $\mathbb{V}(S) = \mathbb{V}(f_1, \dots, f_r)$.

Proposition-Definition A.31 (Zariski topology). *The sets $\mathbb{V}(\mathfrak{a})$, where $\mathfrak{a}$ runs through the set of ideals of $\mathbb{C}[x_1, \dots, x_n]$, are the closed sets of a topology on $\mathbb{C}^n$, called the Zariski topology.*

If we consider $\mathbb{C}^n$ as a topological space equipped with the Zariski topology, then we will denote this space by $\mathbb{A}^n_{\mathbb{C}}$ and it is called the *affine n-space over $\mathbb{C}$.*

By knowledge of general topology, any non-empty open subset of $\mathbb{A}^n_{\mathbb{C}}$ is dense.

Definition A.32 (Affine variety)**.** *Closed subsets of $\mathbb{A}^n_{\mathbb{C}}$ are called affine variety*[1]*.*

For example, $\mathbb{A}^n_{\mathbb{C}}$ itself is an affine variety. Moreover, we have that

- $\mathbb{V}(f) \subset \mathbb{A}^1_{\mathbb{C}}$ *is an affine variety for any non-zero polynomial* $f \in \mathbb{C}[x]$, *consisting of* $\deg f$ *points in* $\mathbb{C}$ (*counted with multiplicity*)*;*
- *the hyperplane*

$$H(a_1, \ldots, a_n) := \left\{ (x_1, \ldots, x_n) \in \mathbb{A}^n_{\mathbb{C}} : \sum_{i=1}^{n} a_i x_i = 0 \right\} \subset \mathbb{A}^n_{\mathbb{C}} \quad \text{(A.67)}$$

  is an affine variety for any $(a_1, \ldots, a_n) \in \mathbb{C}^n \setminus \{0\}$.

Let $X \subset \mathbb{A}^n_{\mathbb{C}}$ be an affine variety.

Definition A.33 (Dimension)**.** *The dimension of* X *is the dimension of its underlying topological space.*

For example, we have

- $\dim \mathbb{A}^n_{\mathbb{C}} = n$,
- $\dim \mathbb{V}(f) = 0$ *for any non-zero polynomial* $f \in \mathbb{C}[x]$,
- $\dim H(a_1, \ldots, a_n) = n-1$ *with* H *defined by* (A.67) *for any* $(a_1, \ldots, a_n) \in \mathbb{C}^n \setminus \{0\}$.

Definition A.34 (Degree)**.** *The degree of* X *is the number of points* (*counted with multiplicity*) *in the intersection* $X \cap L$ *for some sufficiently general affine* $(n - \dim X)$*-space* $L \subset \mathbb{A}^n_{\mathbb{C}}$.

[1] Compared to the usual definition, we do not require irreducibility here.

A priority of the definition needs it to be well-defined, i.e. for generic affine $(n - \dim X)$*-space* $L \subset \mathbb{A}^n_{\mathbb{C}}$*, the number of points (counted with multiplicity) in the intersection* $X \cap L$ *is constant. Note it is true – a proof of this can be found in (Görtz and Wedhorn, 2020, §(14.31)).*

For example, we have

- $\deg \mathbb{A}^n_{\mathbb{C}} = 1$ *since the only affine* 0*-space is* $\{0\}$*,*
- $\deg \mathbb{V}(f) = \#\mathbb{V}(f) = \deg f$ *for any non-zero polynomial* $f \in \mathbb{C}[x]$*,*
- $\deg H(a_1, \dots, a_n) = 1$ *with* H *defined by* (A.67) *for any* $(a_1, \dots, a_n) \in \mathbb{C}^n \setminus \{0\}$*.*

Definition A.35. *For two polynomials* $\mathbf{P}_1(z), \mathbf{P}_2(z) \in \mathbb{Q}[z]$*, we say* $\mathbf{P}_1(z) \succ \mathbf{P}_2(z)$ *if* $\mathbf{P}_1(n) \geqslant \mathbf{P}_2(n)$ *for all* $n \gg 0, n \in \mathbb{Z}$*.*

Results

Theorem A.36 ((Gunning and Rossi, 2009), Chapter I, Section B, Corollary 10)**.** *For any non-zero polynomial* $f \in \mathbb{C}[x_1, \dots, x_m]$*, we have that* $\mathbb{V}(f) \subset \mathbb{C}^m$ *is of* λ_m*-measure zero, where* λ_m *is the Lebesgue measure on* $\mathbb{C}^m$*.*

Theorem A.37 ((Hartshorne, 1977), Proof of Corollary III.9.10)**.** *Let* $X \subset \mathbb{P}^n_{\mathbb{C}}$ *be a projective variety of dimension* r*. Then the degree of its Hilbert polynomial is* r *and the coefficient of its leading term is* $\deg X / r!$*. That is, the Hilbert polynomial of* $X \subset \mathbb{P}^n_{\mathbb{C}}$ *has the form*

$$\mathbf{P}_X(z) = \frac{\deg X}{r!} z^r + \textit{lower degree terms} \in \mathbb{Q}[z]$$

In particular, if $r = 0$*, then* $\mathbf{P}_X(z) = \deg X = \#X(\mathbb{C})$ *is constant.*

Theorem A.38 (Special case of (Nitsure, 2005), Theorem 5.13)**.** *Let* $\mathscr{X} \subset \mathbb{A}^m_{\mathbb{C}} \times \mathbb{P}^n_{\mathbb{C}}$ *be a closed subvariety. For any* $t \in \mathbb{A}^m_{\mathbb{C}}$*, let* $\mathscr{X}_t \subset \mathbb{P}^n_{\mathbb{C}}$ *be the fiber over* t *under the projection* $\mathscr{X} \subset \mathbb{A}^m_{\mathbb{C}} \times \mathbb{P}^n_{\mathbb{C}} \to \mathbb{A}^m_{\mathbb{C}}$ *and* $\mathbf{P}_t(z) \in \mathbb{Q}[z]$ *its Hilbert* *polynomial. If* f *is the minimal element in* $\mathscr{P} := \{\mathbf{P}_t(z) : t \in \mathbb{A}^m_{\mathbb{C}}\}$ *with respect to* $\succ$ *(in the sense of Definition A.35), then*

$$S_f := \{t \in \mathbb{A}^m_{\mathbb{C}} : \mathbf{P}_t(z) = f\} \subset \mathbb{A}^m_{\mathbb{C}}$$

is a Zariski open dense subset.

In Theorem A.38, it is often useful to view $\mathscr{X}$ as a family of projective schemes $\mathscr{X}_t \subset \mathbb{P}^n_{\mathbb{C}}$ parametrized by $t \in \mathbb{A}^m_{\mathbb{C}}$.

Lemma A.39. *Any Zariski closed proper subset of* $\mathbb{A}^m_{\mathbb{C}}$ *is of* $\lambda^{\mathbb{C}}_m$*-measure zero.*

Proof. Let $Z \subsetneq \mathbb{A}^m_{\mathbb{C}}$ be a Zariski closed proper subset. We write

$$Z = \mathbb{V}(f_1, \dots, f_r) = \bigcap_{i=1}^{r} \mathbb{V}(f_i)$$

for some polynomials $f_i \in \mathbb{C}[x_1, \dots, x_m]$. Since Z is proper, at least one of f_i's is non-zero, say f_1. By (Gunning and Rossi, 2009, Chapter I, Section B, Corollary 10), it holds that $\mathbb{V}(f_1) \subset \mathbb{C}^m$ is of $\lambda^{\mathbb{C}}_m$-measure zero. As a subset of a set of $\lambda^{\mathbb{C}}_m$-measure zero, Z is again of $\lambda^{\mathbb{C}}_m$-measure zero (since the Lebesgue measure on $\mathbb{C}^m$ is complete). □

A.2.2 Derivations for Section 3.4.3

Let $\{i : i = 1, \dots, n\}$ denote a set of n alternatives. A ranking of $(1, \dots, n)$ determined by $\sigma \in \mathfrak{S}_n$ is an ordering

$$\sigma(1) \succ \sigma(2) \succ \cdots \succ \sigma(n).$$

Given a parameter $\theta = (\theta_1, \dots, \theta_n)$, the probability of the ranking σ is given by

$$\mathbb{P}[\sigma] := \prod_{i=1}^{n} \frac{\theta_{\sigma(i)}}{\sum_{j \geqslant i} \theta_{\sigma(j)}}.$$

We are going to prove the following statement for $n \geqslant 3$ by induction,

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \mathsf{others}\right] = \frac{\theta_{\sigma(1)}}{\sum_{i=1}^{n} \theta_{\sigma(i)}} \frac{\theta_{\sigma(2)}}{\sum_{i=2}^{n} \theta_{\sigma(i)}}$$

For the base case $n = 3$, we get by definition that

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \mathsf{others}\right] = \mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \sigma(3)\right]$$
$$= \frac{\theta_{\sigma(1)}}{\sum_{i=1}^{3} \theta_{\sigma(i)}} \frac{\theta_{\sigma(2)}}{\sum_{i=2}^{3} \theta_{\sigma(i)}} \frac{\theta_{\sigma(3)}}{\theta_{\sigma(3)}} = \frac{\theta_{\sigma(1)}}{\sum_{i=1}^{3} \theta_{\sigma(i)}} \frac{\theta_{\sigma(2)}}{\sum_{i=2}^{3} \theta_{\sigma(i)}}.$$

For the inductive step, assume we have

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \mathsf{other}\ n-3\ \mathsf{items}\right] = \frac{\theta_{\sigma(1)}}{\sum_{i=1}^{n-1} \theta_{\sigma(i)}} \frac{\theta_{\sigma(2)}}{\sum_{i=2}^{n-1} \theta_{\sigma(i)}}. \quad \text{(A.68)}$$

Then we split our case of n into $n-2$ subcases,

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \mathsf{others}\right] = \sum_{i \neq \sigma(1), \sigma(2)} \mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ i \succ \mathsf{other}\ n-3\ \mathsf{items}\right].$$

Note that the probability formula on $\sigma(2) \succ i \succ \mathsf{other}\ n-3\ \mathsf{items}$ does not depend on $\theta_{\sigma(1)}$ and has $n-1$ items in it. Then by the hypothesis (A.68), we can further get

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ i \succ \mathsf{other}\ n-3\ \mathsf{items}\right] = \frac{\theta_{\sigma(1)}}{\sum_{j=1}^{n} \theta_{\sigma(j)}} \frac{\theta_{\sigma(2)}}{\sum_{j=2}^{n} \theta_{\sigma(j)}} \frac{\theta_i}{\sum_{k \neq \sigma(1), \sigma(2)}^{n} \theta_k}.$$

Hence, it yields that

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \mathsf{others}\right] = \frac{\theta_{\sigma(1)}}{\sum_{j=1}^{n} \theta_{\sigma(j)}} \frac{\theta_{\sigma(2)}}{\sum_{j=2}^{n} \theta_{\sigma(j)}} \sum_{i \neq \sigma(1), \sigma(2)} \frac{\theta_i}{\sum_{k \neq \sigma(1), \sigma(2)}^{n} \theta_k}$$
$$= \frac{\theta_{\sigma(1)}}{\sum_{j=1}^{n} \theta_{\sigma(j)}} \frac{\theta_{\sigma(2)}}{\sum_{j=2}^{n} \theta_{\sigma(j)}}.$$

Without loss of generality, we were assuming $\sum_{i=1}^{n} \theta_i = 1$. Thus, we conclude

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \text{others}\right] = \frac{\theta_{\sigma(1)}}{1} \frac{\theta_{\sigma(2)}}{1-\theta_{\sigma(1)}} = \frac{\theta_{\sigma(1)}\theta_{\sigma(2)}}{1-\theta_{\sigma(1)}}.$$

For two mixtures, we then have

$$\mathbb{P}\left[\sigma(1) \succ \sigma(2) \succ \text{others; two mixtures}\right] = p_1 \frac{a_{\sigma(1)} a_{\sigma(2)}}{1-a_{\sigma(1)}} + p_2 \frac{b_{\sigma(1)} b_{\sigma(2)}}{1-b_{\sigma(1)}},$$

where a_i's are parameters for the first mixture and b_i's are parameters for the second mixture.

Finally, the last formula to be proved is

$$\begin{aligned}
&\mathbb{P}\left[\sigma(1) \succ \text{others; two mixtures}\right] \\
&\quad = \sum_{i \neq \sigma(1)} \mathbb{P}\left[\sigma(1) \succ i \succ \text{others; two mixtures}\right] \\
&\quad = p_1 \frac{a_{\sigma(1)}}{1-a_{\sigma(1)}} \sum_{i \neq \sigma(1)} a_i + p_2 \frac{b_{\sigma(1)}}{1-b_{\sigma(1)}} \sum_{i \neq \sigma(1)} b_i \\
&\quad = p_1 \frac{a_{\sigma(1)}}{1-a_{\sigma(1)}} (1 - a_{\sigma(1)}) + p_2 \frac{b_{\sigma(1)}}{1-b_{\sigma(1)}} (1 - b_{\sigma(1)}) \\
&\quad = p_1 a_{\sigma(1)} + p_2 b_{\sigma(1)}.
\end{aligned}$$

A.2.3 Results for Mixtures of MNL Models with 2-slate and 3-slate

In this example, we consider the mixtures of MNL models with 2-slate and 3-slate for $n \geqslant 3$. Let $a_{1:n}$, $b_{1:n}$ be the score parameters of the two mixtures. Then, we obtain that

$$\begin{aligned} \forall i \neq j, i \in [n], j \in [n], \quad \eta_{i,j} &= p_1 \frac{a_i}{a_i + a_j} + p_2 \frac{b_i}{b_i + b_j}, \\ \forall i \neq j \neq k, i, j, k \in [n], \quad \eta_{i,j,k} &= p_1 \frac{a_i}{a_i + a_j + a_k} + p_2 \frac{b_i}{b_i + b_j + b_k}. \end{aligned} \tag{A.69}$$

Here, we choose to scale up $a_{1:n}$ by multiplying a constant such that we get $a_1 = 1$ and similarly manipulate $b_{1:n}$ to have $b_1 = 1$. This won't influence the values of $\eta_{i,j}$'s and $\eta_{i,j,k}$'s. In Chierichetti et al. (2015), the authors scale up $a_{1:n}$ to get $a_1 + a_2 + a_3 = 1$ and $b_{1:n}$ to have $b_1 + b_2 + b_3 = 1$. Our choice is for the convenience of defining the set of bad parameters in our set up.

Given p_1 and p_2, to determine the scores of two mixtures, we try to solve the following equation system for $\mathbf{x} := x_{1:n}, \mathbf{y} := y_{1:n}$ being variables and for $n \geqslant 3$,

$$\begin{cases} \forall i \neq j, i, j \in [n],\ p_1 \dfrac{x_i}{x_i + x_j} + p_2 \dfrac{y_i}{y_i + y_j} = \eta_{i,j}, \\ \forall i \neq j \neq k, i, j, k \in [n],\ p_1 \dfrac{x_i}{x_i + x_j + x_k} + p_2 \dfrac{y_i}{y_i + y_j + y_k} = \eta_{i,j,k}, \\ x_1 = y_1 = 1. \end{cases} \tag{A.70}$$

Let $Q^{2n-2}_{MNL23} := \prod_{i=1}^{2n-2} [r_i, R_i] \subseteq \mathbb{R}^{2n-2}$ be the domain of $(a_{2:n}, b_{2:n})$. And we assume $R_i > r_i > 0$. This interval assumption of the domain is just to diverse the choices in this draft. Then the set of bad parameters that do not have the identifiability property is:

$$\begin{aligned}
&N^{2n-2}_{MNL23} = \big\{(a_{2:n}, b_{2:n}) \in Q^{2n-2}_{MNL23} : \exists \left(a^{\#}_{2:n}, b^{\#}_{2:n}\right) \in Q^{2n-2}_{MNL23}, \text{s.t.} \\
&(a^{\#}_{2:n} \neq a_{2:n} \vee b^{\#}_{2:n} \neq b_{2:n}) \quad \wedge \\
&\left(\forall i < j, i \in [n], j \in [n],\ \eta_{i,j}(\mathbf{a}^{\#}, \mathbf{b}^{\#}) = \eta_{i,j}(\mathbf{a}, \mathbf{b}) \text{ for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\right) \quad \wedge \\
&\left(\forall i \in [n], j \in [n], k \in [n],\ \eta_{i,j,k}(\mathbf{a}^{\#}, \mathbf{b}^{\#}) = \eta_{i,j,k}(\mathbf{a}, \mathbf{b}) \text{ for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\right)\big\}\,.
\end{aligned} \tag{A.71}$$

We will later show N^{2n-2}_{MNL23} has lebesgue measure zero for the mixtures of MNL models being identifiable.

Similar to previous, when we consider p as an variable in the polynomial functions, our domain $Q^{2n-1}_{MNL23,p}$ becomes to $Q^{2n-2}_{MNL23,p} \times (0,1) \subseteq \mathbb{R}^{2n-1}$. And we define the set of bad parameters as

$$\begin{aligned}
N^{2n-1}_{MNL23,p} = \big\{&(a_{2:n}, b_{2:n}, p_1) \in Q^{2n-1}_{MNL23,p} : \exists \left(a^{\#}_{2:n}, b^{\#}_{2:n}, p^{\#}\right) \in Q^{2n-1}_{MNL23,p}, \text{s.t.} \\
&(a^{\#}_{2:n} \neq a_{2:n} \vee b^{\#}_{2:n} \neq b_{2:n} \vee p^{\#} \neq p_1) \quad \wedge \\
&\big(\forall i < j, i \in [n], j \in [n],\ \eta_{i,j}(\mathbf{a}^{\#}, \mathbf{b}^{\#}, p^{\#}) = \eta_{i,j}(\mathbf{a}, \mathbf{b}, p_1) \\
&\quad \text{for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big) \quad \wedge \\
&\big(\forall i \in [n], j \in [n], k \in [n],\ \eta_{i,j,k}(\mathbf{a}^{\#}, \mathbf{b}^{\#}, p^{\#}) = \eta_{i,j,k}(\mathbf{a}, \mathbf{b}, p_1) \\
&\quad \text{for } a^{\#}_1 = b^{\#}_1 = a_1 = b_1 = 1\big)\big\}\,.
\end{aligned} \tag{A.72}$$

Now we will consider two parameter spaces. One is the space of $(a_{2:n}, b_{2:n})$, for which Chierichetti et al. (2018) has perfectly solved the identifiability issue (c.f. Theorem 13) for a uniform mixture ($p_1 = 0.5$), where the technique is different from ours. The other is the space of $(a_{2:n}, b_{2:n}, p_1)$.

Parameter space of $(a_{2:n}, b_{2:n})$

In terms of the conclusion, the generic identifiability is obvious for this mixture model since we already have the generic identifiability for ether pairwise comparisons (mixtures of BTL models) or triplet comparisons (mixtures of MNL models with 3-slate). But now we are going to go through our method for the generic identifiability in the first parameter space. We will separately consider two cases $p_1 \neq 0.5$ and $p_1 = 0.5$. For $p_1 \neq 0.5$, we show that the equation system achieves generic identifiability. For $p_1 = 0.5$, we show that the equation system achieves generic identifiability up to reordering. A proposition is rigorously written below.

Proposition A.40. *Suppose* $n \geqslant 3$. *If* $p_1 = 0.7, p_2 = 0.3$, *then* (A.70) *has a unique solution in* $\mathbb{C}$ *for all* $(a_{2:n}, b_{2:n}) \in Q^{2n-2}_{MNL23}$ *but a set of* λ_{2n-2}*-measure zero, given by* $(\mathbf{x}, \mathbf{y}) = (\mathbf{a}, \mathbf{b})$.

That is, if $p_1 = 0.7, p_2 = 0.3$, *then the generic identifiability of two mixtures of MNL model with 2-&3-slate holds.*

Proof of Proposition A.40. To apply the results from Section 3.3, we first translate (A.70) into the following (equivalent) equation system by multiplying $(x_i + x_j)(y_i + y_j)$ or $(x_i + x_j + x_k)(y_i + y_j + y_k)$ on both sides,

$$\begin{cases} p_1 x_i(y_i + y_j) + p_2 y_i(x_i + x_j) - \eta_{i,j}(x_i + x_j)(y_i + y_j) = 0, \quad \forall i \neq j \in [n], \\ p_1 x_i(y_i + y_j + y_k) + p_2 y_i(x_i + x_j + x_k) = \\ \eta_{i,j,k}(x_i + x_j + x_k)(y_i + y_j + y_k), \quad \forall i \neq j \neq k \in [n], \\ x_1 = y_1 = 1, \\ t_{i,j}(x_i + x_j) = 1, \quad \forall i \neq j \in [n] \\ h_{i,j}(y_i + y_j) = 1, \quad \forall i \neq j \in [n] \\ t_{i,j,k}(x_i + x_j + x_k) = 1, \quad \forall i \neq j \neq k \in [n] \\ h_{i,j,k}(y_i + y_j + y_k) = 1, \quad \forall i \neq j \neq k \in [n] \end{cases} \tag{A.73}$$

Similar to the case in Section 3.4.1, we can keep translating (A.73) into the on that has coefficients as polynomials in $(\mathbf{a}, \mathbf{b}, p_1)$, by multiplying $(a_i+a_j)(b_i+b_j)$ or $(a_i+a_j+a_k)(b_i+b_j+b_k)$ on both sides. Note that the parameter space $Q^{2n-2}_{MNL23} \subset \mathbb{R}^{2n-2}_{+}$ guarantees that neither $(a_i+a_j)(b_i+b_j)$ nor $(a_i+a_j+a_k)(b_i+b_j+b_k)$ is zero, so we can further conclude that the generic identifiability of the new polynomial system is equivalent to that of (A.73), and hence equivalent to that of (A.70).

$$\begin{cases} (a_i+a_j)(b_i+b_j)(p_1x_i(y_i+y_j)+p_2y_i(x_i+x_j)) \\ \quad -(p_1a_i(b_i+b_j)+p_2b_i(a_i+a_j))(x_i+x_j)(y_i+y_j)=0,\ \forall i \neq j \in [n], \\ (a_i+a_j+a_k)(b_i+b_j+b_k)(p_1x_i(y_i+y_j+y_k)+p_2y_i(x_i+x_j+x_k)) \\ \quad -(p_1a_i(b_i+b_j+b_k)+p_2b_i(a_i+a_j+a_k))(x_i+x_j+x_k)(y_i+y_j+y_k)=0, \\ \qquad \forall i \neq j \neq k \in [n], \\ x_1=y_1=1, \\ t_{i,j}(x_i+x_j)=1, \forall i \neq j \in [n] \\ h_{i,j}(y_i+y_j)=1, \forall i \neq j \in [n] \\ t_{i,j,k}(x_i+x_j+x_k)=1, \forall i \neq j \neq k \in [n] \\ h_{i,j,k}(y_i+y_j+y_k)=1, \forall i \neq j \neq k \in [n] \end{cases} \tag{A.74}$$

In this part, we will argue the case where $n=3$ first and making use of this result to prove the cases where $n \geqslant 4$.

Case $n=3$

In this case, we consider the following subset of the original polynomial system,

$$\begin{cases} (a_i + a_j)(b_i + b_j)(p_1 x_i(y_i + y_j) + p_2 y_i(x_i + x_j)) \\ \quad -(p_1 a_i(b_i + b_j) + p_2 b_i(a_i + a_j))(x_i + x_j)(y_i + y_j) = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall (i,j) \in \{(1,2),(1,3),(2,3)\}, \\ (a_i + a_j + a_k)(b_i + b_j + b_k)(p_1 x_i(y_i + y_j + y_k) + p_2 y_i(x_i + x_j + x_k)) = \\ \quad (p_1 a_i(b_i + b_j + b_k) + p_2 b_i(a_i + a_j + a_k))(x_i + x_j + x_k)(y_i + y_j + y_k), \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall (i,j,k) \in \{(1,2,3),(2,1,3)\}, \\ x_1 = 1, \quad y_1 = 1, \end{cases} \tag{A.75}$$

We now compute its Gröbner basis via Magma.

Listing A.1: Gröbner basis of MNL models involving 2-&3-slate with fixed p

```
P<x1,x2,x3,y1,y2,y3,a1,a2,a3,b1,b2,b3>:=FreeAlgebra(Rationals(),12,"lex");

I:=ideal<P|
(a1+a2)*(b1+b2)*(7/10*x1*(y1+y2)+(1-7/10)*y1*(x1+x2))-
(7/10*a1*(b1+b2)+(1-7/10)*b1*(a1+a2))*(x1+x2)*(y1+y2),
(a1+a3)*(b1+b3)*(7/10*x1*(y1+y3)+(1-7/10)*y1*(x1+x3))-
(7/10*a1*(b1+b3)+(1-7/10)*b1*(a1+a3))*(x1+x3)*(y1+y3),
(a2+a3)*(b2+b3)*(7/10*x2*(y2+y3)+(1-7/10)*y2*(x2+x3))-
(7/10*a2*(b2+b3)+(1-7/10)*b2*(a2+a3))*(x2+x3)*(y2+y3),
(a1+a2+a3)*(b1+b2+b3)*(7/10*x1*(y1+y2+y3)+(1-7/10)*y1*(x1+x2+x3))-
(7/10*a1*(b1+b2+b3)+(1-7/10)*b1*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
(a1+a2+a3)*(b1+b2+b3)*(7/10*x2*(y1+y2+y3)+(1-7/10)*y2*(x1+x2+x3))-
(7/10*a2*(b1+b2+b3)+(1-7/10)*b2*(a1+a2+a3))*(x1+x2+x3)*(y1+y2+y3),
x1-1,
y1-1>;

GroebnerBasis(I);
```

This gives

$$\text{Bad}(\mathbf{a}_{2:3}, \mathbf{b}_{2:3}) = \begin{cases} 10 + 3a_i + 7b_i, (4 - 3a_i)b_i, 3 - a_i(4 + 7b_i), 3b_i + a_i(7 + 10b_i), & \forall i \in \{2,3\} \\ 7a_i + 3b_i, a_i + b_i, & \forall i \in \{2,3\} \\ 3a_ib_j + 7a_jb_i, 7 + 7a_j + 4b_i + 7b_j, 7b_i + a_i(7 + 4b_j + 7b_i), & \forall i \neq j \in \{2,3\} \\ (3 - 4a_i)b_j + 3a_j(1 + b_j), 10a_ib_i + 3a_jb_i + 7a_ib_j, & \forall i \neq j \in \{2,3\} \\ 3a_ib_i - 4a_jb_i - 7ajb_j, 3(1 + a_j + b_j) - 4a_i, & \forall i \neq j \in \{2,3\} \\ -3(1 + a_2)b_3 - a_3(7 + 7b_2 + 10b_3), & \\ 7 - 3(1 + a_3)b_2 - a_2(7 + 10b_2 + 7b_3), & \end{cases} \tag{A.76}$$

and the corresponding $Z(\mathbf{a}_{2:3}, \mathbf{b}_{2:4})$.

Based on Theorem 3.12, we claim by checking Assumption 3.2′ and 3.3 for (A.75) that (A.75) has a unique solution in $\mathbb{C}$ (counted with multiplicity) for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^6_{MNL}$ but a set of λ_6-measure zero.

Assumption 3.2: This is clear since $(\mathbf{x}_{1:4}, \mathbf{y}_{1:4}) = (\mathbf{a}_{1:4}, \mathbf{b}_{1:4})$ is a solution for all $(\mathbf{a}_{2:4}, \mathbf{b}_{2:4}) \in Q^4_{MNL23}$.

Assumption 3.3: Choose $(\mathbf{a}'_{1:3}, \mathbf{b}'_{1:3}) = (2, 3; 5, 4)$. It is routine to check that $(\mathbf{a}'_{2:3}, \mathbf{b}'_{2:3}) \notin Z(\mathbf{a}_{2:3}, \mathbf{b}_{2:4})$ using (3.38). We can use Magma code in Listing A.1. to check that (A.75) has exactly 1 solution in $\mathbb{C}$ (counted with multiplicity) for this $(\mathbf{a}'_{2:3}, \mathbf{b}'_{2:3})$.

Listing A.2: Dimension and degree Computations of MNL models involving 2-&3-slate with fixed p

```
a:=[1,2,3];
b:=[1,5,4];
p1:=7/10;

k:=Rationals();
A<x1,x2,x3,y1,y2,y3>:=AffineSpace(k,6);
P:=Scheme(A,[
(a[1]+a[2])*(b[1]+b[2])*(p1*x1*(y1+y2)+(1-p1)*y1*(x1+x2))-
(p1*a[1]*(b[1]+b[2])+(1-p1)*b[1]*(a[1]+a[2]))*(x1+x2)*(y1+y2),
(a[1]+a[3])*(b[1]+b[3])*(p1*x1*(y1+y3)+(1-p1)*y1*(x1+x3))-
```

```
(p1*a[1]*(b[1]+b[3])+(1-p1)*b[1]*(a[1]+a[3]))*(x1+x3)*(y1+y3),
(a[2]+a[3])*(b[2]+b[3])*(p1*x2*(y2+y3)+(1-p1)*y2*(x2+x3))-
(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3]))*(x2+x3)*(y2+y3),
(a[1]+a[2]+a[3])*(b[1]+b[2]+b[3])*(p1*x1*(y1+y2+y3)+(1-p1)*y1*(x1+x2+x3))-
(p1*a[1]*(b[1]+b[2]+b[3])+(1-p1)*b[1]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
(a[1]+a[2]+a[3])*(b[1]+b[2]+b[3])*(p1*x2*(y1+y2+y3)+(1-p1)*y2*(x1+x2+x3))-
(p1*a[2]*(b[1]+b[2]+b[3])+(1-p1)*b[2]*(a[1]+a[2]+a[3]))*(x1+x2+x3)*(y1+y2+y3),
x1-1,
y1-1]);

Dimension(P);
-> 0
Degree(P);
-> 1
```

Therefore, applying Theorem 3.12 concludes the proof for $n = 3$.

Case $n \geqslant 4$

From the case where $n = 3$, we know there exists a measure zero set N^4_{MNL23} such that for $(\mathbf{a}_{2:3}, \mathbf{b}_{2:3}) \in Q^4_{MNL23} \backslash N^4_{MNL23}$. When we consider the case for $n \geqslant 4$, we can first restrict our vision in a set $N_{3,n}$, where

$$N_{3,n} = \{(\mathbf{a}_{2:n}, \mathbf{b}_{2:n}) \in Q^{2n-2}_{MNL23} \mid (\mathbf{a}_{2:3}, \mathbf{b}_{2:3}) \notin N^4_{MNL23}\}.$$

Now we determine (x_i, y_i) for each $i \geqslant 4$. For this we consider the following part of the origin polynomial system which contains (x_i, y_i) as the only undetermined variables.

$$\begin{cases} (1-\eta_{1,i})x_1y_1 + (p_2-\eta_{1,i})y_1x_i + (p_1-\eta_{1,i})x_1y_i - \eta_{1,i}x_iy_i = 0, \\ (1-\eta_{2,i})x_2y_2 + (p_2-\eta_{2,i})y_2x_i + (p_1-\eta_{2,i})x_2y_i - \eta_{2,i}x_iy_i = 0, \\ (1-\eta_{3,i})x_3y_3 + (p_2-\eta_{3,i})y_3x_i + (p_1-\eta_{3,i})x_3y_i - \eta_{3,i}x_iy_i = 0. \end{cases} \tag{A.77}$$

We first plug into $x_1 = y_1 = 1, x_2 = a_2, y_2 = b_2, x_3 = a_3, y_3 = b_3$ and then cancel x_iy_i terms using the first two equations and the last two equations to get two linear equations,

$$\eta_{2,i}(1-\eta_{1,i}) + (\eta_{2,i}-1)a_2b_2 + (\eta_{2,i}-\eta_{1,i})\,p_2x_i + (\eta_{2,i}-\eta_{1,i})\,p_1y_i = 0,$$
$$(\eta_{3,i}(1-\eta_{2,i})a_2b_2 - \eta_{2,i}(1-\eta_{3,i})a_3b_3) + (\eta_{3,i}(p_2-\eta_{2,i})b_2 - \eta_{2,i}(p_2-\eta_{3,i})b_3)\,x_i$$
$$+ (\eta_{3,i}(p_1-\eta_{2,i})a_2 - \eta_{2,i}(p_1-\eta_{3,i})a_3)\,y_i = 0.$$

As long as the coefficient matrix A has rank 2, we have a unique solution for x_i, y_i,

$$A = \begin{bmatrix} (\eta_{2,i}-\eta_{1,i})\,p_2 & (\eta_{2,i}-\eta_{1,i})\,p_1 \\ \eta_{3,i}(p_2-\eta_{2,i})b_2 - \eta_{2,i}(p_2-\eta_{3,i})b_3 & \eta_{3,i}(p_1-\eta_{2,i})a_2 - \eta_{2,i}(p_1-\eta_{3,i})a_3 \end{bmatrix}.$$

Note that we can plugging into a set of values for $a_2 = 2, a_3 = 3, b_2 = 5, b_3 = 4, a_i = b_i = 0$ to see that $rank(A) = 2$. Therefore, there exists a measure zero set $\tilde{N}_i$ such that $rank(A) < 2$. There are finite of the measure zero set for each $i \in \{4, \ldots, n\}$. Thus, we conclude the generic uniqueness for the cases where $n \geqslant 4$.

□

Parameter space of $(a_{2:n}, b_{2:n}, p_1)$

In this case of MNL model with 2-slate and 3-slate in parameter space $(a_{2:n}, b_{2:n}, p_1)$, we study the equation system in variables $(\mathbf{x}, \mathbf{y}, p)$,

$$\begin{cases} \forall i \neq j \in [n],\ p\dfrac{x_i}{x_i + x_j} + (1-p)\dfrac{y_i}{y_i + y_j} = \eta_{i,j}, \\ \forall i \neq j \neq k \in [n],\ p\dfrac{x_i}{x_i + x_j + x_k} + (1-p)\dfrac{y_i}{y_i + y_j + y_k} = \eta_{i,j,k}, \\ x_1 = y_1 = 1. \end{cases} \tag{A.78}$$

Proposition A.41. *If* $n \geqslant 4$*, then* (A.78) *has exactly two solutions in* $\mathbb{C}$ *for all* $(a_{2:n}, b_{2:n}, p_1) \in Q^{2n-2}_{MNL23} \times (0,1)$ *but a set of* λ_{2n-1}*-measure zero, given by* $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{a}, \mathbf{b}, p_1)$ *and* $(\mathbf{x}, \mathbf{y}, p) = (\mathbf{b}, \mathbf{a}, 1-p_1)$.

Proof of Proposition A.41. To apply the results from §3.4, we first translate (A.78) into the following (equivalent) equation system by multiplying $(x_i + x_j)(y_i + y_j)$ or $(x_i + x_j + x_k)(y_i + y_j + y_k)$ on both sides,

$$\begin{cases} px_i(y_i + y_j) + (1-p)y_i(x_i + x_j) = \\ \quad \eta_{i,j}(x_i + x_j)(y_i + y_j) = 0, \ \forall i \neq j \in [n], \\ px_i(y_i + y_j + y_k) + (1-p)y_i(x_i + x_j + x_k) = \\ \quad \eta_{i,j,k}(x_i + x_j + x_k)(y_i + y_j + y_k) = 0, \ \forall i \neq j \neq k \in [n], \\ x_1 - 1 = 0, \\ y_1 - 1 = 0, \\ t_{i,j}(x_i + x_j) = 1, \forall i \neq j \in [n] \\ h_{i,j}(y_i + y_j) = 1, \forall i \neq j \in [n] \\ t_{i,j,k}(x_i + x_j + x_k) = 1, \forall i \neq j \neq k \in [n] \\ h_{i,j,k}(y_i + y_j + y_k) = 1, \forall i \neq j \neq k \in [n] \end{cases} \tag{A.79}$$

Similar to the case in Section 3.4.1, we can keep translating (A.79) into the one, denoted by $\mathcal{P}_{MNL23,abp}$, that has coefficients as polynomials in $(\mathbf{a}, \mathbf{b}, p_1)$, by multiplying $(a_i + a_j)(b_i + b_j)$ or $(a_i + a_j + a_k)(b_i + b_j + b_k)$ on both sides. Note that the parameter space $Q^{2n-1}_{MNL23} \subset \mathbb{R}^{2n-2}_{+}$ guarantees that neither $(a_i + a_j)(b_i + b_j)$ nor $(a_i + a_j + a_k)(b_i + b_j + b_k)$ is zero, so we can further conclude the generic identifiability of $\mathcal{P}_{MNL23,abp}$ is equivalent to that of (A.79), and hence equivalent to that of (A.78).

The arguments are similar as before by checking two assumptions 3.2′, 3.3 and applying Theorem 3.12 to conclude the polynomial system has exactly 2 solutions, which means the MNL model with 2 and 3 slate is generically identifiable up to reordering. Since we have a lot of details written in the three illustrative examples in the main text, we just leave two chunks of Magma code here for checking Gröbner basis and to check the number of solutions for some specific $\mathbf{a} = [1, 2, 3, 4], \mathbf{b} = [1, 5, 4, 2]$ below.

Listing A.3: Gröbner basis of MNL models involving 2-&3-slate with variable p

```
P<p,x2,x3,x4,y2,y3,y4,a2,a3,a4,b2,b3,b4,p1>
:=FreeAlgebra(Rationals(),14,"lex");

I:=ideal<P|
(1+a2)*(1+b2)*(p*1*(1+y2)+(1-p)*1*(1+x2))-
(p1*1*(1+b2)+(1-p1)*1*(1+a2))*(1+x2)*(1+y2),
(1+a3)*(1+b3)*(p*1*(1+y3)+(1-p)*1*(1+x3))-
(p1*1*(1+b3)+(1-p1)*1*(1+a3))*(1+x3)*(1+y3),
(1+a4)*(1+b4)*(p*1*(1+y4)+(1-p)*1*(1+x4))-
(p1*1*(1+b4)+(1-p1)*1*(1+a4))*(1+x4)*(1+y4),
(a2+a3)*(b2+b3)*(p*x2*(y2+y3)+(1-p)*y2*(x2+x3))-
(p1*a2*(b2+b3)+(1-p1)*b2*(a2+a3))*(x2+x3)*(y2+y3),
(a2+a4)*(b2+b4)*(p*x2*(y2+y4)+(1-p)*y2*(x2+x4))-
(p1*a2*(b2+b4)+(1-p1)*b2*(a2+a4))*(x2+x4)*(y2+y4),
(a3+a4)*(b3+b4)*(p*x3*(y3+y4)+(1-p)*y3*(x3+x4))-
(p1*a3*(b3+b4)+(1-p1)*b3*(a3+a4))*(x3+x4)*(y3+y4),
(1+a2+a3)*(1+b2+b3)*(p*1*(1+y2+y3)+(1-p)*1*(1+x2+x3))-
(p1*1*(1+b2+b3)+(1-p1)*1*(1+a2+a3))*(1+x2+x3)*(1+y2+y3),
(1+a2+a4)*(1+b2+b4)*(p*1*(1+y2+y4)+(1-p)*1*(1+x2+x4))-
(p1*1*(1+b2+b4)+(1-p1)*1*(1+a2+a4))*(1+x2+x4)*(1+y2+y4),
(1+a2+a3)*(1+b2+b3)*(p*x2*(1+y2+y3)+(1-p)*y2*(1+x2+x3))-
(p1*a2*(1+b2+b3)+(1-p1)*b2*(1+a2+a3))*(1+x2+x3)*(1+y2+y3),
(1+a2+a4)*(1+b2+b4)*(p*x2*(1+y2+y4)+(1-p)*y2*(1+x2+x4))-
(p1*a2*(1+b2+b4)+(1-p1)*b2*(1+a2+a4))*(1+x2+x4)*(1+y2+y4)>;

GroebnerBasis(I);
```

Listing A.4: Dimension and degree computations of MNL models involving 2-&3-slate with variable p

```
a:=[1,2,3,4];
b:=[1,5,4,2];
p1:=7/10;

k:=Rationals();
A<x2,x3,x4,y2,y3,y4,p>:=AffineSpace(k,7);
P:=Scheme(A,
[
(1+a[2])*(1+b[2])*(p*1*(1+y2)+(1-p)*1*(1+x2))-
(p1*1*(1+b[2])+(1-p1)*1*(1+a[2]))*(1+x2)*(1+y2),
(1+a[3])*(1+b[3])*(p*1*(1+y3)+(1-p)*1*(1+x3))-
(p1*1*(1+b[3])+(1-p1)*1*(1+a[3]))*(1+x3)*(1+y3),
(1+a[4])*(1+b[4])*(p*1*(1+y4)+(1-p)*1*(1+x4))-
```

```
(p1*1*(1+b[4])+(1-p1)*1*(1+a[4]))*(1+x4)*(1+y4),
(a[2]+a[3])*(b[2]+b[3])*(p*x2*(y2+y3)+(1-p)*y2*(x2+x3))-
(p1*a[2]*(b[2]+b[3])+(1-p1)*b[2]*(a[2]+a[3]))*(x2+x3)*(y2+y3),
(a[2]+a[4])*(b[2]+b[4])*(p*x2*(y2+y4)+(1-p)*y2*(x2+x4))-
(p1*a[2]*(b[2]+b[4])+(1-p1)*b[2]*(a[2]+a[4]))*(x2+x4)*(y2+y4),
(a[3]+a[4])*(b[3]+b[4])*(p*x3*(y3+y4)+(1-p)*y3*(x3+x4))-
(p1*a[3]*(b[3]+b[4])+(1-p1)*b[3]*(a[3]+a[4]))*(x3+x4)*(y3+y4),
(1+a[2]+a[3])*(1+b[2]+b[3])*(p*1*(1+y2+y3)+(1-p)*1*(1+x2+x3))-
(p1*1*(1+b[2]+b[3])+(1-p1)*1*(1+a[2]+a[3]))*(1+x2+x3)*(1+y2+y3),
(1+a[2]+a[4])*(1+b[2]+b[4])*(p*1*(1+y2+y4)+(1-p)*1*(1+x2+x4))-
(p1*1*(1+b[2]+b[4])+(1-p1)*1*(1+a[2]+a[4]))*(1+x2+x4)*(1+y2+y4),
(1+a[2]+a[3])*(1+b[2]+b[3])*(p*x2*(1+y2+y3)+(1-p)*y2*(1+x2+x3))-
(p1*a[2]*(1+b[2]+b[3])+(1-p1)*b[2]*(1+a[2]+a[3]))*(1+x2+x3)*(1+y2+y3),
(1+a[2]+a[4])*(1+b[2]+b[4])*(p*x2*(1+y2+y4)+(1-p)*y2*(1+x2+x4))-
(p1*a[2]*(1+b[2]+b[4])+(1-p1)*b[2]*(1+a[2]+a[4]))*(1+x2+x4)*(1+y2+y4)
]);

Dimension(P);
-> 0

Degree(P)
-> 2
```

□

REFERENCES

Adamczak, R. 2015. A note on the Hanson-Wright inequality for random vectors with dependencies. *Electronic Communications in Probability* 20.

Arora, Sanjeev, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *International conference on machine learning*, 280–288.

Atiyah, M. F., and I. G. Macdonald. 1969. *Introduction to commutative algebra*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont.

Awasthi, Pranjal, Avrim Blum, Or Sheffet, and Aravindan Vijayaraghavan. 2014. Learning mixtures of ranking models. In *Advances in neural information processing systems*, 2609–2617.

Balabanović, Marko, and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. *Communications of the ACM* 40(3):66–72.

Bosma, Wieb, John Cannon, and Catherine Playoust. 1997. The Magma algebra system. I. The user language. *J. Symbolic Comput.* 24(3-4):235–265. Computational algebra and number theory (London, 1993).

Bradley, Ralph Allan, and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39(3/4):324–345.

Cadamuro, G., R. Gilad-Bachrach, and X. Zhu. 2016. Debugging machine learning models. In *Icml workshop on reliable machine learning in the wild*.

Caniglia, L., A. Galligo, and J. Heintz. 1991. Equations for the projective closure and effective Nullstellensatz. vol. 33, 11–23. Applied algebra, algebraic algorithms, and error-correcting codes (Toulouse, 1989).

Chakraborty, A., M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Chen, Xi, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. 2013. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth acm international conference on web search and data mining*, 193–202.

Chen, Yuxin, and Changho Suh. 2015. Spectral mle: Top-k rank aggregation from pairwise comparisons. In *International conference on machine learning*, 371–380.

Chierichetti, Flavio, Anirban Dasgupta, Ravi Kumar, and Silvio Lattanzi. 2015. On learning mixture models for permutations. In *Proceedings of the 2015 conference on innovations in theoretical computer science*, 85–92.

Chierichetti, Flavio, Ravi Kumar, and Andrew Tomkins. 2018. Learning a mixture of two multinomial logits. In *International conference on machine learning*, 960–968.

Cox, David A., John Little, and Donal O'Shea. 2015. *Ideals, varieties, and algorithms*. 4th ed. Undergraduate Texts in Mathematics, Springer, Cham. An introduction to computational algebraic geometry and commutative algebra.

Feng, Jiashi, Huan Xu, Shie Mannor, and Shuicheng Yan. 2014. Robust logistic regression and classification. *Advances in neural information processing systems* 27:253–261.

Fergus, Rob, Yair Weiss, and Antonio Torralba. 2009. Semi-supervised learning in gigantic image collections. In *Nips*, vol. 1, 2. Citeseer.

Fortuna, Elisabetta, Patrizia Gianni, and Barry Trager. 2001. Degree reduction under specialization. vol. 164, 153–163. Effective methods in algebraic geometry (Bath, 2000).

Foygel, R., and L. Mackey. 2014. Corrupted sensing: Novel guarantees for separating structured signals. *IEEE Transactions on Information Theory* 60(2):1223–1247.

Ge, Yang. 2008. Bayesian inference with mixtures of logistic regression: Functional approximation, statistical consistency and algorithmic convergence. Ph.D. thesis, Northwestern University.

Görtz, Ulrich, and Torsten Wedhorn. 2020. *Algebraic geometry I. Schemes—with examples and exercises*. Springer Studium Mathematik—Master, Springer Spektrum, Wiesbaden. Second edition.

Gunning, Robert C., and Hugo Rossi. 2009. *Analytic functions of several complex variables*. AMS Chelsea Publishing, Providence, RI. Reprint of the 1965 original.

Hartshorne, Robin. 1977. *Algebraic geometry*. Graduate Texts in Mathematics, No. 52, Springer-Verlag, New York-Heidelberg.

Henderson, H. V., and S. R. Searle. 1981. On deriving the inverse of a sum of matrices. *Siam Review* 23(1):53–60.

Hoeffding, W. 1994. Probability inequalities for sums of bounded random variables. In *The collected works of wassily hoeffding*, 409–426. Springer.

Horn, R. A., and C. R. Johnson. 1994. *Topics in matrix analysis*. Cambridge University Press.

Huber, P.J., and E.M. Ronchetti. 2011. *Robust statistics*. Wiley Series in Probability and Statistics, Wiley.

Hwang, S.-G. 2004. Cauchy's interlace theorem for eigenvalues of Hermitian matrices. *The American Mathematical Monthly* 111(2):157–159.

Iannario, Maria. 2010. On the identifiability of a mixture model for ordinal data. *Metron* 68(1):87–94.

Jamieson, Kevin G, and Robert Nowak. 2011. Active ranking using pairwise comparisons. In *Advances in neural information processing systems*, 2240–2248.

Khetan, Ashish, and Sewoong Oh. 2016. Data-driven rank breaking for efficient rank aggregation. In *International conference on machine learning*, 89–98. PMLR.

Li, Yuanzhi, and Yingyu Liang. 2018. Learning mixtures of linear regressions with nearly optimal complexity. *arXiv preprint arXiv:1802.07895*.

Liu, Qing. 2002. *Algebraic geometry and arithmetic curves*, vol. 6 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford. Translated from the French by Reinie Erné, Oxford Science Publications.

Lu, Tyler, and Craig Boutilier. 2014. Effective sampling and learning for mallows models with pairwise-preference data. *The Journal of Machine Learning Research* 15(1):3783–3829.

Luce, R Duncan. 1959. *Individual choice behavior: A theoretical analysis*. Wiley.

McLachlan, Geoffrey J, and Kaye E Basford. 1988. *Mixture models: Inference and applications to clustering*, vol. 38. M. Dekker New York.

Meinshausen, N., and B. Yu. 2009. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics* 37(1): 246–270.

Nguyen, N. H., and T. D. Tran. 2013. Robust Lasso with missing and grossly corrupted observations. *IEEE Transactions on Information Theory* 4(59):2036–2058.

Nitsure, Nitin. 2005. Construction of Hilbert and Quot schemes. In *Fundamental algebraic geometry*, vol. 123 of *Math. Surveys Monogr.*, 105–137. Amer. Math. Soc., Providence, RI.

Oh, Sewoong, and Devavrat Shah. 2014. Learning mixed multinomial logit model from ordinal data. In *Advances in neural information processing systems*, 595–603.

Okamoto, Masashi. 1973. Distinctness of the eigenvalues of a quadratic form in a multivariate sample. *Ann. Statist.* 1:763–765.

Pendergrass, Robert N, and Ralph A Bradley. 1959. Ranking in triple comparisons. Tech. Rep., VIRGINIA AGRICULTURAL EXPERIMENT STATION BLACKSBURG.

Plackett, Robin L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24(2):193–202.

Ravikumar, P., M. J. Wainwright, and J. D. Lafferty. 2010. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics* 38(3):1287–1319.

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 acm conference on computer supported cooperative work*, 175–186.

Reynolds, Douglas A. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741.

Rousseeuw, Peter J, and Katrien Van Driessen. 2006. Computing lts regression for large data sets. *Data mining and knowledge discovery* 12(1): 29–45.

Sasai, T., and H. Fujisawa. 2020. Robust estimation with Lasso when outputs are adversarially contaminated. *arXiv preprint arXiv:2004.05990*.

Seber, G. A. F. 2008. *A matrix handbook for statisticians*, vol. 15. John Wiley & Sons.

She, Y., and A. B. Owen. 2011. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association* 106(494): 626–639.

Slawski, M., and E. Ben-David. 2017. Linear regression with sparsely permuted data. *arXiv preprint arXiv:1710.06030*.

Tang, Y., J.-P. P. Richard, and J. C. Smith. 2016. A class of algorithms for mixed-integer bilevel min–max optimization. *Journal of Global Optimization* 66(2):225–262.

Thurstone, Louis L. 1927. A law of comparative judgment. *Psychological review* 34(4):273.

Train, Kenneth E. 2009. *Discrete choice methods with simulation*. Cambridge university press.

Tsukida, Kristi, and Maya R Gupta. 2011. How to analyze paired comparison data. Tech. Rep., Washington Univ Seattle Dept of Electrical Engineering.

Veit, Andreas, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 839–847.

Vershynin, R. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

———. 2018. *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge University Press.

Wainwright, M. J. 2009. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory* 55(5):2183–2202.

———. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48. Cambridge University Press.

Wauthier, Fabian, Michael Jordan, and Nebojsa Jojic. 2013. Efficient ranking from pairwise comparisons. In *International conference on machine learning*, 109–117.

Wu, Rui, Jiaming Xu, Rayadurgam Srikant, Laurent Massoulié, Marc Lelarge, and Bruce Hajek. 2015. Clustering and inference from pairwise comparisons. In *Proceedings of the 2015 acm sigmetrics international conference on measurement and modeling of computer systems*, 449–450.

Xu, P., and L. Wang. 2014. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & operations research* 41:309–318.

Yi, Xinyang, Constantine Caramanis, and Sujay Sanghavi. 2014. Alternating minimization for mixed linear regression. In *International conference on machine learning*, 613–621.

Yu, Kai, Shenghuo Zhu, John Lafferty, and Yihong Gong. 2009. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd international acm sigir conference on research and development in information retrieval*, 211–218.

Zeng, B., and Y. An. 2014. Solving bilevel mixed integer program by reformulations and decomposition. *Optimization Online* 1–34.

Zhang, Xuezhou, Xiaojin Zhu, and Stephen Wright. 2018. Training set debugging using trusted items. In *Proceedings of the aaai conference on artificial intelligence*, vol. 32.

Zhao, Zhibing, Peter Piech, and Lirong Xia. 2016. Learning mixtures of plackett-luce models. In *International conference on machine learning*, 2906–2914.

Zhao, Zhibing, and Lirong Xia. 2018. Composite marginal likelihood methods for random utility models. In *International conference on machine learning*, 5922–5931. PMLR.

Zorich, Vladimir A. 2016. *Mathematical analysis. II*. 2nd ed. Universitext, Springer, Heidelberg. Translated from the fourth and the sixth corrected (2012) Russian editions by Roger Cooke and Octavio Paniagua T.