Title

Decision Making on Noisy Data with Additional Knowledge

Author

Ye, Yuting

Publication Date

2021

Peer reviewed|Thesis/dissertation

Decision Making on Noisy Data with Additional Knowledge

by

Yuting Ye

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Biostatistics

and the Designated Emphasis

in

Computational and Genomic Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Haiyan Huang, Co-chair
Professor Peter J. Bickel, Co-chair
Professor Peng Ding

Summer 2021

Decision Making on Noisy Data with Additional Knowledge

Abstract

Decision Making on Noisy Data with Additional Knowledge

by

Yuting Ye

Doctor of Philosophy in Biostatistics

and the Designated Emphasis in

Computational and Genomic Biology

University of California, Berkeley

Professor Haiyan Huang, Co-chair

Professor Peter J. Bickel, Co-chair

This dissertation addresses two statistical problems of dealing with noisy data with the aid of additional knowledge. My purpose is to highlight that in the era of big data, there is an increasing number of complicated problems with low signal-to-noise ratio, which cannot be simply solved by existing statistical or machine learning methods. For instance, biological data is notorious for its limited sample size but a substantial number of features (a typical $p \gg n$ problem). Fortunately, there is always additional knowledge from experts or insights that can be employed to devise smart methods to tackle these noisy data.

Chapter 2 discusses my work supervised by Professor Haiyan Huang on the hierarchical multi-label classification. This project is motivated by automatic disease diagnosis, where we aim to predict the patient's status with limited samples in each disease. The structural information that depicts the relationship between diseases can mitigate the low signal-to-noise-ratio issue. We introduce a new statistic called multidimensional-local-precision-rate (mLPR) for each object in each class. We show that classification decisions made by simply sorting objects across classes, in the descending order of mLPRs, can in theory ensure the class hierarchy and meanwhile leading to the maximization of CATCH, a pre-defined performance metric related to the area under a hit curve. In practical implementation, we need to estimate mLPRs from data. Ranking the objects across classes in the descending order of estimated mLPRs,

however, would not ensure the optimization of CATCH and/or the class hierarchy anymore. In response to this, we introduce a new ranking algorithm called HierRank, which optimizes an empirical version of CATCH defined based on the estimated mLPRs. The ranking results from HierRank are ensured to satisfy the hierarchical constraint. The superior performance of our approach over state-of-art methods in literature is demonstrated with a synthetic dataset and two real datasets.

Chapter 3 discusses my work supervised by Professor Peter J. Bickel on the binomial mixture model with the U-shape constraint under the regime that the binomial size m can be relatively large compared to the sample size n. This project is motivated by the GeneFishing method (Liu et al., 2019), whose output is a combination of the parameter of interest and the subsampling noise. To tackle the noise in the output, we utilize the observation that the density of the output has a U shape and model the output with the binomial mixture model under a U shape constraint. We first analyze the estimation of the underlying distribution F in the binomial mixture model under various conditions for F. Equipped with these theoretical understandings, we propose a simple method Ucut to identify the cutoffs of the U shape and recover the underlying distribution based on the Grenander estimator. It has been shown that when $m = \Omega(n)$, the identified cutoffs converge at the rate $O(n^{-1/3})$. The L_1 distance between the recovered distribution and the true one decreases at the same rate. To demonstrate the performance, we apply our method to varieties of simulation studies, a GTEX dataset used in (Liu et al., 2019) and a single cell dataset from Tabula Muris.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, most profound thanks to my advisors Professor Peter J. Bickel and Professor Haiyan Huang, whose generous help makes this dissertation possible. Throughout countless meetings in their offices and via Zoom over the past six years, it has been a pleasure and honor to work with them. Peter is one of the most ingenious statisticians I have ever met. I am often impressed by his unlimited creative ideas and his encyclopedia-size knowledge. I cannot forget numerous moments when I worked out a puzzle with excitement and surprise by following his ideas that appeared to be abstract at first glance or reading old literature recommended by him. Haiyan has a great gift and significant experience in identifying interesting problems from biology or pharmacogenomics data, then developing methods to solve them with statistics in an original style. I still remember how astonished I was when Haiyan pointed a phenomenon in the data that turned out to be a breakthrough of the project. Both of them have provided me with tremendous support and patience along the journey. They are amazing advisors, offering sharp insights to many questions, having great tastes in research, and showing remarkable passions in numerous statistical fields. Besides advising me academically, Peter and Haiyan have also been my life mentors. Peter often invited me to lunch, sharing his stories in career and life. Haiyan has provided countless thoughtful suggestions on my career path with her own experiences. Both of them are willing to spend time discussing trade-offs in my career choice.

I would also like to express my thanks to faculty and staff members in Biostatistics and Statistics. Lexin Li, a committee member in my qualifying exam and one of my collaborators, has given constructive suggestions on my projects in brain networks. He offered much help in my career development as well. Peng Ding, a committee member in my dissertation and the instructors of STAT 230A for which I worked as a GSI, has equipped me with profound knowledge in linear models and causal inference. His intelligence and vast knowledge in statistical history have laid a significant impact on my vision. I want to thank Yi Ma, Bin Yu, Michael I. Jordan for offering excellent group meetings, from which I acquired various edging knowledge in statistics, machine learning, deep learning. The discussions with them were always very inspiring. I also learned a lot from lectures and courses given by many professors at Berkeley, including Michael I. Jordan, Chris Paciorek, Bin Yu, Martin J. Wainwright, Ben Recht, Sandrine Dudoit, Mark van der Laan, Alan Hubbard, Maya Peterson, Jitendra Malik, Alexei A. Efros, Sam Pimentel, Sergey Levine, Jennifer Listgarten, Venkatachalam Anantharam, Yi Ma. Course instructors for which I worked as a GSI also helped me improve my teaching ability, including Steve Selvin, Noureddine El Karoui, Peng Ding, Haiyan Huang, Adam Lucas. Staff members at the division of Biostatistics, the department of Statistics, and the center for computational biology have also supported me throughout my graduate study. Sharon Norris, La Shana Porlaris, Sumaiya Elahi, Kate Chase are kind and always ready to help with my questions and problems.

Next, my gratitude goes to my collaborators in various projects during my Ph.D. career. My first research was motivated by Jingyi Jessica Li, my undergraduate advisor during my visit to UCLA in 2014, who taught me statistical knowledge and principled methodology in doing research. My collaboration with Lexin Li and Yin Xia aroused my interest in brain networks. This interest was further satisfied when exploring major psychiatric disorders using brain networks in cooperation with Miao Chang and Fei Wang. Outside the academy, Hongxia Yang, Xuwu Wang, Chao Jiang, Jingren Zhou, Kunyang Jia, Yanghua Xiao, Feng Wang, Yang Yao, Mingyang Yin, Chao Yang, Wenyang Liu, Cathy Jiao have exposed me to the research in the industry during my internship at Alibaba and Amazon. There are also beautiful experiences in collaborating with my peer fellows Zhiyue Hu, Calvin Chi, Zoe Vernon on pharmacogenomics projects, and my friends Lihua Lei, Wenpin Tang, Cheng Ju, Da Xu on a variety of machine learning projects. Many thanks to other excellent collaborators who have provided invaluable feedback, including Bin Chen, Christine Ho, Ci-Ren Jiang, Wayne Tai Lee, Chuanwei Ruan.

I am fortunate to have known many academic mentors and friends at Berkeley and outside Berkeley, who have witnessed my ups and downs in research. At Berkeley, I enjoyed the interactions with Jianbo Chen, Xiao Li, Yu Wang, Jason Wu, Ke Liu, Xiang Lyu, Yun Zhou, Boying Gong, Chi Zhang, Yaoyang Zhang, Yuqing Gao, Wei Ni, Lei Kang, Jonathan Levy, Suzanne Dufault, Ivana Malenica, Yuansi Chen, Hanzhong Liu, Yuting Wei, Yumeng Zhang, Feng Ruan. Outside Berkeley, I benefited a lot from the discussions with Xinwei Zhang, Cheng Meng, Chen Hu, Weizhou Yue, Junfeng Liu, Wei Li, Xin Xu, Sheng Xu. Further, I am indebted to my close friends, including Shu Wu, Jianyu Han, Shiwei Qin, Dongqi Shou, Shouqi Zhou, Zhiguo Zhou, Lei Jin, Yangyiman Shalen Fu, Xiaoqian Deng, Fan Huang, Hui Xiang, Ao Sun, Jiangming Xiang, Xiongyi Zhang, Haocheng Yu, Peicong Zhou. Last but not least, I would like to give special thanks to Shengxian Wang for her support in the majority of my Ph.D. career.

I owe the most to my family. Thank my parents, Xiping Guo and Xiaoyong Ye, for their sustainable support despite thousands of distance between them and me since 2011. I could not have made any achievements without their endless love and support.

Chapter 1

Introduction

With the advent of the data deluge since the 1990s, the term "big data" has been catching the attention of the whole world (Cai and Zhu, 2015). Even my aunt is constantly talking about this concept recently, who can hardly turn on and turn off a desktop with Windows 10. But it is more than a fancy phrase used by social media to play to the gallery. Big data has become entrenched in various scientific disciplines, e.g., computer science, statistics, economics, biology, public health, to name a few. As a statistician whose essential mission is "the reduction of data" (Fisher, 1922), I am fortunate to have numerous opportunities to deal with large volumes of data collected in a loose form or structured in good shape.

My journey to the mysterious world of data started on a bioinformatics project about isoform selection when I was a visiting college student at the University of California, Los Angeles, in 2014. Since then, I have seen a wide and diverse variety of data, including DNA microarray (Nuwaysir et al., 2002), bulk RNA sequencing data (Wang, Gerstein, and Snyder, 2009), single-cell RNA sequencing data (scRNA-seq) (Haque et al., 2017), multiomics data (Vilanova and Porcar, 2016; Hasin, Seldin, and Lusis, 2017), brain networks measured by functional Magnetic resonance imaging (MRI) (Buckner, Krienen, and Yeo, 2013) or diffusion MRI (Sporns, 2013), electronic health record (Heart, Ben-Assuli, and Shabtai, 2017), etc. These data share one characteristic in common — there might be insufficient samples, but each sample has numerous features, e.g., genes, brain voxels. It is the typical high-dimensional regime where the number of features exceeds the number of objects by far.

The primary challenge in tackling the above data is to address the low signal-to-noise-ratio (SNR) issue. On the one hand, the inadequacy of samples refrains the signal from standing out amongst the noise. On the other hand, some of these data are intrinsically noisy per se. For instance, scRNA-seq suffers from a high level of technical noise than bulk RNA-seq data due to bias of transcript coverage, low capture efficiency, and limited sequencing coverage (Kolodziejczyk et al., 2015). Brain networks measured by functional MRI often see false discoveries or false negatives because of head motion, physiologic noise, and neurovascular

uncoupling (Silva et al., 2018). Off-the-shelf statistical and machine learning tools for the high-dimension regime, such as dimension reduction methods and sparse regression methods, can be leveraged as an initial means to look into these data. Nonetheless, the chances are that they can barely reveal interesting discoveries due to the lack of considerations into the characteristics of the data.

It usually demands additional knowledge about the data to expose the secrets hidden in the data, especially when SNR is low. For instance, the multiomics area is established on the integrative analysis of distinct biological researches, including genomics, genetics, proteomics, and metabolomics. Even though limited by the number of subjects in the genomics study, a comprehensive assessment of the complex diseases is accessible by borrowing knowledge from other measurements. Another example is the systematic disease diagnosis. Modern medicine has been accumulating understating of the associations between diseases, which enables the physicians to make decisions with the aid of experts from other departments. The same principle can be applied when developing an automatic disease diagnosis system based on the structure of diseases (Huang, Liu, and Zhou, 2010).

In most situations, however, auxiliary data is not easily available. We have to perform exploratory data analysis (EDA) to dig out information useful for modeling or further analysis. In John W. Tukey's book "Exploratory Data Analysis", he held that too much emphasis was put on statistical hypothesis analysis; more emphasis needed to be put on using data to suggest hypotheses to test. In particular, the objectives of EDA includes:

(1) Suggest hypotheses about the causes of observed phenomena.

(2) Assess assumptions on which statistical inference will be based.

(3) Support the selection of appropriate statistical tools and techniques.

(4) Provide a basis for further data collection through surveys or experiments.

In my research experience, EDA plays a broader role than the above four points. For instance, upon obtaining new data, I'm accustomed to first visualizing the data using box plot, histogram, scatter plot, etc. Then, if the data contains too many features, some dimension reduction tools are employed like principle component analysis (Pearson, 1901; Hotelling, 1933), multidimensional scaling (Mead, 1992) or t-distributed stochastic neighbor embedding (Maaten and Hinton, 2008). After such explorations, which usually take long, I can get some idea of what potential problem can be answered by this data, how to formulate the problem, and what characteristics this data possesses.

In my dissertation, I will study two problems where additional knowledge plays an essential role in analyzing the data of low SNR. The knowledge of the first study is given in advance, while EDA discovers that of the second one. Specifically, the first study concentrates on the

hierarchical multi-label classification (HMC) problem, whose data is comprised of features, labels, and the structure that describes the relationship between labels. One intriguing example of this kind is the disease diagnosis problem. Given the patients' disease status and their microarray measurements, we manage to design an automatic disease diagnosis based on a subject's microarray measurement. For each disease, there are only hundreds of patients but dozens of thousands of genes. To improve the diagnosis performance, we incorporate the relationship structure of diseases into this study. The second study deals with the output of the GeneFishing method (Liu et al., 2019), which is applied to high-dimensional data with limited samples but a substantial number of features. Although the output is one-dimensional, it suffers from measurement error because the observation is a combination of the parameter of interest and a binomial random variable that depends on the parameter. Our goal is to help make decisions on the GeneFishing method. To this end, we utilize the information we obtain from the histograms that the underlying distribution of the GeneFishing output appears in a U shape. I sketch the two works in each of the following subsections, respectively.

1.1 Background of Chapter 2: Disease Diagnosis

With the rapid development of machine learning and statistics, data mining and analysis becomes increasingly useful to ease the burden of human beings in all kinds of fields, including face recognition, voice/text translation, anomaly detection/prediction of machines, to name a few celebrated examples. In the past decades, much attention has been paid to automating the disease diagnosis process to help physicians diagnose simple diseases (Shen et al., 2019). It has a significant impact on both the academic and social aspects since medical resources are never close to abundance. Our dedication to the field of automatic disease diagnosis originated from Huang, Liu, and Zhou (2010). They developed a classifier for predicting disease along with the Unified Medical Language System (UMLS) directed acyclic graph, trained on public microarray datasets from the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO).

GEO was initially founded in 2000 to systematically catalog the growing volume of data produced in microarray gene expression studies. GEO data typically comes from research experiments where scientists are required to make their data available in a public repository by a grant or journal guidelines. In July 2008, GEO contained 421 human gene expression studies on the three selected microarray platforms (Affymetrix HG-U95A (GPL91), HG-U133A (GPL96), and HG-U133 Plus 2 (GPL570)). In Huang, Liu, and Zhou (2010), 100 studies were collected, yielding a total of 196 datasets for training the classifier.

Labels for each dataset were obtained by mapping text from descriptions on GEO to concepts in the UMLS, an extensive vocabulary of concepts in the biomedical field organized as a directed acyclic graph. The mapping resulted in a directed acyclic graph of 110 concepts

matched to the 196 datasets at two levels of similarity – a match at the GEO submission level and a match at the dataset level, with the latter being a stronger match. The disease concepts and their GEO matches are listed in Table S2 in the supplementary information for Huang, Liu, and Zhou (2010).

Challenge. The task of automatic disease diagnosis is challenging because there might be a deficiency of samples in some diseases compared to the number of genes in the genomics data. Consequently, the learner may have distinct powers and abilities to control the false discovery rate across diseases. Moreover, some diseases can have an extreme imbalance issue, which means that the number of negative samples is way more than that of positive ones. Both issues lead to a relatively low SNR that impedes the patients from trusting the diagnosis results output by the machine.

Additional Knowledge. Fortunately, diseases do not exist separately from one another. We obtain knowledge of how one disease connects to others from UMLS. For the diseases studied in Huang, Liu, and Zhou (2010), the full hierarchy graph of the diseases is partitioned into two parts as respectively shown in Figures 2.9 and 2.10 (in Chapter 2, Section 2.7.3). With such knowledge, we can increase the effective sample size for each disease by resorting to other related diseases. Then it becomes a hierarchical multi-label classification problem.

Our solution. In Chapter 2, we perform the hierarchical multi-label classification in two stages. We follow the Bayesian approach used in Huang, Liu, and Zhou (2010) to train the first-stage classifier and place our focus on the second stage. We introduce a new statistic called multidimensional-local-precision-rate (mLPR) for each object in each class. Under a Bayesian setting, mLPR in HMC is analogous to multidimensional-local-true-discovery-rate (mltdr) in hierarchical hypothesis testing (HHT). We show that classification decisions made by simply sorting objects across classes, in the descending order of mLPRs, can in theory ensure the class hierarchy and meanwhile leading to the maximization of CATCH, a pre-defined performance metric related to the area under a hit curve. In practical implementation, we need to estimate mLPRs from data. Ranking the objects across classes in the descending order of estimated mLPRs, however, would not ensure the optimization of CATCH and/or the class hierarchy anymore. In response to this, we introduce a new ranking algorithm called HierRank, which optimizes an empirical version of CATCH defined based on the estimated mLPRs. The ranking results from HierRank are ensured to satisfy the hierarchical constraint. The superior performance of our approach over state-of-art methods in literature is demonstrated with a synthetic dataset and two real datasets. One real dataset comes from a study of disease diagnosis using gene expression data, and the other is from a document categorization application.

This chapter is adapted from my joint work with Professor Haiyan Huang, Christine Ho, Ci-Ren Jiang, and Wayne Tai Lee. This is a follow-up work of Huang, Liu, and Zhou (2010) and Jiang et al. (2014). Huang, Liu, and Zhou (2010) first built an automated disease diagnosis system, but they ranked the first-stage classifier scores without considering the

hierarchy and comparability of these scores across classes. Jiang et al. (2014) proposed local precision rates (LPRs) that are shown to be comparable between classes when the classes are independent. We addressed the problem completely by taking into account the hierarchy and the comparability issue simultaneously. Haiyan Huang provided valuable supervision on this work while other collaborators gave many useful suggestions on the paper writing.

1.2 Motivating Example of Chapter 3: GeneFishing

In biological studies, it is pretty standard that the wet lab experiments only involve hundreds of subjects, but each subject can have tons of thousands of measurements, e.g., sequencing over dozens of thousands of genes. To handle the low SNR issue, Liu et al. (2019) proposed the GeneFishing method. Provided some knowledge involved in a biological process as "bait", GeneFishing was designed to "fish" (or identify) discoveries that are yet identified related to this process. In this work, the authors used a set of pre-identified 21 "bait genes", all of which have known roles in cholesterol metabolism, and then applied GeneFishing to three independent RNAseq datasets of human lymphoblastoid cell lines. They found that this approach identified other genes with known roles not only in cholesterol metabolism but also with high levels of consistency across the three datasets. They also applied GeneFishing to GTEx human liver RNAseq data and identified gene glyoxalase I (GLO1). In a follow-up wet-lab experiment, GLO1 knockdown increased levels of cellular cholesterol esters.

The GeneFishing procedure is as follows, as shown in Figure 1.1:

1. Split the n candidate genes randomly into many sub-search-spaces of L genes per sub-group (e.g., $L = 100$), then added to with the bait genes. This step is the key reduction of search space, facilitating making the "signal" standing out from the "noise".

2. Construct the Spearman co-expression matrices for gene-pairs contained within each sub-search-space. Apply the spectral clustering algorithm (with the number of clusters equal to 2) to each matrix separately. In most cases, the bait genes are separated from the candidate genes. But in some instances, candidate gene(s) related to the bait genes will cluster with them. When this occurs, the candidate gene is regarded as being "fished out".

3. Repeat steps 1 and 2 (defining one round of GeneFishing) m times (e.g., $m = 10,000$) to reduce the impact that a candidate gene may randomly co-cluster with the bait genes.

4. Aggregate the results from all rounds, and the i-th gene is fished out X_i times out of m. The final output records the "capture frequency rate" ($CFR_i := \hat{s}_i = X_i/m$). The "fished-out" genes with large CFR values are thought of as "discoveries". Notwithstanding,

instead of considering these "discoveries" to perform a specific or similar function as the bait genes, we only believe they are likely to be functionally related to the bait genes. Figure 1.2 displays the distribution of X_i's with $m = 10,000$ and the number of total genes $n = 21,000$ on four tissues for the cholesterol-relevant genes.

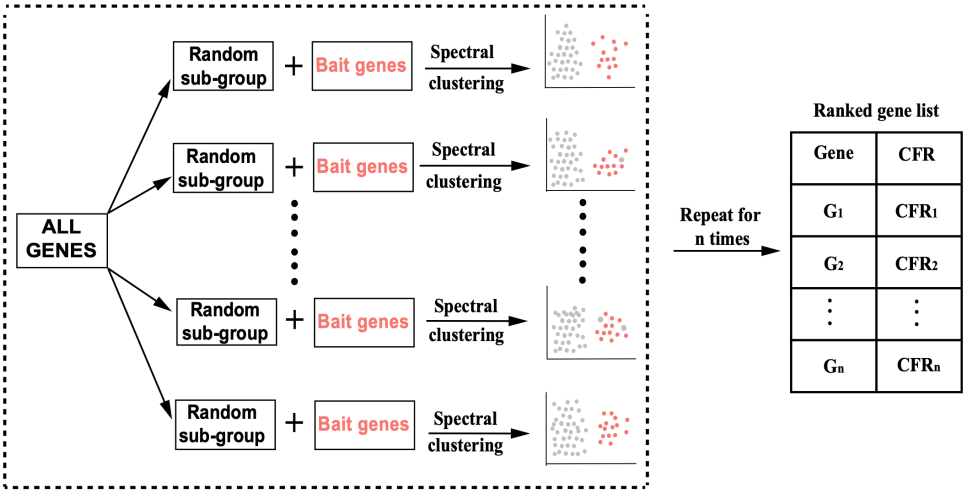


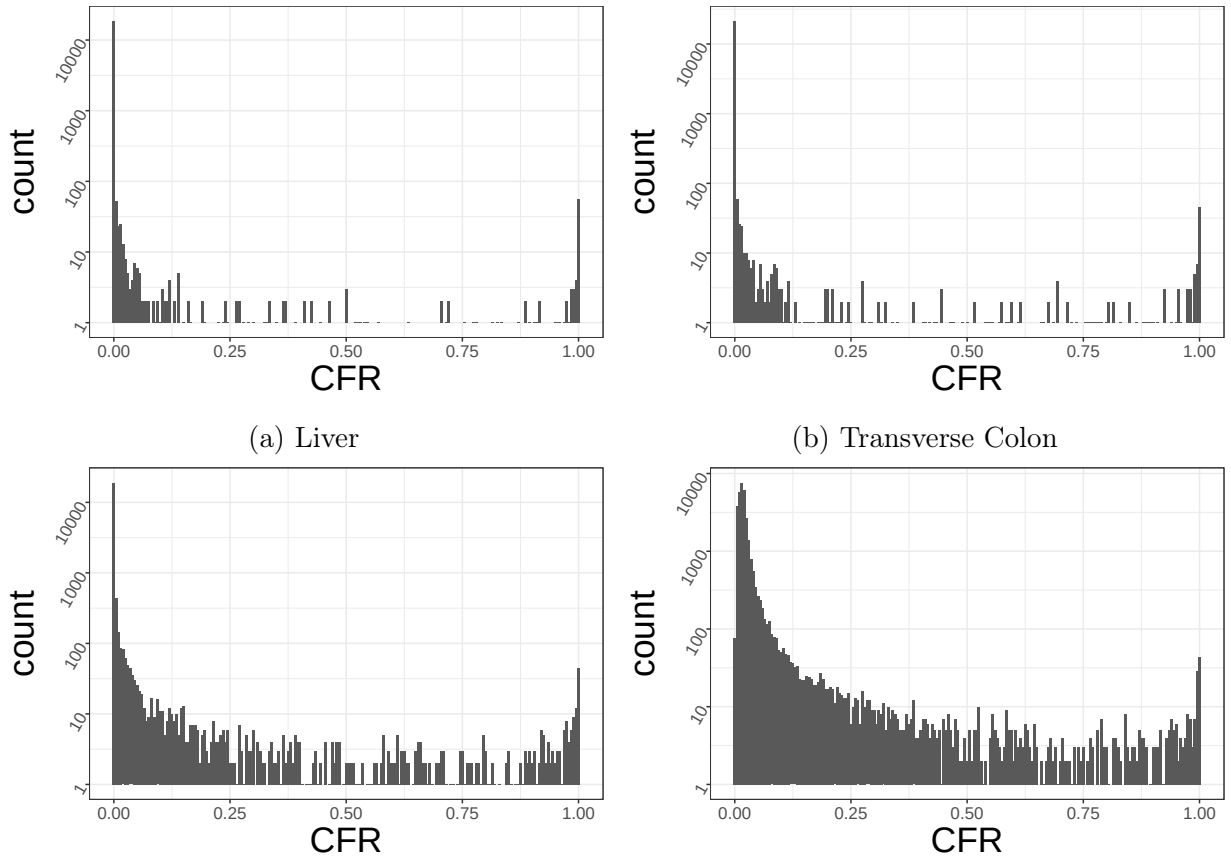
Figure 1.1: Workflow of GeneFishing (Fig 1 (e) of Liu et al. (2019)). CFR stands for Capture Frequency Rate.

Compared to other works for "gene prioritization", GeneFishing has three merits. First, it takes care of extreme inhomogeneity and low signal-to-noise ratio in high-throughput data by using dimensionality reduction by subsampling. Second, it uses clustering to identify 21 tightly clustered bait genes, which is a data-driven confirmation of the domain knowledge. Finally, GeneFishing leverages the technique of results aggregation (motivated by a bagging-like idea) in order to prioritize genes relevant to the biological process and reduce false discoveries.

Challenge. Nonetheless, there remains an open question on how large a CFR should be so that the associated gene is labeled "discovery". One difficulty results from the fact that the CFR is not exactly equivalent to the fishing rate, which reflects the extent to which one candidate gene is functionally related to the bait genes. Instead, CFR is a mix of the fishing rate and the noise induced by subsampling that hinges on the fishing rate. When we only perform a few rounds of GeneFishing (m is small) because of a limited computational budget, this noise cannot be neglected.

In literature, we have found three strategies to this end.

- Liu et al. (2019) picked a large cutoff 0.99 by eye. It is acceptable when the histograms are sparse in the middle as in the liver or the transverse colon tissues (Figure 1.2 (a)(b)),



(a) Liver

(b) Transverse Colon

(c) Artery Coronary

(d) Testis

Figure 1.2: Histograms of the CFRs on different tissues.

since cutoff= 0.25 and cutoff= 0.75 make little difference in determining the discoveries. On the other hand, for the artery coronary and testis tissues (Figure 1.2 (c)(d)), the non-trivial middle part of the histogram is a mixture of the null (irrelevant to the biological process) distribution and the alternative (relevant to the biological process) distribution. The null and the alternative are hard to separate since the middle part is quite flat, and hence we need a better-selected cutoff.

- Existing tools using parametric models to estimate local false discovery rates are applied (Gauran et al., 2018). However, these parametric models are not able to account for the middle flatness well. As a result, they tend to select a smaller cutoff and produce excessive false discoveries.

- Liu et al. (2019) also provided a permutation-like procedure to compute approximate p-values and false discovery rates (FDRs). Nonetheless, there are two problems with this procedure. On the one hand, it is substantially computationally expensive, considering another round of permutation is added on top of numerous fishing rounds. On the other hand, the permutation idea is based on a strong null hypothesis that none of the candidate non-bait genes are relevant to the bait genes, thus producing an extreme p-value or FDR, which is unrealistic.

Additional Knowledge. To resolve the cutoff selection issues mentioned above, we utilize the knowledge obtained by EDA. Note that in Figure 1.2 there exists a clear pattern that the histogram is decreasing on the left-hand side and increasing on the right-hand side for all four tissues. In the middle, liver and transverse colon display sparse densities while artery coronary and testis exhibit flat ones. Thus, we can impose a U shape constraint on the associated density of F (see Section 3.4 for details). The incorporation of such shape constraint mitigates the subsampling noise issue. And the original problem becomes finding out the cutoff where the flat middle part transits to the increasing part on the right-hand side.

Our solution. In this chapter, we use the binomial mixture model with the U-shape constraint to model the CFRs generated by GeneFishing. We first analyze the estimation of the underlying distribution F in the binomial mixture model under various conditions for F, in the regime where the binomial size m can be relatively large compared to the sample size n. Armed with these theoretical understandings, we propose a simple method for GeneFishing to identify the cutoffs of the U shape and recover the underlying distribution based on the Grenander estimator (Grenander, 1956). It has been shown that when $m = \Omega(n)$, the identified cutoffs converge at the rate $\mathcal{O}(n^{-1/3})$. The L_1 distance between the recovered distribution and the true one decreases at the same rate. The performance of our method is demonstrated with varieties of synthetic datasets, a GTEX dataset used in Liu et al. (2019) and a single cell dataset from Tabula Muris.

This chapter is adapted from my joint work with Professor Peter J. Bickel. This work was motivated by Liu et al. (2019). To the best of our knowledge, it is also the first theoretical inquiry into the binomial mixture model with a large m in a finite sample regime. Peter J. Bickel provided extensive valuable advising on this work.

Chapter 2

Decision Making for Hierarchical Multi-label Classification with Multi-dimensional Local Precision Rate

2.1 Introduction

Hierarchical multi-label classification (HMC) concerns the situation where additional knowledge of the dependency relationships among classes is available and needs to be incorporated, on top of the multi-label classification where each object is assigned to one or multiple classes (Zhang and Zhou, 2013). The class dependency in HMC is usually assumed to follow a hierarchical structure represented with a tree or a directed acyclic graph. HMC, an important problem in many applications, has recently attracted a large amount of attention in statistics and machine learning research. In biology and biomedicine, example applications of HMC include the disease diagnosis along a directed acyclic graph (DAG) composed of terms from the Unified Medical Language System (UMLS)[1]; the assignment of genes to multiple gene functional categories defined by the Gene Ontology DAG[2]; the categorization of proteins along the MIPS FunCat rooted tree[3] among others (Alves, Delgado, and Freitas, 2010; Barutcuoglu, Schapire, and Troyanskaya, 2006; Blockeel et al., 2006; Clare, 2003; Kiritchenko, Matwin, and Famili, 2005; Valentini, 2009; Valentini, 2011). Outside of biology, HMC is commonly used in text classification, music categorization, and image recognition – three fields where labels following a hierarchical structure are common.

A seminal line of HMC research has handled the problem in two stages. In the first

[1]https://www.nlm.nih.gov/research/umls/index.html
[2]http://www.webgestalt.org/2017/GOView/
[3]http://mips.gsf.de/projects/funcat

stage, classifiers are trained for each class without considering the class hierarchy, as if these are multiple independent classification problems. The task of the second stage is then to make a decision on each class for each object, given the first-stage classifier scores, the class hierarchy, and a predefined performance criterion (Koller and Sahami, 1997; Wu, Zhang, and Honavar, 2005; Holden and Freitas, 2005; Silla and Freitas, 2009; Gauch, Chandramouli, and Ranganathan, 2009). The two-stage method is popular for its flexibility — a variety of classification methods can be applied in the first stage. This first stage also tends to be computationally efficient since the class-specific classifiers can be learned in parallel. However, it remains an open question for the second stage how to balance the two essential goals of HMC: 1) respecting the given class hierarchy; 2) achieving the best possible classification performance, evaluated by metrics like accuracy, precision rate, recall rate, F-measure, etc.

Among the above two goals in the second stage, a majority of prior efforts have focused on one while paying less attention to the other or considered them as separate goals. One common approach has been to determine class-specific cutoffs on the first-stage classifier scores by optimizing an objective like H-loss or F-measure (Barutcuoglu, Schapire, and Troyanskaya, 2006; Triguero and Vens, 2016). Because these cutoffs are determined without full consideration of the hierarchical structure (sometimes they are determined independently), the decisions may not respect the hierarchy. In order to alleviate this problem, the initial decisions are adjusted to satisfy the hierarchical constraint. But this brings in another problem: these final decisions may now no longer be optimal concerning the original performance objective (Sun and Lim, 2001; Ananpiriyakul, Poomsirivilai, and Vateekul, 2014).

Another approach for the second stage is to rank the objects for their assignments in all classes, given the classifier scores of the objects in every class and the class hierarchy. Then a single cutoff on the ranking suffices to produce all the decisions. Jiang et al. (2014) described an optimal way to rank all the objects across all classes in the general multi-class problem: transforming the first-stage classifier scores to local precision rates (LPRs), then obtaining the ranking by sorting LPRs in descending order. It has been shown that the resulting ranking maximizes the pooled precision rate at any pooled recall rate. The LPR value has a nice Bayesian interpretation that it is equivalent to the local true discovery rate under certain probabilistic assumptions. However, this method lacks the consideration of the hierarchical structure. Bi and Kwok (2011) used an algorithm that maximizes the sum of top L first-stage classifier scores while respecting the hierarchy, where L is predefined. This method can produce a list of ranked events (the status of an object in a class) by varying L. Nonetheless, it might not be appropriate to directly sum up these classifier scores due to potential statistical differences among the classifiers from different classes. When not taken care of properly, such differences can lead to poor decisions; see Figure 2.2 for an illustration example. Bi and Kwok (2015) extended Bi and Kwok (2011) by introducing an algorithm to optimize some objective function (instead of the sum of top L classifier scores) under the hierarchical constraint. They provided three candidate objective / risk functions.

However, they do not suggest which risk or objective to use among the three candidates they investigated. Moreover, they do not have a clear statistical interpretation or justification on the risks and the hyperparameters involved in the three risks.

In this chapter, we introduce a new statistic called multidimensional local precision rates (mLPR), given the first-stage classifier scores, for each object in each class. Similar to LPR, mLPR has a nice Bayesian interpretation that it is equivalent to the multidimensional local true discovery rate used in hierarchical hypothesis testing (HHT) under certain probabilistic assumptions (Ploner et al., 2006). It is also demonstrated that in theory, sorting mLPRs in descending order automatically satisfies the hierarchical consistency and optimizes a new objective function, the Conditional expected Area under The Curve of the Hit curve (CATCH) given the first-stage classifier scores. In a hit curve, the x-axis represents the number of discoveries, and the y-axis represents the number of true discoveries (i.e., the hit number). The new objective function inherits the characteristic of the hit curve that the maximization of the area under the hit curve favors a large precision rate when the recall rate is small (Herskovic, Iyengar, and Bernstam, 2007). We advocate optimizing CATCH for HMC when the initial decisions are of more importance than the subsequent ones.

In practical implementation, since we can only estimate mLPRs instead of obtaining their true values, the naive sorting procedure (sorting estimated mLPRs in descending order) might fail to guarantee the optimization of CATCH or violate the hierarchy consistency. The deviation from these two goals can be significant if the data is too noisy or the sample size is limited. To this end, we develop the ranking algorithm HierRank (*Hierarchical Ranking*) that sorts the objects across classes using estimated mLPRs while obeying the hierarchical constraint. This algorithm is shown to achieve the optimization of an empirical version of CATCH given the estimated mLPRs and at the same time satisfy the hierarchical constraint. In addition, it has relatively low time complexity $\mathcal{O}(n \log n)$, where n is the number of decisions to be made, which is the product of the number of classes K and the number of samples M. Therefore, HierRank can be adapted to a large graph with numerous classes.

For evaluation, we first consider a synthetic dataset. Our method is shown to outperform other competing methods universally in terms of the truncated area under the precision-recall curve. Then, we study two real datasets. On the dataset for disease diagnosis, we show how the accuracy of the mLPR estimation influences the performance of our method. On the dataset for text classification, we emphasize the statistical differences in fitted values and predicted values of the classifier scores, which gives a practical guideline on how to train the two-stage method.

The rest of the chapter is organized as follows. In Section 2.2, we introduce the model. In Section 2.3, we introduce the concepts of hit curve and local precision rate. In Section 2.4, we introduce the performance metric CATCH and the statistic mLPR. We show that sorting mLPRs in descending order can maximize CATCH while respecting the hierarchy. In Section 2.5, we propose the ranking algorithm HierRank that sorts objects with associated estimated

mLPRs under the hierarchy constraint. We discuss an equivalent algorithm of HierRank and its extension to DAG in 2.6. We assess the performance of our method on a synthetic dataset and two case studies in Section 2.7. Finally, we conclude this chapter in Section 2.8.

2.2 Notation and Model

2.2.1 Notation

There are K classes of interest, which are structured in a hierarchy $\mathcal{G}$, e.g., Figure 2.1 (a). In $\mathcal{G}$, denote by $pa(k)$ the set of the parent nodes of the k-th node, by $anc(k)$ the set of its ancestor nodes, and by $nbh(k)$ the set of its immediate neighbors. For example, $pa(F) = \{C\}$, $anc(F) = \{A, C\}$, $nbh(F) = \{C, G, I\}$ in Figure 2.1 (a), where we abuse the node index and the node symbol when there is no ambiguity. A random object (the person or item to be classified) can possess multiple positive labels out of the K classes, thus associated with K classification decisions to be made. Throughout this chapter, we call an object's membership status in a class a classification "event". Each event has a binary variable Y indicating the corresponding class status/label of the considered object and a pre-given classifier score S reflecting the likelihood that the event is positive. Specifically, for a random object, denote by Y_k and S_k the status and the classifier score on the k-th class. We denote $\mathbf{Y} = (Y_1, \ldots, Y_K)^{\mathrm{T}}$, $\mathbf{S} = (S_1, \ldots, S_K)^{\mathrm{T}}$.

In practice, we observe M objects, and thus there are $n = K \times M$ classification events in total. We use $Y_k^{(m)}$ and $S_k^{(m)}$ to denote the status and the pre-defined classifier score for the k-th class of the m-th object. Denote $\mathbf{Y}^{(m)} = (Y_1^{(m)}, \ldots, Y_K^{(m)})^{\mathrm{T}}$ and $\mathbf{S}^{(m)} = (S_1^{(m)}, \ldots, S_K^{(m)})^{\mathrm{T}}$ for the m-th object. For ease of notation, throughout the rest of the chapter, we say Event i is an ancestor of Event i' (or equivalently, $i \in anc(i')$), if the two events i and i concern the same object and that the class node associated with Event i is an ancestor node of that of Event i'. We define the ranking $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ on n events as a permutation of $(1, 2, \ldots, n-1, n)$. We say a ranking $\boldsymbol{\pi}$ has the **hierarchical consistency** or is a **topological** ordering for $\mathcal{G}$ if it satisfies

$$\pi_i < \pi_{i'} \quad \text{if Event } i \text{ is an ancestor of Event } i'.$$

2.2.2 Model

The hierarchy $\mathcal{G}$ for the K classes is assumed to be a tree/DAG. It can be disconnected and consist of multiple connected components like Figure 2.1 (a). We mainly discuss the tree structure in this chapter. The extension to the DAG structure is discussed in Section 2.6.3.

Given the hierarchy $\mathcal{G}$, the status/label variables $\mathbf{Y}$ and the first-stage classifier scores $\mathbf{S}$ of a random individual/sample, we consider an augmented graph $\bar{\mathcal{G}}$ (Figure 2.1 (b)) by assuming the conditional independence stated in Assumption 1.

Assumption 1 *The scores are conditional independent given the associated class labels, i.e.,*

$$S_{k'} \perp S_k | (Y_k, Y_{k'}) \quad \textit{for any} \quad k' \neq k.$$

The conditional independence is reasonable for the two-stage method of HMC, because the first-stage classification training is executed class by class. With this assumption, we propose the following model $\mathcal{H}$ to characterize the relationship between $\mathbf{Y}$ and $\mathbf{S}$ of a random object:

(i) $\mathbb{P}(S_k = s | S_1, \ldots, S_{k-1}, S_{k+1}, \ldots, S_K, Y_1, \ldots, Y_K) = \mathbb{P}(S_k = s | Y_k)$.

(ii) $\mathbb{P}(Y_k = 1 | Y_{pa(k)} = 1) \in [0, 1]$; $\mathbb{P}(Y_k = 1 | Y_{pa(k)} = 0) = 0$.

(iii) S_k has a mixture model, i.e., $\mathbb{P}(S_k \leq s | Y_k) = F_0^{(k)}(s)\, \mathbb{I}(Y_k = 0) + F_1^{(k)}(s)\, \mathbb{I}(Y_k = 1)$, where $F_0^{(k)}$ denotes the null cumulative distribution function (CDF) $\mathbb{P}(S_k \leq s | Y_k = 0)$ and $F_1^{(k)}$ denotes the alternative CDF $\mathbb{P}(S_k \leq s | Y_k = 1)$ for the k-th class.

(iii') S_k has a mixture model, i.e., $\mathbb{P}(S_k \leq s) = \mathbb{P}(S_k \leq s, Y_k = 0) + \mathbb{P}(S_k \leq s, Y_k = 1) = F_0^{(k)}(s)\, \mathbb{P}(Y_k = 0) + F_1^{(k)}(s)\, \mathbb{P}(Y_k = 1)$, where $F_0^{(k)}$ denotes the null cumulative distribution function (CDF) $\mathbb{P}(S_k \leq s | Y_k = 0)$ and $F_1^{(k)}$ denotes the alternative CDF $\mathbb{P}(S_k \leq s | Y_k = 1)$for the k-th class.

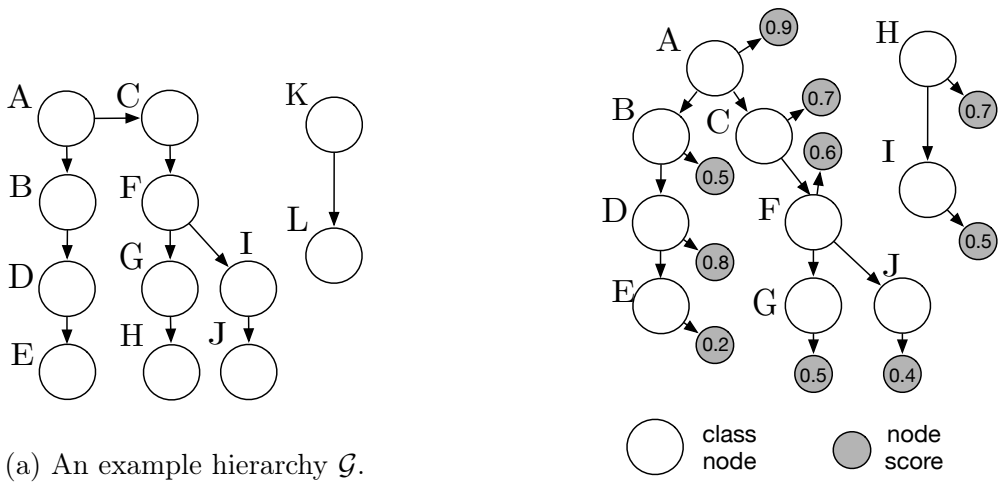
The assumption (i) follows from the conditional independence; the assumption (ii) reflects that the labels respect the hierarchy $\mathcal{G}$, i.e., a negative node implies all of its descendants are negative as well; the assumption (iii) means that the classifier score is generated from a class-specific mixture model.

On the sample level, the M objects are independent and identically distributed, i.e., $(\mathbf{Y}^{(m)}, \mathbf{S}^{(m)}) \overset{i.i.d}{\sim} \mathcal{H}$, $m = 1, \ldots, M$. For a fixed k, scores $\{S_k^{(m)} : m = 1, \ldots, M\}$ follow the same mixture distribution. But if k and k' correspond to different nodes in $\mathcal{G}$, $S_k^{(m)}$ and $S_{k'}^{(m')}$ would follow different distributions, and so are not directly comparable; see Figure 2.2 for an example. When making joint decisions across all nodes, we need to take into consideration such distinct statistical distribution properties across classes.

2.3 Preliminary

2.3.1 Hit Curve

The hit curve has been explored in the information retrieval community as a useful alternative to the ROC curve and the PR curve. In a hit curve, the x-axis represents the



(a) An example hierarchy $\mathcal{G}$.

(b) The augmented graph $\bar{\mathcal{G}}$.

Figure 2.1: An example hierarchical graph $\mathcal{G}$ and the associated augmented graph $\bar{\mathcal{G}}$.

number of discoveries, and the y-axis represents the number of true discoveries (i.e., the hit number); see Figure 2.3. The hit curve is widely used in situations where the users are more interested in the top-ranked instances. For example, in evaluating the performance of a web search engine, the relevance of the top-ranked pages is more important than those that appear lower in search results because users expect that the most relevant results appear first. The hit curve can serve well in this situation as a graphic representation of the ranker's performance since it would plot the results in order of decreasing relevance, and the y-axis would indicate the result's relevance to the target. On the other hand, the ROC curve, which plots the true positive rates (TPR) against the false positive rates (FPR) at various threshold settings, does not depend on the prevalence of positive instances (Davis and Goadrich, 2006; Hand, 2009). In the case of search results, the number of relevant pages is tiny compared to the size of the World Wide Web (i.e., low prevalence of positive instances), which would result in an almost zero FPR for the top-ranked pages. That is to say, with very few true positives, the early part of the ROC curve would fail to visualize the search ranking performance meaningfully. In the case of many hierarchical multi-label classification problems, like disease diagnosis problems, this issue exists as well; there are many candidate diseases to consider while few are actually relevant to the patient. Although the PR curve accounts for prevalence to a degree (i.e., showing the trade-off between precision and recall for different threshold), Herskovic, Iyengar, and Bernstam, 2007 provided a simple example where the hit curve can be the more informative choice: with only five positive cases out of 1000, the hit curve's shape clearly highlighted the call order of a method that had called 100 instances

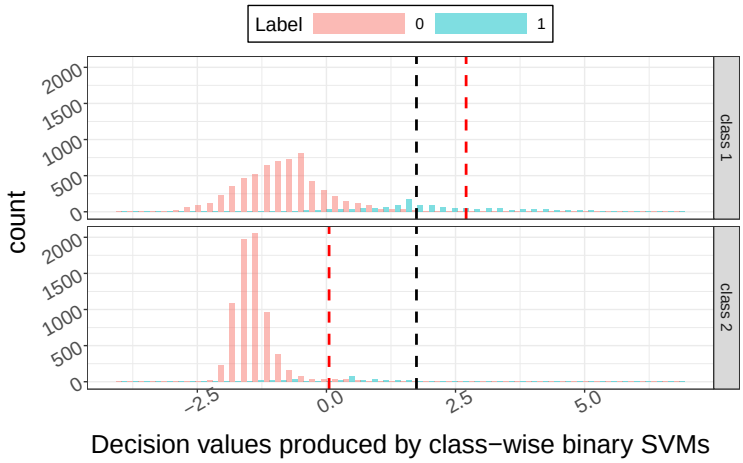


Figure 2.2: Distributions of the SVM decision values of two classes of the RCV1v2 dataset (see Section 2.7.4 for details). The red vertical dashed lines indicate the respective 95% quantiles of the two classes, and the black vertical dashed line indicates the 95% quantile of the mix of the two classes. Making decisions on the mix of the decision values of the two classes is likely to lead to a small power on the second class, e.g., the decision rule is to take the top 5% values as positive.

before the five true positives, whereas the corresponding PR curve was uninformative (i.e., both the recall and precision rates are zero for the first 100 called instances).

2.3.2 The local precision rate

The local precision rate was developed to maximize precision with respect to recall in the multi-label setting (Jiang et al., 2014). Specifically, it aims to maximize an expected population version of the micro-averaged precision, i.e., $\frac{\sum_k TP_k}{\sum_k (TP_k + FP_k)}$, and recall rate given by Pillai, Fumera, and Roli, 2013, where TP_k and FP_k are the number of true and false positives for class k, respectively.

The expected precision of the classifier for class k with threshold λ_k can be written as

$$G_k(\lambda_k) = \mathbb{P}(Y_k = 1 | S_k > \lambda_k) = \frac{\tau_k (1 - F_1^{(k)}(\lambda_k))}{1 - F^{(k)}(\lambda_k)},$$

where $\tau_k = \mathbb{P}(Y_k = 1)$.

We also have the joint probability $\mathbb{P}(S_k > s \text{ and } Y_k = 1)$ as $(1 - F^{(k)}(s)) G_k(F^{(k)}(s))$. By pooling decisions across all K classes with the thresholds $\lambda_1, \ldots, \lambda_k$, the expected pooled precision rate (ppr) can be written as

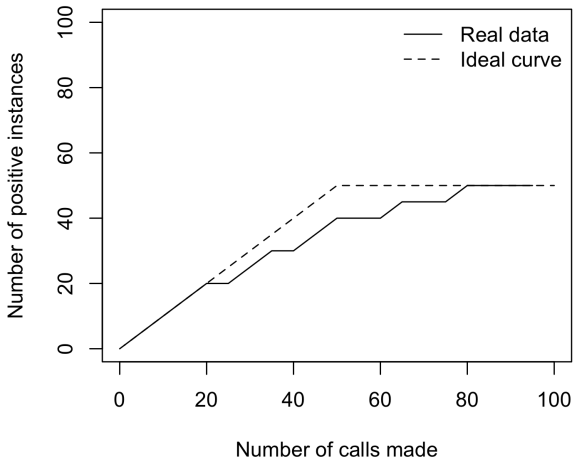


Figure 2.3: The hit curve.

$$ppr = \frac{\sum_k (1 - F^{(k)}(\lambda_k)) G_k(\lambda_k)}{\sum_k 1 - F^{(k)}(\lambda_k)},$$

where the denominator represents the *a priori* expected number of times a given instance will be assigned to a class with the decision thresholds $\lambda_1, \ldots, \lambda_k$. The pooled recall rate (prr) can be written in the same form, except with $\sum_k Y_k$ as the denominator instead.

In order to maximize ppr with respect to prr, it suffices to maximize $\sum_k (1 - F^{(k)}(\lambda_k)) G_k(\lambda_k)$ while fixing $\sum_k 1 - F^{(k)}(\lambda_k)$ considering $\sum_k Y_k$ is a constant. Then, the local precision rate (LPR) is defined as

$$LPR_k(s) = -\frac{d}{dF^{(k)}(s)}\{(1 - F^{(k)}(s)) G_k(s)\} = G_k(s) - (1 - F^{(k)}(s))\frac{dG_k(s)}{dF^{(k)}(s)}$$

In the main theoretical result Theorem 2.1 of Jiang et al. (2014), they showed that if the LPR for each class is monotonic, then ranking the LPRs calculated for each event maximizes the expected ppr with respect to a fixed recall rate. The monotonicity requirement is equivalent to having monotonicity in the likelihood of the positive class, which is satisfied when higher classifier scores correspond to a greater chance of being from the positive class–this rules out poorly behaved classifiers.

To better understand LPR, note that $\frac{dG_k(s)}{dF^{(k)}(s)} = \frac{dG_k(s)}{ds}\frac{ds}{dF^{(k)}(s)}$. It follows that

$$
\begin{aligned}
LPR_k(s) &= G_k(s) - (1 - F^{(k)}(s))\frac{dG_k(s)}{dF^{(k)}(s)} \\
&= G_k(s) - (1 - F^{(k)}(s))\left[-\frac{\tau_k f_1^{(k)}(s)}{(1 - F^{(k)}(s))f^{(k)}(s)} + \frac{\tau_k(1 - F_1^{(k)}(s))}{(1 - F^{(k)}(s))^2}\right] \\
&= G_k(s) + \frac{\tau_k f_1^{(k)}(s)}{f^{(k)}(s)} - G_k(s) \\
&= \frac{\tau_k f_1^{(k)}(s)}{f^{(k)}(s)} \\
&= \mathbb{P}(Y_k = 1 | S_k = s) = l\text{tdr},
\end{aligned}
$$

where $f^{(k)}$ and $f_1^{(k)}$ are the derivatives of $F^{(k)}$ and $F_1^{(k)}$, $l$tdr is short for local true discovery rate. The local false discovery rate, $l$fdr $= 1 - l$tdr is its more well known relative; it has been studied extensively for Bayesian large-scale inference (Efron, 2012).

To estimate LPR in practice, Jiang et al., 2014 discussed two methods. The first method employs kernel density estimates for $f_0^{(k)}$, $f^{(k)}$, and then plugs in τ_k's after expressing $LPR_k(s)$ as the $l$tdr. The second method employs a local quadratic kernel smoother to simultaneously estimate $G_k(s)$ and $G_k'(s)$. They showed that, under certain conditions, the first method has a faster convergence than the second. However, the second method performed better than the first in simulation studies. The difference results from the difficulty in estimating the densities $f_0^{(k)}$ and $f^{(k)}$: the estimates of $l$tdr are poor in any situation, which would make kernel density estimation difficult. Furthermore, $f_k^{(k)}$ and $f^{(k)}$ are estimated separately, so they have different levels of bias and variance. Particularly, since the estimation of $f_0^{(k)}$ only relies on the negative class cases, it has a larger variance. In comparison, the functions $G_k(u)$ and $G_k'(u)$ are estimated jointly in the second method.

2.4 Problem Formulation and an Theoretical Solution

In this section, we first introduce the motivation of using the hit curve to define a new objective function CATCH. Then we give the explicit form of CATCH, which leads to the key statistic mLPR. By showing the theoretical advantages of the mLPR, we conclude that sorting mLPRs in descending order not only gives a good classification performance in terms of CATCH but guarantees the hierarchical consistency as well. Finally, we make a brief discussion on the application of mLPRs to inference.

For simplicity, we vectorize $(\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(M)})$ and $(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M)})$ to get $\tilde{\mathbf{Y}} = (Y_1^{(1)}, \ldots, Y_K^{(1)}, \ldots, Y_1^{(M)}, \ldots, Y_K^{(M)})^{\mathrm{T}}$ and $\tilde{\mathbf{S}} = (S_1^{(1)}, \ldots, S_K^{(1)}, \ldots, S_1^{(M)}, \ldots, S_K^{(M)})^{\mathrm{T}}$, respectively. By repre-

senting the index i of $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{S}}$ as $i = (m-1) \cdot K + k$ where m denotes the object index and k denotes the class index, we write $\tilde{\mathbf{Y}} = (Y_1, \ldots, Y_n)^{\mathrm{T}}$, $\tilde{\mathbf{S}} = (S_1, \ldots, S_n)^{\mathrm{T}}$ when there is no confusion. We keep using this notation throughout the rest of the chapter.

2.4.1 Motivation

Given all the n scores $\tilde{\mathbf{S}}$ and the model $\mathcal{H}$, we attempt to produce a reasonable ranking of the associated n events. It leads to a natural question of what objective function to be employed to guide this ranking. Measures like ROC-AUC, precision, recall and F-measure are widely used in classification problems without hierarchy constraint. However, consensus on standard metrics has not yet been achieved for evaluation metrics of the HMC problem, and the development of such metrics remains an active research topic today. For example, H-loss (Cesa-Bianchi, Gentile, and Zaniboni, 2006b; Rousu et al., 2006; Cesa-Bianchi, Gentile, and Zaniboni, 2006a), matching loss (Nowak et al., 2010), hierarchical hamming loss, and hierarchical ranking loss (Bi and Kwok, 2015) are designed to blend the false positive rate (FPR) and the false negative rate (FNR) while considering the hierarchical structure. However, all of these loss functions depend on the choice of per-class cost and other coefficients, usually determined with intuition or domain knowledge rather than statistical justification. On the other hand, the hierarchical versions of precision, recall, and F-measure introduced in Kiritchenko, Matwin, and Famili (2005) and Verspoor et al. (2006) do not require determination of the per-class costs but have complicated forms that make them expensive to compute and difficult to optimize over.

A proper kind of metric to look at depends on the user's end goal, which explains why a standard evaluation metric has not yet been agreed upon (Costa et al., 2007). In this chapter, there are two goals to pursue. First, we are interested in a metric that emphasizes the accuracy in the initial set of positive calls rather than capturing all of the true positives in the dataset. This is motivated by the case of disease diagnosis, where physicians need to have confidence that the general category of disease has been correctly diagnosed, and the automatic diagnosis would tolerate or even expect mistakes at more specific levels since those typically need to be corroborated by expert knowledge (Huang, Liu, and Zhou, 2010). Second, we want a metric that fully incorporates the hierarchical information so that the maximization of this objective naturally respects the hierarchy. Thus we do not need to resort to a constraint.

To meet the first goal, we use the hit curve and the corresponding area under curve, where the x-axis represents the number of discoveries and the y-axis represents the number of true discoveries (i.e., the hit number); see Figure 2.3 in Section 2.3.1 for more details. The hit curve can serve well in the disease diagnosis situation as a graphic representation of the ranker's performance since it would plot the results in order of decreasing relevance, and the y-axis would indicate the results' relevance to the target. By contrast, the other traditional

metrics like ROC or PR curves are not sensitive or informative to the top-ranked instances, especially when the number of positive instances is tiny compared to the total size (Davis and Goadrich, 2006; Herskovic, Iyengar, and Bernstam, 2007; Hand, 2009). Additionally, there are three more reasons why the hit curve is preferred:

- There is a close connection between the hit curve and the PR curve, e.g., the slope of the hit curve is the precision rate. Most importantly, a large area under the hit curve corresponds to a large area under the PR curve.
- In the interest of optimization, it is easier to work with the number of total/true discoveries in the hit curve than the true/false discovery rates in the ROC/PR curves.
- The area under the hit curve does not depend on any manually designed hyperparameter.

As for the second goal, traditional metrics like matching loss and H-loss require hyperparameters that assign the class weights to leverage the hierarchy, which can barely guarantee the hierarchical consistency. DeCoro, Barutcuoglu, and Fiebrink (2007) and Bi and Kwok (2015) utilize a Bayesian idea to account for the dependencies, but it is not yet able to ensure the hierarchical consistency. So they leverage an additional constraint to this end. Here to incorporate the hierarchical information into the objective function, we consider the expected area under the hit curve conditional on the scores across all the classes. It turns out that maximizing this objective function without any constraint can, in theory, produce a hierarchical consistent ranking; see Section 2.4.3.

2.4.2 Conditional expected Area under The Curve of the Hit curve and Multidimensional Local Precision Rate

From the motivation in Section 2.4.1, we aim to find out a ranking such that the area under the hit curve is maximized. Formally, we want to solve the optimization problem (2.4.1). Here the hierarchy constraint is added to make the optimization a complete HMC problem. Subsequently, after considering a conditional version of the area under the hit curve, we will show that this constraint is no longer needed.

$$\begin{aligned} \max_{\boldsymbol{\pi}} \quad & \text{area under the hit curve,} \\ s.t. \quad & \pi_i < \pi_{i'} \text{ if Event } i \text{ is an ancestor of Event } i'. \end{aligned} \tag{2.4.1}$$

Note that the x-axis of the hit curve represents the number of calls made, and thus the expression of the area under the hit curve is equivalent to the sum of the number of true

positives among the top i calls, for every i. This yields the convenient expression for (2.4.1):

$$(2.4.1) = \sum_{i=1}^{n}\sum_{j=1}^{i}\mathbb{I}(Y_{\pi_j}=1) = \sum_{i=1}^{n}(n-i+1)\,\mathbb{I}(Y_{\pi_i}=1). \tag{2.4.2}$$

Since Y_{π_i} is a random variable when the inputs are random objects, we consider the population average of the area under the hit curve. Specifically, we take the conditional expected values of (2.4.2) given the classifier scores $\tilde{\mathbf{S}}$, on account of the dependencies, and arrive at

$$\text{CATCH} := \mathbb{E}(\text{AUC of hit curve}|\,\tilde{\mathbf{S}}) = \sum_{i=1}^{n}(n-i+1)\,\mathbb{P}(Y_{\pi_i}=1|S_1,\ldots,S_n). \tag{2.4.3}$$

Here, we call the target metric the **Conditional expected Area under The Curve of the Hit curve (CATCH)**. We call $\mathbb{P}(Y_i=1|S_1,\ldots,S_n)$ **multidimensional local precision rate (mLPR)**. Just as multidimensional local false discovery rate (mℓfdr) extends the traditional ℓfdr (Ploner et al., 2006), mLPR is a multidimensional extension to Local Precision Rate (LPR). More details on LPR can be found in Section 2.4.3 and Section 2.3.2. Proposition 1 below shows a desired property of mLPR.

Proposition 1 *Under model $\mathcal{H}$ defined in Section 2.2.2, for two events i and i', if $i \in anc(i')$, then $mLPR_i \geq mLPR_{i'}$.*

Proof Under model $\mathcal{H}$ defined in Section 2.2.2, $i \in anc(i')$ means that the two events i and i' concern the same object, and that the associated class node of Event i is an ancestor of that of Event i'.

For any i' and $i \in anc(i')$, it follows that

$$\begin{aligned}
mLPR_{i'} &= \mathbb{P}(Y_{i'}=1|S_1,\ldots,S_n)\\
&= \sum_{Y_1,\ldots Y_{i'-1},Y_{i'+1},\ldots,Y_n} \mathbb{P}(Y_1,\ldots,Y_{i'-1},Y_{i'}=1,Y_{i'+1},\ldots,Y_n|S_1,\ldots,S_n)\\
&\overset{(a)}{=} \sum_{Y_j:j\neq i,j\neq i'} \mathbb{P}(Y_1,\ldots,Y_{i'}=1,Y_i=1,\ldots,Y_n|S_1,\ldots,S_n)\\
&\leq \sum_{Y_j:j\neq i} \mathbb{P}(Y_1,\ldots,Y_{i-1},Y_i=1,Y_{i+1},\ldots,Y_n|S_1,\ldots,S_n)\\
&= mLPR_i,
\end{aligned}$$

where Equation (a) is obtained by the condition (ii) of the model $\mathcal{H}$ (Section 2.2.2). ■

Proposition 1 tells that the mLPR value of an event can not be smaller than those of its descendants. Besides, it can be shown that a larger mLPR indicates the associated event is more likely to be positive; see Section 2.4.3. Based on the two properties, we propose a population-level solution to maximizing CATCH under the hierarchical constraint.

2.4.3 Sorting mLPRs in Descending Order

We aim to find the ranking that maximizes CATCH (2.4.3) while respecting the hierarchy. This can be mathematically written as the optimization problem.

$$\begin{aligned} \max_{\boldsymbol{\pi}} \quad & CATCH, \\ s.t. \quad & \pi_i \leq \pi_{i'} \text{ if Event } i \text{ is an ancestor of Event } i'. \end{aligned} \tag{2.4.4}$$

We can generate the ranking by naively sorting any scores (here, we use mLPRs) from the largest to the smallest. We call this method **naive sorting**. Proposition 1 indicates that the ranking by applying naive sorting on mLPRs satisfies the hierarchical consistency. It immediately implies that this ranking is the solution to the problem (2.4.4), as shown in Proposition 2. In other words, if we can get access to the population mLPRs, solving the optimization problem (2.4.4) does not require the hierarchical constraint. This conclusion is reasonable because the hierarchy information has been fully incorporated into mLPRs.

Proposition 2 *Under Model $\mathcal{H}$ defined in Section 2.2.2, the ranking obtained by naive sorting on mLPRs is a topological ordering for $\mathcal{G}$ and maximizes* (2.4.3)*.*

Proof Proposition 1 indicates that sorting mLPRs from the largest to the smallest can guarantee the hierarchy constraint that ancestors rank ahead of their descendants. Meanwhile, the maximum of CATCH (2.4.3) is just obtained by sorting mLPRs in this manner. ■

Furthermore, we will show in Proposition 3 that an event at the top of the ranking obtained by applying naive sorting on mLPRs is more likely to be positive than an event in the tail. In other words, given a decision rule induced by imposing a cutoff on such a ranking, the event taken as positive by this rule is more likely to be truly positive than those taken as negative. Fundamentally, it reflects that mLPRs account for the statistical differences across classes. Thus, with Proposition 1 and Proposition 3 together, it is statistically justified to directly compare mLPRs in the scenario where there are dependencies between classes. It is an extension of a similar result for LPRs shown in Jiang et al. (2014) that sorting LPRs in the decreasing order guarantees the optimal pooled precision at any pooled recall rate if there is no hierarchy constraint among the classes. More details on LPR and its relation to the local true discovery rate (Efron, 2012) are deferred to Section 2.3.2.

Proposition 3 *Given Model $\mathcal{H}$ defined in Section 2.2.2, let $\boldsymbol{\pi}^{ns}$ be the ranking obtained by sorting mLPRs in an descending order. Then for any positive event i and i' with $\pi_i^{ns} < \pi_{i'}^{ns}$, we have*

$$\mathbb{P}(Y_i = 1 | \pi_i^{ns} < \pi_{i'}^{ns}) \geq \mathbb{P}(Y_{i'} = 1 | \pi_i^{ns} < \pi_{i'}^{ns}).$$

Proof For a realization $\tilde{\mathbf{s}}$ of $\tilde{\mathbf{S}}$, the resulting mLPRs give the ranking $\boldsymbol{\pi}^{ns}$ with $\pi_i^{ns} < \pi_{i'}^{ns}$, indicating that $mLPR_i = \mathbb{P}(Y_i = 1 | \tilde{\mathbf{S}} = \tilde{\mathbf{s}}) \geq \mathbb{P}(Y_{i'=1} | \tilde{\mathbf{S}} = \tilde{\mathbf{s}}) = mLPR_{i'}$. Then we have

$$\begin{aligned}
\mathbb{P}(Y_i = 1, \pi_i^{ns} < \pi_{i'}^{ns}) &= \int_{\tilde{\mathbf{s}}: \pi_i^{ns} < \pi_{i'}^{ns}} \mathbb{P}(Y_i = 1, \tilde{\mathbf{S}} = \tilde{\mathbf{s}}) \\
&\geq \int_{\tilde{\mathbf{s}}: \pi_i^{ns} < \pi_{i'}^{ns}} \mathbb{P}(Y_{i'} = 1, \tilde{\mathbf{S}} = \tilde{\mathbf{s}}) \\
&= \mathbb{P}(Y_{i'} = 1, \pi_i^{ns} < \pi_{i'}^{ns}).
\end{aligned}$$

■

2.5 HierRank: Ranking Algorithm based on estimated mLPRs

In this section, we describe a routine on how to compute mLPRs given the observed classifier scores. First, two approximations are provided in light of the strengths of the class dependencies. Then, we develop a ranking method to rank the estimated mLPRs, for which the simple naive sorting can longer guarantee the maximized empirical CATCH (given the estimated mLPRs) and the hierarchical consistency.

2.5.1 Computation of mLPRs

The sound properties of mLPRs and the naive sorting method can be guaranteed when we know the true values of mLPRs. In reality, it is hard to get this ideal solution, and we have to estimate mLPRs. Given the model $\mathcal{H}$ defined in Section 2.2, we are able to investigate

$\mathbb{P}(Y_1, \ldots, Y_n | S_1, \ldots, S_n)$ in a simple manner:

$$\begin{aligned}
\mathbb{P}(Y_1, \ldots, Y_n | S_1, \ldots, S_n) &\overset{(a)}{\propto} \mathbb{P}(S_1, \ldots, S_n | Y_1, \ldots, Y_n) \cdot \mathbb{P}(Y_1, \ldots, Y_n) \\
&\overset{(b)}{=} \prod_{i=1}^{n} \mathbb{P}(S_i | Y_i)\, \mathbb{P}(Y_i | Y_{pa(i)}) \\
&\overset{(c)}{\propto} \prod_{i=1}^{n} \frac{\mathbb{P}(Y_i | S_i)}{\mathbb{P}(Y_i)} \cdot \mathbb{P}(Y_i | Y_{pa(i)}) \\
&\overset{(d)}{=} \prod_{i=1}^{n} LPR_i \cdot \frac{\mathbb{P}(Y_i | Y_{pa(i)})}{\mathbb{P}(Y_i)}, \qquad (2.5.1)
\end{aligned}$$

where (a) and (c) hold by the Bayes rule, (b) holds by using the Markov property with Assumption 1 (conditional independence), and Equation (d) follows from Jiang et al. (2014) that given the scores $\tilde{\mathbf{S}}$, the associated LPR of the i-th node is defined as

$$LPR_i = \mathbb{P}(Y_i | S_i).$$

We estimate LPRs by applying the method in Jiang et al. (2014). We estimate $\mathbb{P}(Y_i | Y_{pa(i)})$ and $\mathbb{P}(Y_i)$ with the empirical proportions (e.g., counting number of objects such that $Y_i = 1$ to estimate $\mathbb{P}(Y_i)$). Then, with these estimates and by applying the sum-product message passing to $\mathbb{P}(Y_1, \ldots, Y_n | S_1, \ldots, S_n)$ with respect to $\mathcal{G}$ (Wainwright and Jordan, 2008), we obtain an estimator $\widehat{mLPR}$ of mLPR. We call it the **full** version of $\widehat{mLPR}$ if we keep strictly to the above estimation procedure of mLPR. When the dependency structure is sparse, some estimation approaches that ignore or simplify the dependency structure may generate reasonable approximations with improvement in computation cost. We consider two strategies to approximate the mLPR in terms of the strength of the dependencies between classes. To be specific,

- **Independence**. We assume that Y_i is independent of $S_{i'}$ for $i' \neq i$. Then we get $mLPR_i \approx \mathbb{P}(Y_i | S_i)$, which is simply LPR_i. This type of computation is called the independence (short as **indpt**) approximation.

- **Neighborhood**. We assume that Y_i is only correlated with $S_{i'}$ for $i' = i$ or $i' \in nbh(i)$. Then we get $mLPR_i \approx \mathbb{P}(Y_i | S_i, S_{nbh(i)})$, which can be computed in the same fashion as Equation (2.5.1). This type of computation is called the neighborhood (short as **nbh**) approximation.

The independence approximation has been widely used in statistical methods such as Naive Bayes and Variational Bayes (Ng and Jordan, 2001; Wainwright and Jordan, 2008).

The neighborhood approximation is a compromise between the independence approximation and the full mLPR computation based on the complete dependencies. The assumption of the neighborhood dependencies can be regarded as a mild augmentation of Assumption 1. In practice, the choice of the three versions of mLPR computations depends on the signal-to-noise ratio. It is reasonable to use the independence approximation or the neighborhood approximation when a weak or local dependence is observed between classes. On the other hand, even though there is an explicit dependency between classes, we will also turn to the independence approximation when the signal is weak due to data insufficiency or bad data quality; see Section 2.7.3 for an example.

Finally, given $\widehat{mLPR}$s, we consider

$$\widehat{CATCH} := \sum_{i=1}^{n} (n - i + 1)\widehat{mLPR}_i, \tag{2.5.2}$$

which is an empirical version of CATCH (2.4.3).

2.5.2 Algorithms

Proposition 2 holds when true mLPRs are available. Sorting $\widehat{mLPR}$s naively (in descending order), however, might violate the hierarchical constraint. Consequently, here we introduce a sorting algorithm, named **HierRank**, to provide a ranking $\boldsymbol{\pi}$ that gives the largest possible $\widehat{CATCH}$ in (2.5.2) among all the rankings that satisfy the hierarchical constraint. We consider this an empirical solution to the constraint optimization problem (2.4.4). Formally, **HierRank** aims to solve the following problem:

$$\begin{aligned} \max_{\boldsymbol{\pi}} \quad & \widehat{CATCH}, \\ s.t. \quad & \pi_i \leq \pi_{i'} \text{ if Object } i \text{ is an ancestor of Object } i'. \end{aligned} \tag{2.5.3}$$

We first define the terminologies that will be used in the algorithm. The terms are also illustrated in Figure 2.4.

- **node**: a node corresponds to an event (the status of an object in a class) in the HMC problem.
- **tree**: a tree is an undirected graph in which any two nodes are connected by exactly one path.
- **Single-child branch/chain**: a branch/chain of the tree, on which every node has at most one child. In Figure 2.4, the chains $B \to D \to E$, $G \to H$, $I \to J$ and $K \to L$ are single-child branches.

- $\boldsymbol{\mathcal{P}_1}$: the set of nodes on all single-child branches, i.e., a node is in $\mathcal{P}_1$ if it and its descendants have at most one child. In Figure 2.4, nodes on the chains $B \to D \to E$, $G \to H$, $I \to J$, $K \to L$ belong to $\mathcal{P}_1$. Node A does not belong to $\mathcal{P}_1$ because it is not on a single-child branch.

- **Starting node** in $\mathcal{P}_1$: a node that is in $\mathcal{P}_1$ but its parent(s) are not. In Figure 2.4, Node B, G, I and K are starting nodes in $\mathcal{P}_1$.

- $\boldsymbol{\mathcal{P}_2}$: a set of nodes with at least two children and those children are in $\mathcal{P}_1$. Any node in $\mathcal{P}_2$ is attached by multiple single-child branches starting from its child nodes. In Figure 2.4, only Node F belongs to $\mathcal{P}_2$. It is attached by the chains $G \to H$ and $I \to J$. Node A does not belong to $\mathcal{P}_2$ because its child C does not belong to $\mathcal{P}_1$.

- $\boldsymbol{\mathcal{P}_3}$: a set of nodes who are the parents/ancestors of the nodes in $\mathcal{P}_2$ and they have only one child. In Figure 2.4, only Node C belongs to $\mathcal{P}_3$. Node A does not belong to $\mathcal{P}_3$ since it has two child nodes.

- $\boldsymbol{\mathcal{C}_{r\to s}}$: a sub-chain that starts from Node r and ends at Node s (a sub-chain/path is unique for a tree, given the two ends). Let $|\,\mathcal{C}_{r\to s}\,|$ be the number of nodes in $\mathcal{C}_{r\to s}$. For example, in Figure 2.4, $\mathcal{C}_{A\to E}$ represents the chain $A \to B \to D \to E$.

- $\boldsymbol{\mathcal{C}_r}$: a simplified notation for $\mathcal{C}_{r\to e}$, if $r \in \mathcal{P}_1$ and Node e is a leaf node. For example, in Figure 2.4, $\mathcal{C}_{\mathrm{B}}$ represents the chain B $\to$ D $\to$ E.

- $\boldsymbol{\mathcal{C}_r(h)}$: a sub-chain that consists of the first h nodes of $\mathcal{C}_r$. For example, in Figure 2.4, $\mathcal{C}_{\mathrm{B}}(2)$ is a sub-chain of $\mathcal{C}_{\mathrm{B}}$ consisting of $\mathcal{C}_{\mathrm{B}}$'s first two nodes, i.e., B $\to$ D.

- $\boldsymbol{\bar{\ell}_{r,h}(\mathbf{S})}$: the average value of scores (e.g., $\widehat{mLPR}$s) of the sub-chain $\mathcal{C}_r(h)$, i.e., $\bar{\ell}_{r,h}(\mathbf{S}) = \frac{1}{|\,\mathcal{C}_r(h)\,|}\sum_{i\in\mathcal{C}_r(h)} S_i$. Here we use $C_r(h)$ to denote the set of nodes in the sub-chain $\mathcal{C}_r(h)$ when there is no ambiguity.

We first consider ranking nodes from multiple disjoint single-child chains $\mathcal{C}_{r_1}, \ldots, \mathcal{C}_{r_p}$. This is equivalent to merging these chains into a single chain. The relative position along the final chain will reflect the ranking. To this end, we introduce Algorithm 1 that is illustrated in Figure 2.5:

(a) Initialize the ranked list $\mathcal{L} = \emptyset$.

(b) For the chains $I \to J$ and $G \to H$, there are four sub-chains: G, $G \to H$, I, $I \to J$ with average scores 0.8, 0.45, 0.3, 0.6 respectively. The sub-chain G has the largest average, so we remove it from the original graph and attach it to $\mathcal{L}$.

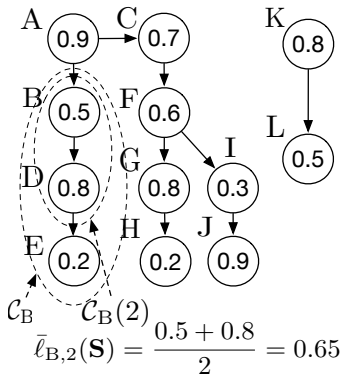


Figure 2.4: Illustration of notation. The numbers inside the nodes are the associated scores.

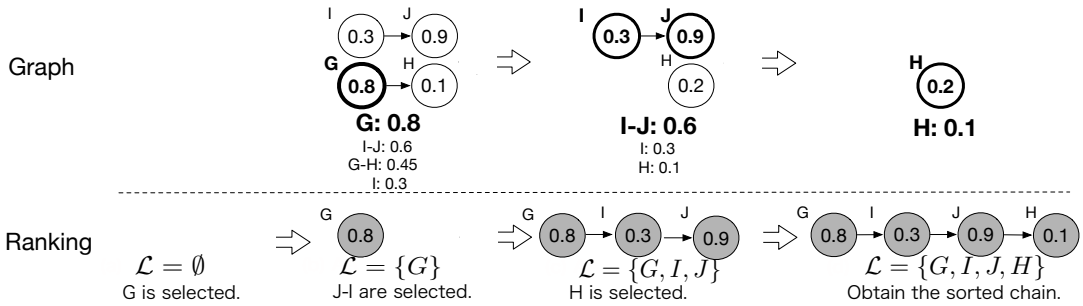


Figure 2.5: An example of the merging process in Algorithm 1: merge the two sub-chains $G \to H$ and $I \to J$ in Figure 2.4. The nodes in bold form a sub-chain of the highest averaging scores, and the nodes filled in grey give a ranking produced by the merging procedure.

(c) In the remaining graph, there are three sub-chains: $I \to J$, I and H with average scores 0.6, 0.3, 0.1. The sub-chain $I \to J$ has the largest average, so we remove it from the remaining graph and attach it to $\mathcal{L}$.

(d) There remains a single node H. We attach it to $\mathcal{L}$. Since there is no node in the remaining graph, $\mathcal{L}$ is the final ranking.

The produced ranking of Algorithm 1 satisfies the hierarchical consistency because it preserves the relative ordering of the nodes in each chain. This ranking also maximizes (2.5.2). The heuristic is that this algorithm essentially sorts the average scores in descending order. The detailed argument is part of the proof of Theorem 4.

For a general tree case, we introduce Algorithm 2 (HierRank), which uses Algorithm 1 repeatedly. Figure 2.6 is used to illustrate this algorithm:

(a) Identify $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. In Figure 2.6 (a), $\mathcal{P}_1 = \{B, D, E, G, H, I, J, K, L\}$. $\mathcal{P}_2 = \{F\}$, $\mathcal{P}_3 = \{C\}$.

(b) For each node in $\mathcal{P}_2$, we apply Algorithm 1 to merge the chains attached to it. Then attach the resulting single chain to this node, and update $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$. In Figure 2.6 (a), $\mathcal{P}_2$ only contains Node F. Apply Algorithm 1 to merge the two sub-chains $G \to H$ and $I \to J$ attached to node F. Attach the resulting chain $G \to I \to J \to H$ to node F, and we get Figure 2.6 (b). Update $\mathcal{P}_1 = \{B, D, E, C, F, G, I, J, H, K, L\}$, $\mathcal{P}_2 = \{A\}$, $\mathcal{P}_3 = \emptyset$.

(c) Repeat step (b) until $\mathcal{P}_2$ is empty (then $\mathcal{P}_3$ is empty as well). In Figure 2.6 (b), $\mathcal{P}_2$ only contains node A. Apply Algorithm 1 to merge the two sub-chains $B \to D \to E$ and $C \to F \to G \to I \to J \to H$ that are attached to node A. Attach the resulting chain $C \to F \to G \to B \to D \to I \to J \to E \to H$ to node A, and we obtain Figure 2.6 (c). Update $\mathcal{P}_1 = \{\text{all nodes}\}$, $\mathcal{P}_2 = \emptyset$, $\mathcal{P}_3 = \emptyset$. Since $\mathcal{P}_2$ is empty now, we terminate the loop.

(d) Apply Algorithm 1 to merge the remaining single-child chains. In Figure 2.6 (c), there remain two sub-chains $K \to L$ and $A \to C \to F \to G \to B \to D \to I \to J \to E \to H$. Apply Algorithm 1 to merge them, and we obtain the final ranking $A \to K \to C \to F \to G \to B \to D \to I \to J \to L \to E \to H$.

In the very beginning, some nodes are put in $\mathcal{P}_1$, $\mathcal{P}_2$ or $\mathcal{P}_3$, and the other nodes are left out. As the algorithm proceeds, the nodes in $\mathcal{P}_2$ and $\mathcal{P}_3$ are transferred to $\mathcal{P}_1$, and some left-out nodes are transferred to $\mathcal{P}_2$ and $\mathcal{P}_3$ until all the nodes are put in $\mathcal{P}_1$ and there remains a single chain. It can be seen that $\mathcal{P}_2$ are updated upwards along the graph, so HierRank works in a bottle-up fashion. As we note above, when we repeat using Algorithm 1 in HierRank, the local ranking satisfies the hierarchy constraint and attains the maximum of Equation (2.5.2), conditional on the involved sub-graph. We show in Theorem 4 that HierRank enjoys the desired optimality that it produces a topological ordering of $\mathcal{G}$ with the maximum of Equation (2.5.2). The detailed proof is deferred to Appendix A.1.

Theorem 4 *Let $\mathcal{G}'$ be a new graph obtained by replacing any sub-tree in $\mathcal{G}$ with the corresponding merged chain given by Algorithm 2. Then an optimal topological ordering of $\mathcal{G}'$ is also an optimal topological ordering of $\mathcal{G}$. Hence, Algorithm 2 leads to an optimal topological ordering by merging all the trees into a single chain.*

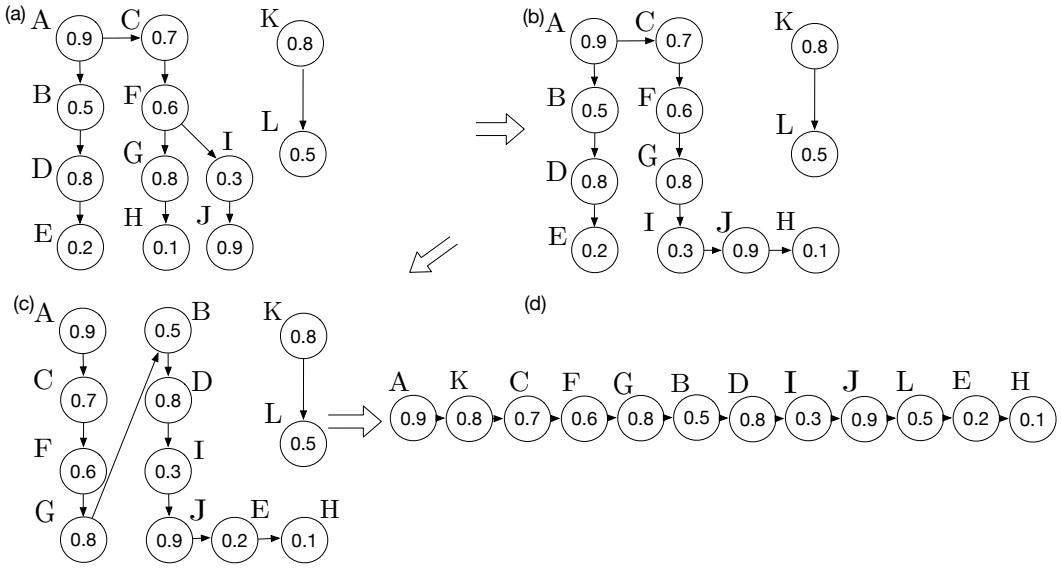


Figure 2.6: An example of the merging process in Algorithm 2: (a)→(b) merge $G \to H$ and $I \to J$ into $G \to I \to J \to H$; (b)→(c) merge $B \to D \to E$ and $C \to F \to G \to I \to J \to H$ to $C \to F \to G \to B \to D \to I \to J \to E \to H$; (c)→(d) merge all nodes to $A \to K \to C \to F \to G \to B \to D \to I \to J \to L \to E \to H$.

Algorithm 1 The Chain-Merge algorithm.

Input: p chains $\mathcal{D} = \{\text{node} \in \mathcal{C}_r : r = r_1, \ldots, r_p\}$, the node scores $\tilde{\mathbf{S}}$ (e.g., classifier scores, or $\widehat{mLPRs}$).

Procedure:

1: Set $\mathcal{L} = \emptyset$;
2: Compute $\{\bar{\ell}_{r,i}(\tilde{\mathbf{S}}) : i = 1, \ldots \,|\, \mathcal{C}_r \,|, r = r_1, \ldots, r_p\}$.
3: **while** $\mathcal{D} \neq \emptyset$ **do**
4: $\quad (r', h') = \arg \max_{C_r(h) \subset \mathcal{D}} \bar{\ell}_{r,h}(\tilde{\mathbf{S}})$.
5: $\quad \mathcal{L} \leftarrow \mathcal{L} \oplus C_{r'}(h')$, where $\oplus$ indicates the concatenation of two sequences.
6: $\quad \mathcal{D} \leftarrow (\mathcal{D} \setminus C_{r'}) \cup (C_{r'} \setminus C_{r'}(h'))$.
7: $\quad$ Update the average scores of the remaining nodes as step 2.
8: **end while**

Output: $\mathcal{L}$.

Algorithm 2 The HierRank algorithm for the tree hierarchy.

Input: The tree graph $\mathcal{G}$, node scores $\tilde{\mathbf{S}}$ (e.g., classifier scores, or $\widehat{mLPR}s$).
Procedure:

1: Figure out $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$.
2: **while** $\mathcal{P}_2 \neq \emptyset$ **do**
3: Pop out one v from $\mathcal{P}_2$. Take all children of v, i.e., r_1, r_2, ...
4: Feed C_{r_1}, C_{r_2}, ... into Algorithm 1 and obtain $\mathcal{L}(r_1, r_2, \ldots)$.
5: Replace C_{r_1}, C_{r_2}, ... with $\mathcal{L}(r_1, r_2, \ldots)$.
6: Update $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$.
7: **end while**
8: **if** There remain multiple chains **then**
9: Apply Algorithm 1 to these chains.
10: **end if**
11: Let $\mathcal{L}$ be the resulting chain.
[1] **Output:** a ranking $\mathcal{L}$.

We have three remarks for HierRank. First, the property stated in Proposition 3 is not guaranteed when using $\widehat{mLPR}$s instead of mLPRs. If we have good estimates of $\mathbb{P}(Y_i)$, $\mathbb{P}(Y_i|Y_{pa(i)})$ and LPR_i, the naive sorting behaves similarly to HierRank when ranking $\widehat{mLPR}s$; see Section 2.7.2 for an example. If the estimates are way off the true values, $\widehat{mLPR}s$ will miss the hierarchy information in $\mathcal{G}$, and thus HierRank outperforms naive sorting significantly; see Section 2.7.3 for an example. Second, the time complexity of HierRank reaches up to $\mathcal{O}(K^3)$ for each individual. It implies that the ranking over the K nodes in $\mathcal{G}$ across M individuals costs $\mathcal{O}(nK^2 + n \log M)$ computations ($n = MK$). This high complexity stems from the exhaustive merging and repeated computations of the moving average at each iteration. To improve the efficiency, we modify Algorithm 2 by segmenting a chain into blocks, which are defined by the maximal running average. In this fashion, we eliminate the redundant computations and obtain Algorithm 2′. Algorithm 2′ is equivalent to Algorithm 2 in light of the output and only costs $\mathcal{O}(n \log n)$ operations. In addition, we note that there is an existing algorithm Condensing Sort and Select Algorithm (CSSA) (Bi and Kwok, 2011) that has the same time complexity as Algorithm 2′. CSSA was designed to make the first L decisions (L is a positive integer) and can be applied to the ranking problem. It produces almost the same ranking as HierRank except for some local parts; see Section 2.6.2 for the details of Algorithm 2′ and CSSA. Third, we can extend Algorithm 2 that is designed for the tree graph to handle the DAG graph. The details of this algorithm are deferred to Section 2.6.3.

2.6 Discussion on HierRank

2.6.1 An equivalent algorithm

We find an existing algorithm called Condensing Sort and Select Algorithm (CSSA) (Baraniuk and Jones, 1994) that is also of $\mathcal{O}(n \log n)$ complexity and can be adapted to solve (2.5.3). Bi and Kwok, 2011 first extended CSSA in their proposed decision rule for the HMC problem. In their paper, CSSA was used to provide an approximate solution to the integer programming problem

$$\max_{\Psi} \quad \sum_{k \in \mathcal{T}} B(k)\Psi(k) \tag{2.6.1}$$

$$s.t. \quad \Psi(k) \in \{0,1\}, \forall k, \quad \sum_{k \in \mathcal{T}} \Psi(k) = L, \tag{2.6.2}$$

$$\Psi \text{ is } \mathcal{T}\text{-non-increasing},$$

where $\mathcal{T}$-nonincreasing means that $\Phi(k) \leq \Phi(k')$ if node k' is the ancestor of node k; $B(k)$ is a score produced by kernel dependency estimation (KDE) approach (Weston et al., 2003). Instead of directly solving (2.6.1) with (2.6.2), Bi and Kwok, 2011 tackles a relaxed problem by replacing the binary constraint (2.6.2) with

$$\Psi(k) \geq 0, \forall k, \quad \Psi(0) = 1, \sum_{k \in \mathcal{T}} \Psi(k) \leq L. \tag{2.6.3}$$

Bi and Kwok, 2011 proposed CSSA (Algorithm 3) to solve this problem. They showed that this algorithm can produce the optimal result that maximizes the objective function (2.6.1) while respecting (2.6.3).

Algorithm 3 The CSSA algorithm

Input: A collection of trees $\mathcal{T}$, scores $\mathbf{S}$ (e.g., $\widehat{mLPR}_i$'s)
Denote $Par(T_k)$ as the parent of supernode T_k, $n(T_k)$ as the number of nodes in T_k, and Ψ as a vector indicating which nodes are selected.

Initialize $\Psi(0) \leftarrow 1$; $\Gamma \leftarrow 1$.
Initialize all other nodes as supernodes with $\Psi(k) \leftarrow 0$ and sort them according to the scores.
while $\Gamma < L$ **do**
  Find $k = \arg\max_{k'} \frac{1}{n(T_{k'})} \sum_{i \in T_{k'}} S_i$
  if $\Psi(Par(T_k)) = 1$ **then**
    $\Psi(T_k) \leftarrow \min\{1, (L - \Gamma)/n(T_k)\}$
    $\Gamma \leftarrow \Gamma + n(T_k)$
  else
    Condense T_k and $Par(T_k)$ as a new supernode.
  end if
end while

Output: Vector $\mathbf{\Psi} = (\Psi(1), \Psi(2), \ldots)$.

Note that CSSA has a property that $\Psi(k) = 1$ for L implies $\Phi(k) = 1$ for L' (the same node k) when $L < L'$. It indicates this algorithm is able to produce a ranking by varying L. It turns out that CSSA can be modified as Algorithm 4 that is shown to generate the same result as HierRank (see Theorem 5). On the other hand, we note CSSA and Algorithm 2 differ in the following aspects.

First, HierRank is independently introduced and interpreted in the context of CATCH, with a statistical justification for ordering nodes using mLPRs in particular. CSSA originates in signal processing and has been successfully used in wavelet approximation and model-based compressed sensing (Baraniuk and Jones, 1994; Baraniuk, 1999; Baraniuk et al., 2010).

Second, there might be local differences between the ranking of HierRank and that of CSSA. This results from the relaxation condition (2.6.3) — the same set of nodes can be selected for L and $L + 1$, thus CSSA cannot differentiate the ordering of some nodes. For example, consider a simple tree $B \leftarrow A \rightarrow C$, with $S_B = 3.6$, $S_A = 3$, $S_C = 4$. In this case, $\Psi(A) = \Psi(B) = \Psi(C) = 1/3$ when $L = 1$; $\Psi(A) = \Psi(B) = \Psi(C) = 2/3$ when $L = 2$; $\Psi(A) = \Psi(B) = \Psi(C) = 1$ when $L = 3$. So Nodes A, B and C are always picked together. CSSA only knows that A should be ranked ahead of B and C, but cannot determine which of B and C should rank first. On the other hand, HierRank gives the resulting ranking $A \rightarrow C \rightarrow B$.

Finally, HierRank merges the chains from the bottom up, rather than as in CSSA, constructing ordered sets of nodes called supernodes by starting from the node with the largest value in the graph and moving outward. It is easy to see that the blocks defined in Algorithm 2′ (the faster version of HierRank introduced in Section 2.6.2) are essentially the same as the supernodes taken off in Algorithm 4. Hence, our independently proposed algorithm provides some novel insight into CSSA under the HMC setting.

Theorem 5 *Algorithm 2 and Algorithm 4 yield the same ordering, so Algorithm 4 maximizes CATCH as well.*

Algorithm 4 An equivalent algorithm modified from CSSA.

Input: A forest $\mathcal{T}$, scores $\mathbf{S}$ (e.g., $\widehat{mLPR}_i$'s)
Denote $Par(T_k)$ as the parent of supernode T_k, $n(T_k)$ as the number of nodes in T_k, and $\mathcal{L}$ as a vector for holding sorted scores .
Procedure:

Initialize with one node per score value, and each node as its own supernode, $\mathcal{L} = []$ (empty vector).
while $|\,\mathcal{L}\,| < n$ **do**
  Find $k = \arg\max_{k'} \ \frac{1}{n(T_{k'})} \sum_{i \in T_{k'}} S_i$
  if $Par(T_k) = \emptyset$ **then**
    Take the nodes in T_k off the graph and append them to $\mathcal{L}$.
  else
    Condense T_k and $Par(T_k)$ into a supernode.
  end if
end while

Output: A ranking $\mathcal{L}$.

2.6.2 A faster implementation of HierRank

To solve the scalability issue of Algorithm 2, we propose a faster version of HierRank by reducing redundant and repetitive computations in Algorithm 2. The speed-up is motivated by the following observations: 1) Algorithm 1 breaks a single chain into multiple blocks via the formula $(r', h') = \arg \max_{C_r(h) \subset \mathcal{D}} \bar{\ell}_{r,h}(\tilde{\mathbf{S}})$; 2) It can be shown that these blocks can only be agglomerated into a larger block rather than being further partitioned into smaller ones; 3) the agglomeration occurs only between a parent block and its child blocks in the hierarchy. Thus, HierRank can be implemented at the block level so that the partition is only executed once

during multiple merging. By considering the above facts and taking care of other details, we obtain a faster version of HierRank (see Algorithm 2′), which costs $\mathcal{O}(n \log n)$ computations.

In Algorithm 1, we note the fact that each sub-chain in the tree can be partitioned into multiple blocks — given a chain C_r, the breaking points are sequentially defined as

$$p_j := \begin{cases} \max_{1 \le h \le |C_r|} \frac{1}{C_r(h)} \sum_{k \in C_r(h)} S_k, & \text{if } j == 1 \\ \max_{p_{j-1} < h \le |C_r|} \frac{\sum_{k \in C_r(h)/C_r(p_{j-1})} S_k}{|C_r(h)| - |C_r(p_{j-1})|}, & \text{if } j > 1 \end{cases} \tag{2.6.4}$$

For example, Figure 2.7 (i) shows a chain of 6 nodes can be partitioned into two blocks. During the merging procedure of Algorithm 2, it turns out that the blocks defined by the above partitions will not be broken into smaller pieces, but can be further agglomerated. To show this, suppose there are two consecutive blocks in a chain, B_1, B_2, and B_1 locates ahead of B_2. Now we reform the blocks from the nodes in B_1 and B_2, using the rule in (2.6.4). It is obvious that nodes in B_1 will be clustered together. It remains to see which nodes in B_2 will be clustered with the nodes in B_1. Denote by $B_2(h)$ a sub-block consisting of the first h nodes in B_2, and by $\ell_B = \frac{1}{|B|} \sum_{k \in B} S_k$ given a block B. Then, the average scores of the nodes in B_1 and the first h nodes in B_2 is computed as:

$$\ell_{B_1 \cup B_2(h)} \quad = \quad \frac{|B_1| \ell_{B_1} + h \ell_{B_2(h)}}{|B_1| + h} = \ell_{B_1} + \frac{\ell_{B_2(h)} - \ell_{B_1}}{|B_1|/h + 1}. \tag{2.6.5}$$

By the definition of block B_2, we have $\ell_{B_2} \ge \ell_{B_2(h)}$, $\forall h = 1, \ldots, |B_2|$. If $\ell_{B_1} > \ell_{B_2}$, none of the nodes in B_2 will be clustered together with the nodes in B_1. If $\ell_{B_1} \le \ell_{B_2}$, (2.6.5) shows that all the nodes of B_1 and B_2 will form a new block. Therefore, blocks will not be broken into pieces but can be further agglomerated. During the merging of multiple chains whose roots have the same parent, no blocks will be agglomerated since the blocks are sorted in a descending way along the merged chain; see the three descendant blocks of the bold block in Figure 2.7 (ii). On the contrary, blocks can be agglomerated with those from the parent chain. Figure 2.7 (iii) shows that after chains merge, the blocks in the merged chain can be further agglomerated with the parent block (the bold one).

These observations motivate us to propose Algorithm 2′, a faster version of Algorithm 2. We avoid repeatedly computing moving averages by partitioning each chain into blocks, storing the size and the average of each block. Specifically, there are three new components we need for Algorithm 2′:

- **Detect breaking points.** For a chain C_r, breaking points can be detected by (2.6.4). Many existing algorithms can be used to this end. For example, recursion leads to an $\mathcal{O}(|C_r| \log |C_r|)$ time complexity. Figure 2.7 (i) illustrates this step.
- **Merge multiple chains with defined blocks**. Merging m multiple chains with detected blocks can be realized using the k-way merge algorithm. The time complexity

is $\mathcal{O}(s \log m)$, where s is the total number of blocks in these chains. Figure 2.7 (ii) illustrates this step using a tree of five blocks.

- **Agglomerate the upstream chain and the downstream merged chain.** For a node $v \in \mathcal{P}_2$, denote by $C^{(v)}$ the longest chain that ends with the node v and all the nodes in $C^{(v)}$ but v have only one child. Suppose the children of v are $r_1, \ldots, r_H$. Denote by C_v the chain output by merging $C_{r_1}, \ldots, C_{r_H}$ using the k-way merge algorithm. Denote the blocks of $C^{(v)}$ by $B_1^{(up)}, \ldots, B_s^{(up)}$ and the blocks of C_v by $B_1^{(down)}, \ldots, B_t^{(down)}$. Algorithm 5 agglomerates the blocks of $C^{(v)}$ and C_v with a time complexity of $\mathcal{O}(|C^{(v)}| + |C_v|)$. Figure 2.7 (iii) illustrates this step using the output of Figure 2.7 (ii).

Throughout Algorithm 2′, the total time complexity consists of three parts: 1) detecting breaking points requires $\mathcal{O}(n \log K)$ computations; 2) merging multiple chains with defined blocks requires $\mathcal{O}(Dn \log K + n \log M)$ computations, where D is the number of nodes that have multiple children in the graph (for one sample). The quantity D upper bounds the number of times each sub-chain merges during the algorithm; 3) agglomerating the upstream chain and the downstream merged chain requires $\mathcal{O}(Dn)$ computations. In total, the time complexity of Algorithm 2′ is $\mathcal{O}(Dn \log K)$. In reality, most tree structures are shallow with $D < 10$. For example, the $D = 6$ and $D = 5$ in Figure 2.9 and Figure 2.10 respectively. So our algorithm is actually of $\mathcal{O}(n \log K)$ run time for practical use.

Algorithm 5 Agglomerate the blocks in the upstream chain and the downstream chain

Input: Blocks $B_1^{(up)}, \ldots, B_s^{(up)}$ from the upstream chain $C^{(v)}$ and Blocks $B_1^{(down)}, \ldots, B_t^{(down)}$ from the downstream chain C_v.

Procedure:

Let b_0 be $B_1^{(down)}$, b_{-1} be the block ahead of b_0 in $C^{(v)}$ and b_{+1} be the block after b_0 in C_v. Denote by $\ell_{b_0}, \ell_{b_{-1}}, \ell_{b_{+1}}$ the averaging LPR within b_0, b_{-1} and b_{+1} respectively.
while $\ell_{b_0} > \ell_{b_{-1}}$ or $\ell_{b_{+1}} > \ell_{b_0}$ **do**
  if $\ell_{b_0} > \ell_{b_{-1}}$ **then**
    Agglomerate b_0 and b_{-1}. The new block is still called b_0 and the block ahead of the original b_{-1} now is called b_{-1}.
  else
    Agglomerate b_0 and b_{+1}. The new block is still called b_0 and the block after the original b_{+1} now is called b_{+1}.
  end if
end while

[1] **Output:** The new sequence of blocks.

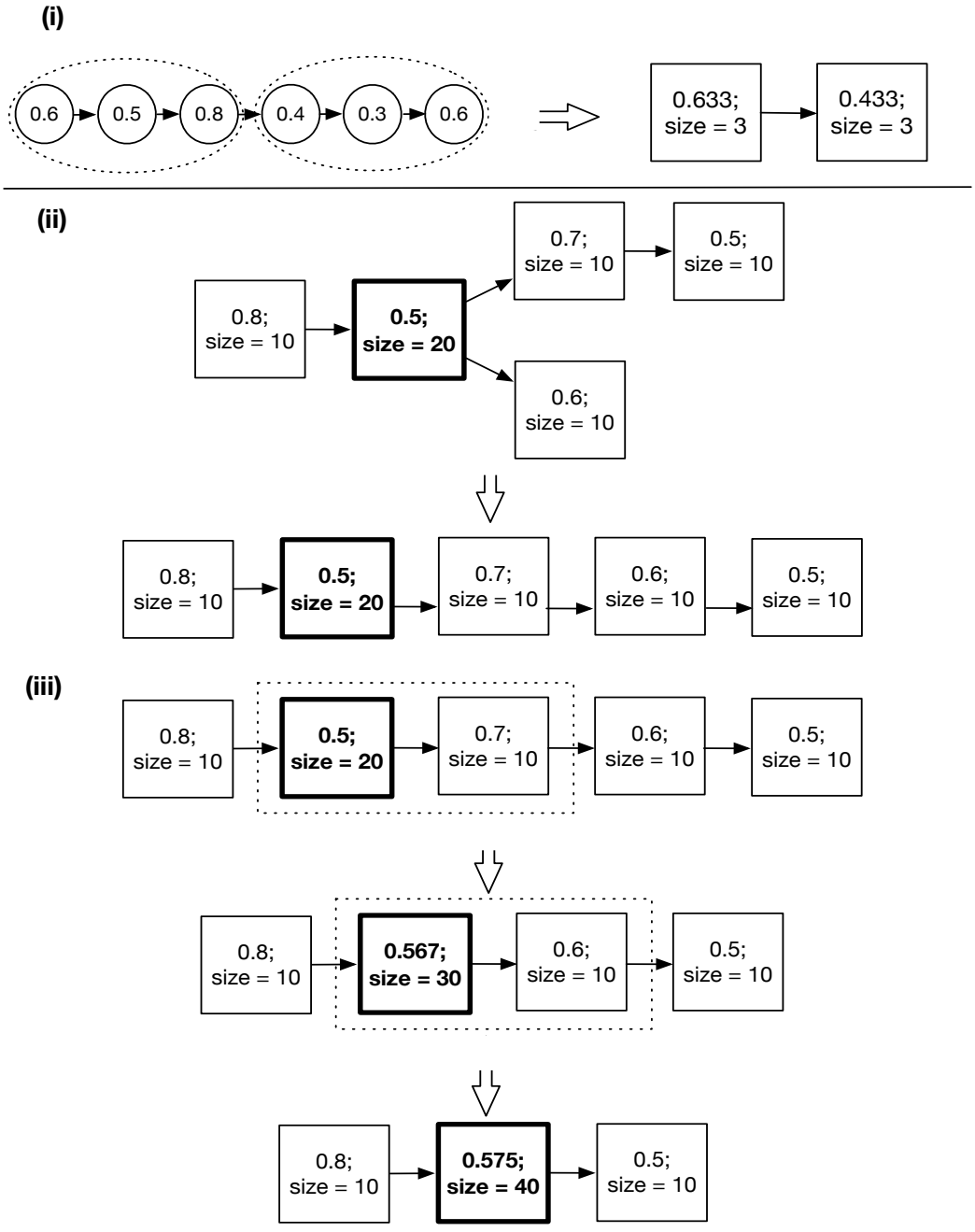


Figure 2.7: Illustrating the three components in Algorithm 2′ using two examples which are separated by the solid line. The first example starts from a tree of six nodes, and the second example starts from a tree of five blocks. (i) Detect breaking points of the chain of six nodes and partition them into two blocks. (ii) Merge the two child chains of the bold block. (iii) Agglomerate the upstream chain and the downstream chain around the bold node. The final list of blocks are positioned in a descending way.

Algorithm 2′ A faster implementation of the HierRank algorithm.

Input: A forest $\mathcal{T}$, scores $\mathbf{S}$.

Procedure:

1: Figure out $\mathcal{P}_2$.
2: **while** $\mathcal{P}_2 \neq \emptyset$ **do**
3: Pop out a v from $\mathcal{P}_2$. Take out all of its children $r_1, \ldots, r_H$. These children's descendants have at most one child. Denote by $C^{(v)}$ the longest chain that ends with the node v and all the nodes in $C^{(v)}$ but v have only one child.
4: Find the breaking points $p_1^{(h)}, \ldots, p_{S_h}^{(h)}$ for C_{r_h} by (2.6.4), $h = 1, \ldots, H$.
5: Merge $C_{r_1}, \ldots, C_{r_H}$, in terms of the averaging score values of the blocks separated by the breaking points. Denote the new chain as C_v.
6: Agglomerate blocks of $C^{(v)}$ and C_v by Algorithm 5.
7: Update $\mathcal{P}_2$.
8: **end while**
9: **if** There remain multiple chains **then**
10: Merge them use the k-way merge algorithm.
11: **end if**
12: Let $\mathcal{L}$ be the resulting chain.

[1] **Output:** a ranking $\mathcal{L}$.

2.6.3 Extension to DAG

Directed acyclic graph (DAG) is a more general hierarchy than the tree structure and is more applicable to real data. In the DAG hierarchy structure, one node can have more than one parent, which brings about an additional decision issue – which parent the node and its descendants should respect. We call it the "AND" constraint if the node respects its all parents and call it the "OR" constraint if the node only respects one of its parents. Denote by $\mathcal{Q}$ all the nodes that have at least two parents.

It is easy to extend our algorithm for the tree hierarchy to the DAG structure by dynamic programming. For each node in $\mathcal{Q}$, we explore all the possible cases where this node respects one of its parents and disconnects the edges to other parents. Such a strategy works for the "OR" constraint. Bi and Kwok, 2011 has shown that at least one case satisfies the "AND" requirement. Each case boils down to a forest; thus, we can use Algorithm 2′ to get a ranking. For the "OR" requirement, we select the case with the highest objective function value. For the "AND" requirement, we select the highest value scenario among those satisfying the requirement. Although the above brute-force strategy looks clumsy and time-consuming, it works for most practical scenarios since most applications have shallow and scattered hierarchy structures. For instance, in Figure 2.9, there are seven nodes that have multiple

parents in a connected part, while there are two such nodes in Figure 2.10. So we only need to explore $2^7 = 128$ cases at most. Considering there are only a limited number of labels, about 100 for most times, the computation time is acceptable.

To adapt HierRank to a complicated DAG with substantial nodes that have multiple parents, we follow the strategy used in CSSA. To be specific, we find the node v with multiple parents and one of its parents has the minimal parent score value:

$$\text{Find } v \in \mathcal{Q} \text{ such that } \min_{u \in pa(v)} S_u = \min_{u \in \cup_{u' \in \mathcal{Q}} pa(u')} S_u.$$

Then we find the parent of v that has the minimal score value:

$$\text{Find } u := \arg\min_{u' \in pa(v)} S_{u'}.$$

Then we assign the parents of v except for u as the new parents of u and disconnect v to its parents but u. Bi and Kwok, 2011 has shown that this strategy works for the "OR" constraint. The detailed algorithm that uses this strategy to extend HierRank to DAG is summarized as Algorithm 5.

Algorithm 5 The HierRank algorithm for the DAG hierarchy.

Input: The DAG graph $\mathcal{G}$, node scores $\mathbf{S}$.

Procedure:

Figure out $\mathcal{P}_2$.
Figure out $\mathcal{Q} := \{v : v \text{ has more than one parent}\}$.
while There is a node with more than one children or more than one parent. **do**
  while $\mathcal{P}_2 \neq \emptyset$ **do**
    Pop out one v from $\mathcal{P}_2$. Take two children of v, r_1 and r_2.
    Feed C_{r_1} and C_{r_2} into Algorithm 1 and obtain $\mathcal{L}(r_1, r_2)$.
    Replace C_{r_1} and C_{r_2} with $\mathcal{L}(r_1, r_2)$.
    Update $\mathcal{P}_2$.
  end while
  if $\mathcal{Q} \neq \emptyset$. **then**
    Find $v \in \mathcal{Q}$ such that $\min_{u \in pa(v)} S_u = \min_{u \in \cup_{u' \in \mathcal{Q}} pa(u')} S_u$. Find $u := \arg\min_{u' \in pa(v)} S_{u'}$.
    Let $pa(u) = pa(u) \cup pa(v)/u$, $pa(v) = u$.
    Update $\mathcal{P}_2$ and $\mathcal{Q}$.
  end if
end while
if There remain multiple chains **then**
  Apply Algorithm 1 to these chains.
end if
Let $\mathcal{L}$ be the resulting chain.

[1] **Output:** a ranking $\mathcal{L}$.

2.7 Evaluation

2.7.1 Setup

In the sequel, a variety of rankings can be produced based on $\widehat{mLPRs}$ (denote by $\widehat{\boldsymbol{mLPR}}$**-Rank** the associated ranking method), which differ in how to estimate $mLPR$s and how to rank $\widehat{mLPRs}$ (the method is implemented in the R language[4]). To obtain $\widehat{mLPRs}$, we estimate $\mathbb{P}(Y_i|Y_{pa(i)})$ and $\mathbb{P}(Y_i)$ using SVM with covariates or the empirical proportions without covariates. We estimate LPR by the local polynomial regression (polyreg) as Jiang et al. (2014), or estimate $\mathbb{P}(S_i|Y_i)$ by modelling $\mathbb{P}(S|Y=0)$ and $\mathbb{P}(S|Y=1)$ as two Gaussian densities (DeCoro, Barutcuoglu, and Fiebrink, 2007). Then the mLPR is estimated in terms of the indpt approximation, the nbh approximation, and the full version. Note that for the indpt approximation, it does not rely on the estimates of $\mathbb{P}(Y_i|Y_{pa(i)})$ and $\mathbb{P}(Y_i)$, so it performs the same regardless of how these quantities are estimated. The ranking is produced via either naive sorting or HierRank based on $\widehat{mLPRs}$.

We compare $\widehat{mLPR}$-Rank to three competing methods of different variants. The first one is simply ranking the raw classifier scores (call the associated ranking method **Raw-Rank**). Next, we consider **HIROM** (Bi and Kwok, 2015) which is the state-of-the-art local HMC classifier. It produces Bayes-optimal predictions that minimize a series of hierarchical risks with a general learning model that is independent of the loss functions. Here we use the hierarchical ranking loss and the hierarchical hamming loss for HIROM, which extends the classic ranking loss and hamming loss to the HMC scenario by considering the hierarchy information. Moreover, we consider another line of efforts for the HMC problem, i.e., the "global" classifier. It solves the classification issue and the hierarchy issue mentioned above simultaneously. Unlike two-stage methods, global methods simultaneously make predictions for the graph rather than on a node by node basis. Here we use **CLUS-HMC**[5] and its variants (Blockeel et al., 2002; Blockeel et al., 2006; Vens et al., 2008), which extend the decision tree for HMC on both tree and DAG label hierarchies. It is a state-of-the-art global HMC classifier. The details of all the above methods are summarized in Table 2.1.

We evaluate $\widehat{mLPR}$-Rank using three HMC datasets: 1) A synthetic dataset with three trees that are comprised of 25 nodes; 2) the disease-gene-expression data (Huang, Liu, and

[4]The implementation can be found in github.com/Elric2718/mLPR.

[5]We use ClusHMC and follow Dimitrovski et al., 2011 by constructing bagged ensembles and use the original settings of Vens et al., 2008, weighting each node equally when assessing distance, i.e. $w_i = 1$ for all i. In addition to node weights, the minimum number of events is set to 5, and the minimum variance reduction is tuned via 5-fold cross-validation from the options 0.60, 0.70, 0.80, 0.90, and 0.95. Following the implementation of Lee, 2013, a default of 100 (Predictive Clustering Trees) PCTs are trained for each ClusHMC ensemble; each PCT is estimated by resampling the training data with replacement and running ClusHMC on the result.

Table 2.1: Details of the competing methods.

Method	Raw-Rank	ClusHMC	HIROM	$\widehat{mLPR}$-Rank
Method type	two-stage (2nd stage)	global	two-stage (1st and 2nd)	two-stage (2nd stage)
Input	Classifier Scores	Labels Y_i's; Covariates	Labels Y_i's; Covariates	Labels Y_i's; Classifier Scores
Estimators of $\mathbb{P}(Y_i)$, $\mathbb{P}(Y_i\|Y_{pa(i)})$	N/A	N/A	SVM, empirical estimator	SVM, empirical estimator
Ranking	naive sorting, HierRank	N/A	HierRank	naive sorting, HierRank
Other variants	N/A	version: vanilla, bagging	loss: Hier. Ranking, Hier. Hamming	Approximation: indpt, nbh, full Estimator of LPR: local polyreg, Gaussian Mixture

Zhou, 2010); 3) the RCV1v2 data (Lewis et al., 2004). We use the truncated area under the PR as the evaluation metric. To be specific, we convert the result into a ranking where the top ones are more likely to be positive. Then we take top $\kappa \times 100\%$ of the samples as positive and the remaining as negative, where $\kappa \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1\}$. For each κ, we get the corresponding recall rate and compute the area under the PR curve truncated at this recall rate.

After obtaining a ranking given by $\widehat{mLPR}$-Rank, the next step is to cut the ranking to make the final decisions. One immediate solution is to resort to the validation set. Suppose the individual classifier for each node in $\mathcal{G}$ and the estimation methods of $\mathbb{P}(Y_i)$'s, $\mathbb{P}(Y_i|Y_{pa(i)})$s and LPRs are trained on the training set. Then we can determine the cutoffs on the validation set. Specifically, the classifiers and the estimation methods of LPRs, $\mathbb{P}(Y_i)$, $\mathbb{P}(Y_i|Y_{pa(i)})$ learned from the training set can be applied to the validation set to produce $\widehat{mLPR}$s. Then HierRank/naive sorting takes these $\widehat{mLPR}$s to produce a ranking of nodes for the validation set. Since the truth is known on this dataset, metrics like F-measure and false discovery rate (FDR) can be computed at an arbitrary cutoff. Then the desired cutoff on this ranking can be chosen to attain the maximal F-measure or a target FDR. The evaluation of the goodness of this cutoff is performed on the testing set. For other methods in Table 2.1, the same strategy can be used to select a cutoff.

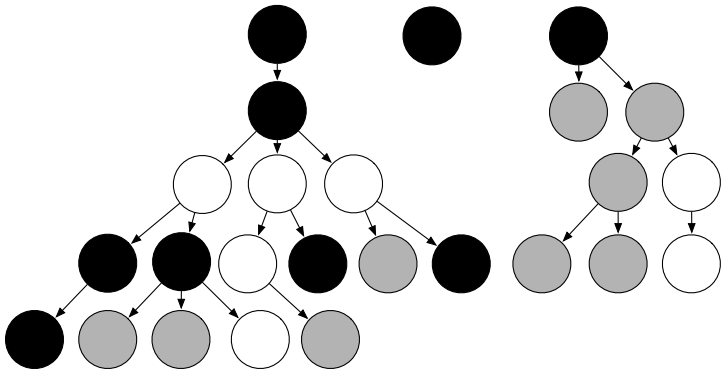
2.7.2 A synthetic data with a complicated tree structure

For the synthetic dataset, the setting comprises three trees with mixes of high- and low-quality nodes and varying levels of dependence between the nodes (Figure 2.8). The quality of a node refers to the ability of the corresponding classifier to distinguish between the positive and the negative. The simulation dataset consists of 5,000 training samples and 1,000 test samples. We generate the true instance status as follows. First, the conditional probabilities $\mathbb{P}(Y_i = 1|Y_{pa(i)} = 1)$'s are randomly generated from a uniform distribution, with the constraint that each dataset has to have a minimum of 15 positive events in the training set, which amounts to a minimum prevalence of 0.3% for any class. Then, given the instance status, the simulated classification score is sampled from the status-specific distribution — data are generated from a Beta(η, 6) distribution for the negative case and a Beta(6, η) distribution for the positive case, where $\eta = 2,\ 4,\ 5.5$ for the high, medium, low node quality respectively. Details of the score generation mechanism can be found in Table 2.2.

Table 2.2: Score distribution in terms of the node quality.

Quality	Positive instance	Negative instance	Node color
High	Beta$(6, 2)$	Beta$(2, 6)$	white
Medium	Beta$(6, 4)$	Beta$(4, 6)$	grey
Low	Beta$(6, 5.5)$	Beta$(5.5, 6)$	black

Figure 2.8: A 25-nodes tree-hierarchy. White, grey, and black correspond to high, medium, and low quality, respectively.



Since there is no covariate for each sample, we just use empirical proportions as the estimates of $\mathbb{P}(Y_i|Y_{pa(i)})$ and $\mathbb{P}(Y_i)$. From Table 2.3, we see that $\widehat{mLPR}$-Rank works best for the HMC task. Among variants of $\widehat{mLPR}$-Rank, the full version outperforms that of the

neighborhood approximation. The independence approximation is the worst. We interpret this result as that the sample size is sufficiently large, and the data quality is sufficiently good to learn the estimators of the LPRs, $\mathbb{P}(Y_i|Y_{pa(i)})$'s and $\mathbb{P}(Y_i)$'s accurately. Therefore, the hierarchy information can be well learned when estimating the mLPRs. One direct evidence, based on Proposition 1, is that $\widehat{mLPR}$-Rank of the full version using the naive sorting performs almost as well as that using HierRank. Finally, $\widehat{mLPR}$-Rank with LPRs learned by modelling $\mathbb{P}(S|Y)$ as Gaussian densities is inferior to those of the other methods. It shows the advantage of using LPRs in Formula (2.5.1) (d) instead of $\mathbb{P}(S|Y)$ in Formula (2.5.1) (b) — it is more robust and more accurate.

Given the ranking produced by $\widehat{mLPR}$-Rank, the next step is to determine the cutoff to make the final decisions. To this end, we split the original training set into a training set and a validation set of equal sizes (2500 samples) and then use the cutoff selection approach discussed at the end of Section 2.7.1. In Table 2.4 and Table 2.5, we show the performance of this strategy. The cutoff is taken to attain an $\alpha \times 100\%$ FDR ($\alpha = 0.01, 0.05, 0.1, 0.2$) or the maximal F-score on the validation set. Then the same cutoff is applied to the testing set. We see that the observed false discovery proportion (FDP) on the testing set is close to the target one for every method except for the Raw-Rank method. For the F-score, the strategy also finds out the nearly maximal value for each method. These results indicate the reliability of the cutoff selection strategy. On the other hand, the results also corroborate the conclusion drawn from Table 2.3: in Table 2.4, $\widehat{mLPR}$-Rank of the full version finds most discoveries before exceeding the given FDR; in Table 2.5, $\widehat{mLPR}$-Rank of the full version gives the ranking with the highest maximal F-score.

2.7.3 Disease Diagnosis

Huang, Liu, and Zhou (2010) developed a classifier for predicting disease along the UMLS directed acyclic graph, trained on public microarray datasets from the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO). They collected 100 studies, including a total of 196 datasets and 110 disease labels. The 110 labels represent 110 nodes, which are grouped into 24 connected DAGs; see Figure 2.9 and 2.10. In general, the graphs have three properties:

- It is shallow rather than deep: the maximum node depth is 6, though the median is 2. Only 10 nodes have more than one child. It occurs because 11 of the connected sets are standalone nodes, while six are simple two-node trees. The two largest sets consist of 28 and 30 nodes, respectively.

- It is scattered rather than highly connected. The graph nearly follows a tree structure. Most nodes have only one parent or are at the root level. Only 15 nodes have 2 parents,

Table 2.3: The recall rate and the area under the PR curve for the synthetic data. Here κ refers to the proportion of events that are classified as positive. All the values are in percentage. The highest values are highlighted in each column.

		obtained recall rates					obtained truncated PR-AUC				
	$\kappa \times 100$	5.0	10.0	20.0	30.0	50.0	5.0	10.0	20.0	30.0	50.0
Raw-Rank	naive sorting	5.3	8.5	14.0	20.2	35.4	1.4	1.8	2.3	2.9	4.3
	HierRank	5.1	13.5	30.4	45.5	69.1	0.9	1.8	5.1	8.1	12.7
ClusHMC	vanilla	32.7	54.4	76.6	85.8	93.9	30.0	48.5	61.8	65.8	68.4
	bagged	33.9	56.7	76.8	86.5	94.3	31.7	50.8	63.6	67.9	70.4
HIROM	hier.ranking	35.5	55.6	81.1	84.1	88.7	33.7	56.1	70.6	71.9	73.2
	hier.hamming	35.7	59.7	85.4	89.6	92.6	34.4	54.9	73.0	75.1	76.1
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + HierRank)	indpt	35.7	62.3	82.8	89.9	95.8	34.7	58.5	72.8	76.1	78.1
	nbh	36.5	64.0	85.7	92.8	97.6	36.0	61.3	77.0	80.4	82.1
	full	**36.6**	**64.7**	**86.8**	**93.9**	**98.6**	**36.2**	**62.1**	**78.3**	**81.8**	**83.4**
$\widehat{mLPR}$-Rank (Gaussian + HierRank)	indpt	34.7	53.7	76.7	87.8	94.3	33.6	49.5	62.7	67.8	69.9
	nbh	34.0	59.8	81.7	89.5	95.3	33.9	54.8	69.8	73.5	75.4
	full	35.2	60.2	81.6	89.9	96.3	34.4	56.1	70.8	74.7	76.8
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + naive sorting)	indpt	35.5	60.7	82.2	89.2	95.6	34.5	56.6	71.4	74.7	76.8
	nbh	36.4	63.8	85.5	92.6	97.5	34.9	60.9	76.7	80.1	81.8
	full	**36.6**	**64.7**	**86.8**	**93.9**	**98.6**	**36.2**	**62.1**	**78.3**	**81.8**	**83.4**

Table 2.4: The false discovery proportion (FDP) on the synthetic testing dataset, which is obtained by the cutoff determined at a FDR on the validation set. All the values are in percentage. For the "prop. of discoveries before the cutoff" panel, the highest values are highlighted in each column.

		observed FDP				prop. of discoveries before the cutoff			
	Target FDR	1.0	5.0	10.0	20.0	1.0	5.0	10.0	20.0
Raw Scores	naive sorting	0.0	0.0	31.3	37.5	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2.6 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$
	HierRank	0.0	5.1	9.5	19.5	$2 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	0.1	1.0
ClusHMC	vanilla	0.0	0.0	9.7	19.2	$7 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	3.5	7.5
	bagged	0.0	5.1	9.6	19.6	$2 \cdot 10^{-3}$	2.2	4.6	8.4
HIROM	hier.ranking	0.0	5.1	9.7	19.0	$2 \cdot 10^{-2}$	2.9	6.2	10.3
	hier.hamming	3.5	5.1	9.5	19.5	0.3	4.6	6.3	8.3
$\widehat{mLPR}$-Rank (mLPR + HierRank)	indpt	1.1	5.0	9.2	18.8	1.3	4.4	7.0	10.5
	nbh	0.8	4.7	9.3	19.1	2.4	5.7	8.0	11.2
	full	0.6	4.4	9.6	19.4	**2.7**	**5.8**	**8.3**	**11.5**
$\widehat{mLPR}$-Rank (Gaussian + HierRank)	indpt	1.5	5.1	9.6	19.5	1.3	3.8	5.4	8.0
	nbh	1.3	4.7	9.8	19.1	1.9	3.3	4.7	9.4
	full	1.1	5.0	9.9	19.4	2.2	4.2	6.0	9.7
$\widehat{mLPR}$-Rank (mLPR + naive sorting)	indpt	1.3	4.6	9.3	19.4	1.5	4.1	6.4	9.9
	nbh	0.8	4.6	9.5	19.3	2.5	5.6	7.9	11.1
	full	0.6	4.4	9.6	19.4	**2.7**	**5.8**	**8.3**	**11.5**

Table 2.5: The F-score on the synthetic testing dataset, which is obtained by the cutoff determined at the maximal F-score on the validation set. All the values are in percentage. The lowest values are highlighted in the "prop. of samples before the cutoff" column, while the highest values are highlighted in other columns.

		prop. of samples before the cutoff	maximal F-score of the ranking	obtained F-score
Raw Scores	naive sorting	47.3	29.1	29.0
	HierRank	98.8	23.3	23.3
ClusHMC	vanilla	14.9	64.4	64.3
	bagged	**11.6**	70.1	70.0
HIROM	hier.ranking	13.6	73.2	73.2
	hier.hamming	13.7	72.0	71.9
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + HierRank)	indpt	11.9	71.5	71.4
	nbh	12.4	74.2	74.2
	full	12.8	**74.9**	**74.8**
$\widehat{mLPR}$-Rank (Gaussian + HierRank)	indpt	18.1	61.5	61.4
	nbh	12.9	70.2	70.2
	full	13.7	70.2	70.1
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + naive sorting)	indpt	12.2	70.4	70.4
	nbh	12.4	73.9	73.9
	full	12.8	**74.9**	**74.8**

and 2 nodes have 3 parents. Most nodes do not have a high positive case prevalence. The largest number of samples belonging to a label is 62, or a 32.63% positive case prevalence. The average prevalence is 5.89%, with a minimum prevalence of 1.53%, corresponding to 3 cases for a label.

- Data redundancy occurs as an artifact of the label mining: usually, the positive events for a disease concept are the same for its parents. Few datasets are tagged with a general label and not a leaf-level one. Twenty-six nodes or 23.64% of all nodes share the same data as their parents, so they have the same classifier, and therefore the identical classifier scores as their parents. If we take the number of nodes that share more than half of their data with their parent, this statistic rises to 50%. A consequence of this redundancy is that the graph is shallower than appears in the figure. For example, the first connected set in the top left of Figure 2.10 appears to have six levels but only has three because the last three levels do not contain any new information.

We simply follow Huang, Liu, and Zhou (2010) to get the first-stage classifier scores. We summarize that process here. In the classifier for a particular disease concept, the negative training events were the profiles among the 196 that did not match with that disease concept. The principal modeling step involved expressing the posterior probability of belonging to a label in terms of the log-likelihood ratio and some probabilities that have straightforward empirical estimates. The log-likelihood ratio was modeled with a log-linear regression. A posterior probability estimate, which would be used as the first stage classifier score, was then obtained for each of the 110×196 events in the data by leave-one-out cross-validation (LOOCV), i.e., estimating the i-th posterior probability based on the remaining ones. It guarantees that, for each class, the classification scores are identically distributed across all samples (see the related discussion in Section 2.7.4). Next, we use another round of LOOCV to compute the mLPRs. Since the disease-gene-expression data has very limited sample size, we just use the empirical proportion for $\mathbb{P}(Y_i|Y_{pa(i)})$ and $\mathbb{P}(Y_i)$. Finally, we apply HierRank on these mLPRs to produce the ranking.

We compare the performance of $\widehat{mLPR}$-Rank against other competing methods (other methods are executed in the same LOOCV fashion as above). The resulting precision-recall curve is shown in Figure 2.11. Overall, $\widehat{mLPR}$-Rank performs better than all of the other methods, and it performs significantly better in the initial portion of the precision-recall curve. It is not a surprise that the indpt approximation performs better than the nbh approximation and the full version since it is hard to estimate $\mathbb{P}(Y_i|Y_{pa(i)})$ and $\mathbb{P}(Y_i)$ due to the limited sample size. In this case, the mLPR estimation does not fully incorporate the hierarchy information, and the assumption of Proposition 1 is violated. Thus, HierRank plays an important role in making use of the hierarchy. In contrast, Raw-Rank with HierRank poorly behaves because the first-stage classifiers are not good enough on their associated classes,

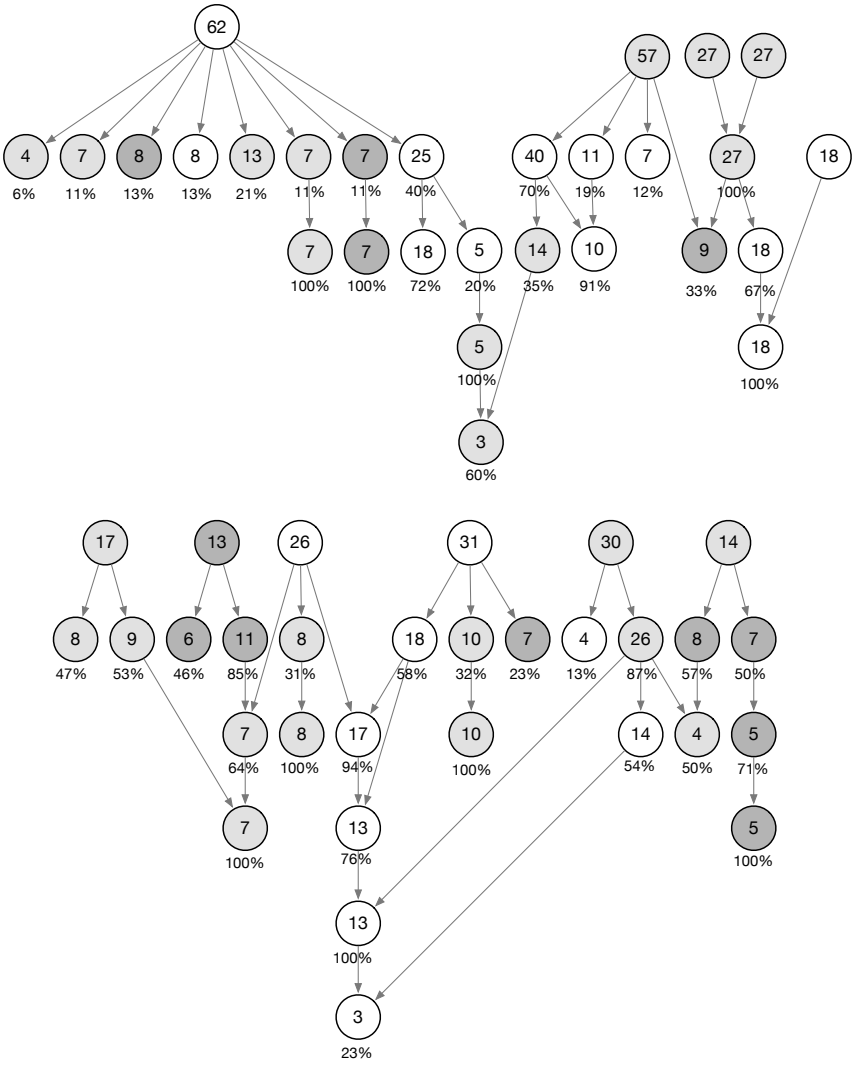


Figure 2.9: Structure of the disease diagnosis dataset, part 1 of 2. The colors correspond to node quality: white indicates that a node's base classifier has AUC between $(0.9, 1]$; light grey, $(0.7, 0.9]$, dark grey, $(0, 0.7]$. The value inside a circle indicates the number of positive cases, while the value underneath gives the maximum percentage of cases shared with a parent node.

and these classification scores are not statistically comparable across classes. HIROM does not work well due to the poor estimators of $\mathbb{P}(Y_i|Y_{pa(i)})$'s and $\mathbb{P}(Y_i)$'s. ClusHMC is better but inferior to $\widehat{mLPR}$-Rank since it is not able to handle disconnected hierarchies.

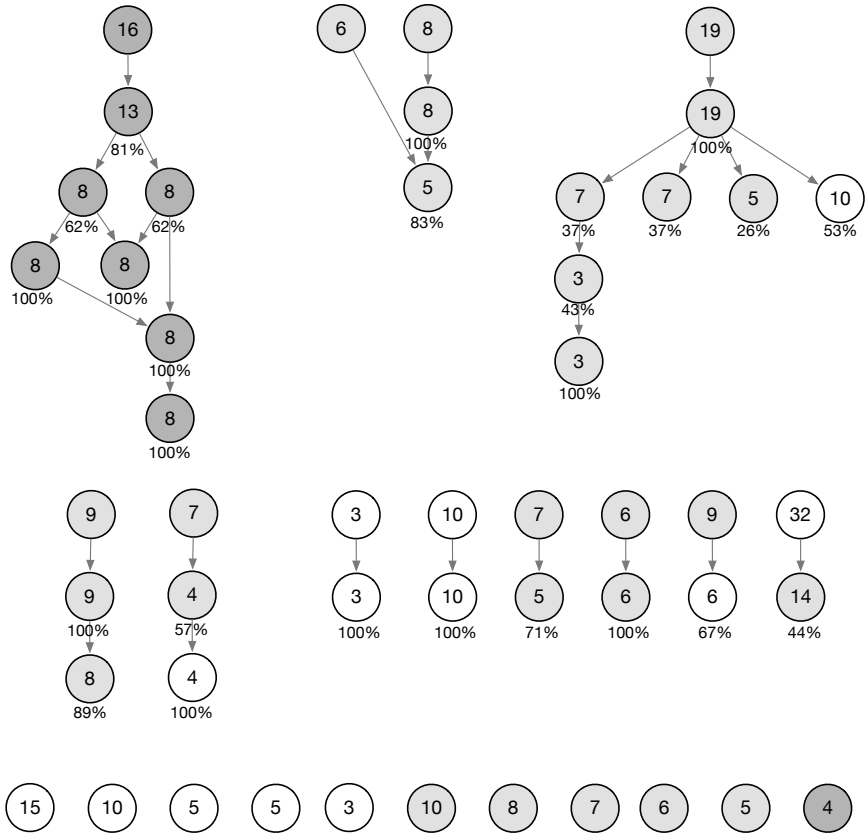
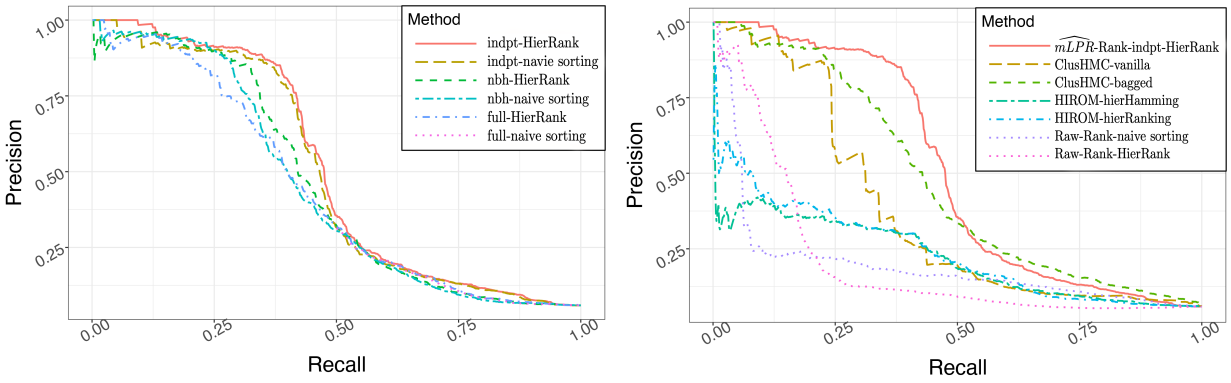


Figure 2.10: Structure of the disease diagnosis dataset, part 2 of 2. The colors correspond to node quality: white indicates that a node's base classifier has AUC between $(0.9, 1]$; light grey, $(0.7, 0.9]$, dark grey, $(0, 0.7]$. The value inside a circle indicates the number of positive cases, while the value underneath gives the maximum percentage of cases shared with a parent node.

2.7.4 RCV example

In this section, we consider a text categorization task using the Reuters Corpus Volume I (RCV1) dataset, which is an archive of over $800,000$ manually categorized newswire stories made available by Reuters, Ltd. To be more specific, we use the corrected version RCV1v2 (Lewis et al., 2004), which describes the coding policy and quality control procedures used in producing the RCV1 data, the intended semantics of the hierarchical category taxonomies, and the corrections necessary to remove errorful data. The RCV1v2 dataset used here contains $30,000$ samples in total and 103 categories. The data has a good quality, and the categorization task is not difficult.



(a) Variants of $\widehat{mLPR}$-Rank (b) All Methods

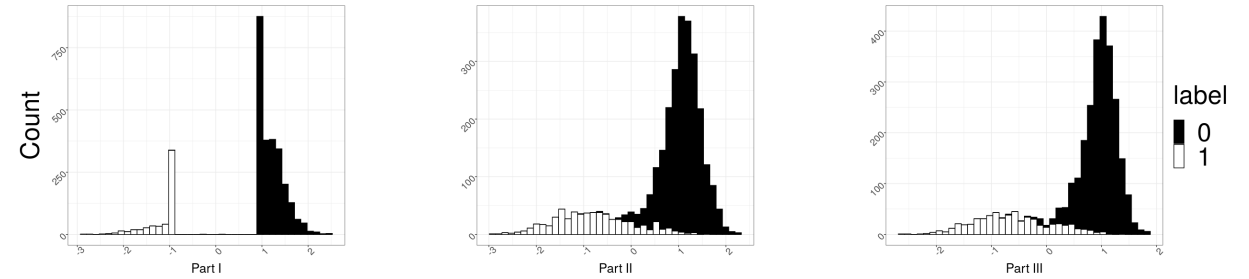
Figure 2.11: Precision-recall curve for several classifiers run on the real dataset of Huang, Liu, and Zhou (2010).

We use this example mainly to illustrate a subtlety in the training process of $\widehat{mLPR}$-Rank. In this method, we have two training stages: 1) train the binary classification scores using SVM, 2) train the LPR model using the SVM scores and train learners to estimate $\mathbb{P}(Y_i)$'s and $\mathbb{P}(Y_i|Y_{pa(i)})$'s. For a fair evaluation, we split the dataset into three partitions: Partition I (25% of the samples), Partition II (25% of the samples), and Partition III (50% of the samples). First, we train the SVM model on partition I, and then predict the classification scores on each partition (Figure 2.12c (a)). The distributions of the classification scores on Partition I differ far from those on Partition II and Partition III, while the latter two distributions are quite similar. It is reasonable since Partition I is the training dataset and Partition II& III are the testing dataset for the first stage. For the estimation of the LPR (and other quantities), there are two possible training strategies:

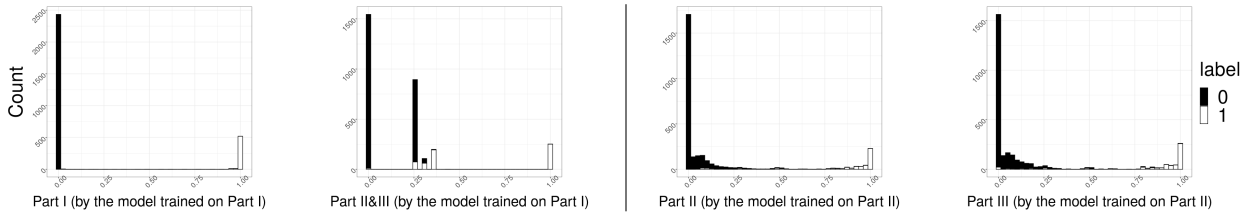
1. Train the model for the LPR on Partition I and then predict the LPR scores on Partition II & III.

2. Train the model for the LPR on Partition II and then predict the LPR scores on Partition III.

We suggest using the second strategy. To elaborate on this point, we investigate the predicted LPR scores for both strategies. As shown in Figure 2.12c (b), for the first strategy, the distributions of the estimated LPRs are mixed between the positive and the negative groups on Partition II & III. In contrast, the distributions are clearly separated on Partition I. It results from the fact that the distribution of the input SVM scores on Partition I deviates far from that on Partition II & III. It leads to a bad generalization from the training data to the



(a) SVM scores.



(b) LPR scores.

(c) Predicted SVM/LPR scores. (a) The SVM model is trained on Partition I, and predicted on all partitions. (b) The LPR models are respectively trained on Partition I then predicted on all partitions (left), and trained on Partition II then predicted on Partition II & III (right).

testing data. By contrast, the second strategy overcomes this issue by training on Partition II and testing on Partition III, which have similar distributions of the SVM scores. As a result, the distributions of the estimated LPRs between the two groups are well separated on both Partition II and III. Similar phenomena are observed for the estimation of $\mathbb{P}(Y_i)$ and $\mathbb{P}(Y_i|Y_{pa(i)})$.

Using the second strategy for data splitting, we evaluate $\widehat{mLPR}$-Rank against the competing methods on the RCV1v2 dataset, as shown in Table 2.6. Almost all the methods, except for the Raw-Rank methods, can find a majority of correct positives in the very beginning since it is easy to classify texts in this dataset. It can be seen that $\widehat{mLPR}$-Rank outperforms all the other methods, which justifies our argument that the full consideration of the hierarchy is significantly beneficial. Also, we observe that the full version performs better than the indpt approximation and the nbh approximation.

Finally, we need to point out that we trim the factor $1/\hat{\mathbb{P}}(Y_i)$ since it can be pretty unstable if $\hat{\mathbb{P}}(Y_i)$ is close to 0. This strategy has been commonly adopted in statistics and machine learning, e.g., the Iterated Probability Weights method in causal inference (Lee, Lessler, and Stuart, 2011) and varieties of deep neural networks (Pascanu, Mikolov, and Bengio, 2013) use the clipping trick.

Table 2.6: The recall rate and the area under the PR curve for the RCV1v2 data. Here κ refers to the proportion of events that are classified as positive. All the values are in percentage. The highest values are highlighted in each column.

		obtained recall rates					obtained truncated PR-AUC				
	$\kappa \times 100$	5.0	10.0	20.0	30.0	50.0	5.0	10.0	20.0	30.0	50.0
Raw Scores	naive sorting	4.0	5.3	6.9	8.5	13.4	0.5	0.6	0.6	0.6	0.6
	HierRank	7.3	11.1	17.5	23.6	35.9	1.2	1.3	1.5	1.7	2.0
ClusHMC	vanilla	68.5	80.0	88.2	90.8	93.6	55.1	59.0	60.6	60.9	61.1
	bagged	72.5	83.7	92.0	95.7	**98.3**	61.6	65.6	67.3	67.7	67.9
HIROM	hier. ranking	77.1	80.0	85.9	89.1	92.1	61.5	62.5	63.6	63.9	64.1
	hier. hamming	75.4	88.8	91.7	93.8	96.1	57.0	62.2	62.8	63.0	63.2
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + HierRank)	indpt	75.8	86.6	92.7	95.2	96.3	63.4	67.5	68.8	69.1	69.2
	nbh	**78.0**	**88.9**	**94.1**	96.5	97.1	**66.7**	70.9	**72.0**	72.3	72.4
	full	77.5	**88.9**	93.6	**97.0**	97.6	66.5	**71.0**	**72.0**	**72.4**	**72.5**
$\widehat{mLPR}$-Rank ($\widehat{mLPR}$ + naive sorting)	indpt	74.9	85.8	92.4	94.4	95.1	62.2	66.3	67.6	67.8	67.9
	nbh	77.8	88.8	93.5	**97.0**	97.8	66.4	70.7	71.7	72.1	72.2
	full	77.5	**88.9**	93.9	96.8	97.7	66.5	70.9	**72.0**	**72.4**	72.4

2.8 Discussion

In this chapter, we present a method that produces the ranking for the second-stage decision in the two-stage HMC method. When true mLPRs are accessible, the resulting ranking obtained by sorting mLPRs in descending order maximizes the objective function CATCH and meanwhile satisfies the hierarchical consistency and has a nice property that a front node in the ranking is more likely to be positive than a tail node (Proposition 3). In practical implementation, we have to resort to estimating mLPRs, which might no longer enjoy these properties. To this end, the ranking algorithm HierRank is developed. It has been theoretically shown to optimize the empirical CATCH (given estimated mLPRs) under the hierarchical constraint.

We demonstrate the advantage of our approach over the competing methods in one simulation study and two real data studies. Our method outperforms the competing methods constantly in terms of the truncated area under the PR curve. We also provide an approach to select a cutoff for the final decision on the ranking. It has been shown that this cutoff selection method can ensure the FDR control or maximize the F-score. Our method finds more discoveries than other methods before exceeding the target FDR. In addition, our method can obtain a higher maximal F-score than other methods. Both results imply that the ranking produced by our method puts more true positives in the front than other rankings. For the above reasons, we recommend our method as a computationally efficient, statistically driven approach that produces a ranking for the second-stage decision in a local classification framework.

We also provide a practical guideline in training $\widehat{mLPR}$-Rank. First, we use different versions of approximations, in terms of the data quality and the sample size, to estimate mLPRs given the first stage classifier scores. The independent version fits when the data is noisy or insufficient; otherwise, a full version is preferred. Second, we should split the training data into two parts, one for the learning of the first-stage classifiers, the other one for the learning of LPR scores, $\mathbb{P}(Y_i|Y_{pa(i)})$'s and $\mathbb{P}(Y_i)$'s. Third, we should use the trimming trick on $\mathbb{P}(Y_i)$ to avoid the inflation of its reciprocal $1/\mathbb{P}(Y_i)$ used to compute mLPR.

Despite the merits of $\widehat{mLPR}$-Rank mentioned above, there remains large room for improvement. First, we note that $\widehat{mLPR}$-Rank of the independent approximation, neighborhood approximation, and the full version perform differently in terms of the data quality and the sample size. It will be useful to have a deep theoretical understanding of how these factors affect the choice of the three types of mLPRs. Second, rather than using CATCH as the objective function in the second stage, it is of great interest to use it as an objective function to train an end-to-end classification system while taking into account the graph hierarchy.

Chapter 3

Binomial Mixture Model With U-shape Constraint

3.1 Introduction

To devise a cost-effective method that is able to yield a conservative cutoff for the GeneFishing method (see Section 1.2 for details), we use the binomial mixture model to model the output of this method. Specifically, we assume there is an underlying fishing rate s_i, reflecting the probability that the i-th gene is fished out in each GeneFishing round. The fishing rates are assumed to be independently sampled from the same distribution F. We mention that the independence assumption is raised mainly for convenience but not realistic de facto since genes may be interactive and correlated in the same pathways or even remotely. Consequently, the effective sample size is smaller than expected. However, this assumption is still acceptable by assuming only a handful of candidate genes are related to the bait genes, which is reasonable from the biological perspective. Furthermore, in Figure 1.2, there is a clear pattern that the histogram is decreasing on the left-hand side and increasing on the right-hand side for all four tissues. In the middle, liver and transverse colon display sparse densities while artery coronary and testis exhibit flat ones. Thus, we can impose a U shape constraint on the associated density of F (see Section 3.4 for details). Then the original problem becomes finding out the cutoff where the flat middle part transits to the increasing part on the right-hand side.

3.1.1 Binomial Mixture Model

We first take a review of the binomial mixture model that has received a lot of attention since the late 1960s. In the field of performance evaluation, Lord (1969), Lord and Cressie (1975), and Sivaganesan and Berger (1993) utilized this model to address the problem of

psychological testing. Thomas (1989) used a two-component binomial mixture distribution to model the individual differences in children's performance on a classification task. Grilli, Rampichini, and Varriale (2015) employed a finite binomial mixture to model the number of credits gained by freshmen during the first year at the School of Economics of the University of Florence. In addition, the binomial N-mixture model is commonly applied to analyze population survey data in ecology. Royle (2004), Kéry (2008), O'Donnell, Thompson III, and Semlitsch (2015), and Wu et al. (2015) estimated absolute abundance while accounting for imperfect detection using binomial detection models. The binomial N-mixture model was also used to estimate bird and bat fatalities at wind power facilities (McDonald et al., 2020).

Formally, we say X is a random variable which has a binomial mixture model if

$$X \sim \int \text{Binomial}(m, s) dQ(m, s) \tag{3.1.1}$$

where $Q(\cdot, \cdot)$ is a bivariate measure of the binomial size m and the success probability s on $\mathbb{N} \times [0, 1]$. In the field of population survey in ecology, m is modeled as Poisson or negative binomial random variable while s is modeled as a beta random variable, linked to a linear combination of additional covariates by a logistic function (Royle, 2004; Kéry, 2008; O'Donnell, Thompson III, and Semlitsch, 2015; Wu et al., 2015; McDonald et al., 2020). Such models are always identifiable thanks to the parametric assumptions. In the field of performance evaluation, m is always known, thus (3.1.1) is reduced to

$$\begin{cases} s \sim F, \\ X|s \sim \text{Binomial}(m, s), \end{cases} \tag{3.1.2}$$

where F is a probability distribution on $[0, 1]$. For instance, m refers to the number of questions of a psychological test in Lord (1969), Lord and Cressie (1975), and Sivaganesan and Berger (1993). The univariate probability distribution F can correspond to a finite point mass function (pmf) as in Thomas (1989) and Grilli, Rampichini, and Varriale (2015) or correspond to a density as in Lord (1969), Lord and Cressie (1975), and Sivaganesan and Berger (1993). Such models suffer from an unidentifiability issue as discussed in Section 3.1.2.

In this study, we are interested in a regime unlike that for performance evaluation where m is known and small (restricted by the intrinsic nature of the problem) and that for population survey in ecology where m is unknown. We concern (3.1.2) with a known m that can be relatively large compared to n, which has not been investigated before in the literature. Notably, suppose there are n objects, e.g., genes; we can determine the parameter m on our own (as in GeneFishing), and then for each object an observation X_i or $\hat{s}_i := X_i/m$ $(i = 1, \ldots, n)$ is i.i.d generated from the binomial mixture model (3.1.2).

3.1.2 The Identifiability Issue

Before diving into the binomial mixture model (3.1.2) with a large m, we first review existing results for mixtures of distributions and binomial mixture model in the literature. A general mixture model is defined as

$$H(x) = \int_{\Omega} h(x|\theta) dF(s), \tag{3.1.3}$$

where $h(\cdot|s)$ is a density function for all $s \in \Omega$, $h(x|\cdot)$ is a Borel measurable function for each x and G is a distribution function defined on Ω. The family $h(x|s)$, $s \in \Omega$ is referred to as the kernel of the mixture and F as the mixing distribution function. In order to devise statistical procedures for inferential purposes, an important requirement is the identifiability of the mixing distribution. Without this condition, it is not meaningful to estimate the distribution either non-parametrically or parametrically. The mixture H defined by (3.1.3) is said to be identifiable if there exists a unique F yielding H, or equivalently, if the relationship

$$H(x) = \int_{\Omega} h(x|s) dF_1(s) = \int_{\Omega} h(x|s) dF_2(s)$$

implies $F_1(s) = F_2(s)$ for all $s \in \Omega$.

The identifiability problems concerning finite and countable mixtures (i.e. when the support of F in (3.1.3) is finite and countable respectively) have been investigated by Teicher (1963), Patil and Bildikar (1966), Yakowitz and Spragins (1968), Tallis (1969), Fraser, Hsu, and JJ (1981), Tallis and Chesson (1982), and Kent (1983). Examples of identifiable finite mixtures include: the family of Gaussian distribution $\{N(\mu, \sigma^2), -\infty < \mu < \infty, 0 < \sigma^2 < \infty\}$, the family of one-dimensional Gaussian distributions, the family of one-dimensional gamma distributions, the family of multiple products of exponential distributions, the multivariate Gaussian family, the union of the last two families, the family of one-dimensional Cauchy distributions, etc.

For sufficient conditions for identifiability of arbitrary mixtures, Teicher (1961) studied the identifiability of mixtures of additive closed families, while Barndorff-Nielsen (1965) discussed the identifiability of mixtures of some restricted multivariate exponential families. Lüxmann-Ellinghaus (1987) has given a sufficient condition for the identifiability of a large class of discrete distributions, namely that of the power-series distributions. Using topological arguments, he has shown that if the family in question is infinitely divisible, mixtures of this family are identifiable. For example, Poisson, negative binomial, logarithmic series are infinitely divisible, so arbitrary mixtures are identifiable.

On the other hand, despite being a power-series distribution, the binomial distribution is not infinitely divisible. So its identifiability is not established for the success parameter (Sapatinas, 1995). In fact, the binomial mixture has often been regarded as unidentifiable, as

F can be determined only up to its first m moments when H is known exactly. To be more specific, $h(x|s)$ is a linear function of the first m moments $\mu_r = \int_0^1 s^r dF(s), r = 1, 2, \ldots, m$ of $F(s)$, for every x. Therefore, with the same first m moments, any other $F'(s)$ will make the same mixed distribution $H(x)$. To ensure the identifiability for the binomial mixture model, it is a common practice to assume that F corresponds to a finite discrete pmf or a parametric density (e.g., beta distribution).

In particular, there are two results for the identifiability of binomial mixture model (Teicher, 1963):

1. If $h(x|s)$ in (3.1.3) is a binomial distribution with a known binomial size m, and the support of F only contain K points. A necessary and sufficient condition that the identifiability holds is that $m \geq 2K - 1$.

2. Consider $h(x|m_j, s_j)$ as a binomial distribution with binomial size m_j and probability s_j, where $0 < s_j < 1$, $j = 1, 2, \ldots$ and the support of F is $\{s_1, s_2, \ldots\}$. If $m_j \neq m_{j'}$ for $j = j'$, then (3.1.3) is identifiable.

In this study, we are interested in the regime where the support size of F may not be finite, and thus the identifiability may fail for the binomial mixture model. Some efforts are made without identifiability. Lord and Cressie (1975) and Sivaganesan and Berger (1993) constructed credible intervals for the Bayes estimators of each point mass and that of F, which rely on the lower order moments of the mixing distribution. Wood (1999) empirically shows that their proposed nonparametric maximum likelihood estimator of F is unique with high probability when m is large. However, these results are far from satisfying in terms of our ultimate goal — estimate or infer the underlying distribution F.

3.1.3 Goals

In this study, the goal is to estimate and infer the underlying distribution F on $[0, 1]$. We have two specific questions to answer:

1. Considering that $m = \infty$, the binomial mixture model (3.1.2) is trivially identifiable, is there a minimal m such that we can have an "acceptable" estimator of F under various conditions:

   - General F without additional conditions.
   - F with a density.
   - F with a continuously differentiable density.
   - F with a monotone density.

- F with a U-shape density.

2. Suppose the underlying distribution F has a density of a U shape; how to make a decision (on the CFR cutoff for GeneFishing) based on the data generated from the binomial mixture model?

When m is sufficiently large, the binomial randomness plays little role, and there is no difference between $\hat{s}_i$ and s_i for estimation or inference purposes. Therefore, the first question naturally arises when looking into the binomial mixture with a large m. This question is related to the identifiability issue of the binomial mixture model. After we obtain some insights into the first question, we are ready to answer the second question that motivates this study, i.e., the cutoff selection problem for GeneFishing.

3.1.4 Main contributions

Our contributions are twofold, which correspond to the two questions raised in Section 3.1.3. One tackles the identifiability issue for the binomial mixture model when m can be relatively large compared to n. The other one answers the motivating question — how to select the cutoff for the output of GeneFishing.

3.1.4.1 New results for large m in Binomial mixture model

Based on the results of Teicher (1963) and Wood (1999), the only hope is to use a large m if we want to solve, or at least alleviate, the identifiability issue for arbitrary mixtures of binomial distributions. We show that regardless of the identifiability of the model (3.1.2), we can find an estimator of F, according to the conditions on F, such that the estimator locates in a ball of radius $r(m)$ of F (when n is sufficiently large) in terms of some metrics such as L_1 distance and Kolmogorov distance, where $r(m)$ is a decreasing function of m. Specifically,

- **[Corollary of Theorem 6]** For general F, if the L_p distance is used, then $r(m) = \frac{C_1}{m^{-\frac{1}{2p}}}$ for the empirical cumulative distribution function, where C_1 is a positive constant that depends on p.

- **[Corollary of Theorem 10]** If F has a bounded density and the Kolmogorov distance is used, then $r(m) = \frac{C_2}{\sqrt{m}}$ for the empirical cumulative distribution function, where C_2 only depends on the maximal value of the density of F.

- **[Corollary of Theorem 13]** If F has a density whose derivative is absolutely continuous, and the truncated integrated L_2 distance is used, then $r(m) = \frac{C_3}{\sqrt[3]{m}}$ for the histogram estimator, where C_3 only depends on the density of F.

- **[Corollary of Theorem 19]** If F has a bounded monotone density and the L_1 distance is used, then $r(m) = \frac{C_4}{\sqrt[3]{m}}$ for the Grenander estimator, where C_4 only depends on the density of F.
- **[Corollary of Theorem 23]** If F has a U-shape density as specified in Section 3.4, then $r(m) = \frac{C_5}{\sqrt[3]{m}}$ for the Ucut introduced in Section 3.5, where C_5 only depends on the density of F.

3.1.4.2 The cutoff selection for GeneFishing

To model the CFRs generated by GeneFishing, we use the binomial mixture model with the U-shape constraint, under the regime where the binomial size m can be relatively large compared to the sample size n. With the theoretical understandings mentioned above, we propose a simple method Ucut to identify the cutoffs of the U shape and recover the underlying distribution based on the Grenander estimator. It has been shown that when $m = \Omega(n)$, the identified cutoffs converge at the rate $O(n^{-1/3})$. The L_1 distance between the recovered distribution and the true one decreases at the same rate. We also show that the estimated cutoff is larger than the true cutoff with high probability if the U-shape model holds. The performance of our method is demonstrated with varieties of simulation studies, a GTEX dataset used in (Liu et al., 2019) and a single cell dataset from Tabula Muris.

3.1.5 Outline

The rest of the chapter is organized as follows. Section 3.2 introduces the notation used throughout the chapter. To answer the first question mentioned in Section 3.1.3, Section 3.3 analyzes the estimation of the underlying distribution F in the binomial mixture model (3.1.2), under various conditions imposed on F. Equipped with these analysis tools, we cast our attention back to the GeneFishing method to answer the second question raised in Section 3.1.3. Section 3.4 introduces a model with U-shape constraint to model the output of GeneFishing. The cutoffs of our interest are also introduced in this model. In Section 3.5, we propose a non-parametric maximum likelihood estimation (NPMLE) method Ucut based on the Grenander estimator to estimate the underlying U-shape density as well as identifying the cutoffs. We also provide a theoretical analysis of the estimator in Section 3.5.3. Next, we apply Ucut to several synthetic datasets in Section 3.6 and real datasets in Section 3.7. Finally, all the detailed proofs of the theorems mentioned in previous sections are put in Appendix B.

3.2 Notation

Denote by F a probability distribution. Let $s \sim F$ and $m \cdot \hat{s} \sim Binomial(m, s)$ given s, where m is a positive integer; see Model (3.1.2) for details. If there is no confusion, we also use F to represent the associated cumulative distribution function (CDF), i.e, $F(x) = \mathbb{P}[s \leq x]$. By $F^{(m)}$ denote the binomial mixture CDF for $\hat{s}$, i.e., $F^{(m)}(x) = \mathbb{P}[\hat{s} \leq x]$.

Suppose there are n samples independently generated from F, i.e., $s_1, \ldots, s_n$. Correspondingly, we have $m \cdot \hat{s}_i | s_i \sim Binomial(m, s_i)$ independently. By F_n and $F_{n,m}$ denote the empirical CDF of s_i's and $\hat{s}_i$'s, respectively. Specifically,

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[s_i \leq x],$$

and

$$F_{n,m}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[\hat{s}_i \leq x].$$

If F has a density, define the Grenander estimator $\tilde{f}_n(x)$ ($\tilde{f}_{n,m}(x)$) for s_i's ($\hat{s}_i$'s) as the left derivative of the least concave majorant of F_n ($F_{n,m}$) evaluated at x (Grenander, 1956).

Define $\alpha_l(x; v) = \mathbb{P}[v \leq x]$, $\alpha_{mid}(x, y; v) = \mathbb{P}[x < v < y]$, $\alpha_r(y; v) = \mathbb{P}[v \geq y]$, where v can be s or $\hat{s}$. Define $N_l(x; \{v_i\}_{i=1}^n) := \#\{v_i \leq x, i = 1, \ldots, n\}$, $N_{mid}(x, y; \{v_i\}_{i=1}^n) := \#\{x < v_i \leq y, i = 1, \ldots, n\}$, $N_r(y; \{v_i\}_{i=1}^n) := \#\{v_i > y, i = 1, \ldots, n\}$, $N(x; \{v_i\}_{i=1}^n) = \#\{v_i = x, i = 1, \ldots, n\}$, where $\{v_i\}_{i=1}^n$ can be $\{s_i\}_{i=1}^n$ or $\{\hat{s}_i\}_{i=1}^n$.

For a density f, we use f_{max} and $f_{\min}$ to denote its maximal and minimal function values on the domain of f. We use $\mathbb{I}$ to denote the indicator function, and use x^+, x^- to denote the right and left limit of x respectively.

3.3 Estimation of F in Binomial Mixture Model

When m is sufficiently large, $\hat{s}$ behaves like s, which implies that we can directly estimate the underlying true F of the binomial mixture model (3.1.2). A natural question arises whether there exists a minimal binomial size m so that the identifiability issue mentioned in Section 3.1.2 is not a concern. We investigate general F, F with a density, F whose density has an absolutely continuous derivative, and F with a monotone density. We consider the empirical CDF estimator for the first two conditions, the histogram estimator for the third condition, and the Grenander estimator for the last condition. The investigations into the estimation of F under various conditions provide us with the analysis tools to design and analyze the cutoff method for GeneFishing.

3.3.1 General F

We begin with the estimation of F, based on $\{\hat{s}_i\}_{i=1}^n$, without additional conditions except that F is defined on $[0,1]$. Towards this end, the empirical CDF estimator might be the first method that comes into one's mind. It is easy to interpret using a diagram, and it exists no matter F corresponds to a density, a point mass function, or a mixture of both.

To measure the deviation of the empirical CDF from F, one might think of the L_p distance with $p \geq 1$. The L_p distance between two distributions F_1 and F_2 is defined as

$$\mathcal{L}_p(F_1, F_2) := \left(\int_{\mathbb{R}} |F_1(x) - F_2(x)|^p dx \right)^{\frac{1}{p}}.$$

The L_p distance has two special cases that are easily interpretable from a geometric perspective. First, when $p = 1$, it looks at the cumulative differences between the two CDFs; see the grey shaded area in Figure 3.1. The L_1 distance is known to be equivalent to the 1-Wasserstein ($\mathcal{W}_1$) distance on $\mathbb{R}$, which is also known as the Kantorovich-Monge-Rubinstein metric. Second, when $p = \infty$, it corresponds to the Kolmogorov-Smirnov (K-S) distance:

$$d_{KS}(F_1, F_2) := \sup_x |F_1(x) - F_2(x)|,$$

which measures the largest deviation between two CDFs; see the length of the red vertical shaded line in Figure 3.1. The K-S distance is a weaker notion of the total variation distance on $\mathbb{R}$ (total variation is often too strong to be useful).

In the sequel, we study $\mathcal{L}_p(F^{(m)}, F)$ for F supported on $[0,1]$. Theorem 6 states that without any conditions imposed on F, the L_p distance between $F^{(m)}$ and F is bounded by $\mathcal{O}(m^{-\frac{1}{2p}})$. One key to this theorem lies in the finite support of F, which enables the usage of the Fubini's theorem. Along with the Dvoretzky-Kiefer-Wolfowi (DKW) inequality (Massart, 1990), it implies that the L_p distance between $F_{n,m}$ and F is bounded by $\mathcal{O}(m^{-\frac{1}{2p}}) + \mathcal{O}(n^{-\frac{1}{2}})$. Particularly, we have $\mathcal{L}_1(F^{(m)}, F) = \mathcal{O}(m^{-\frac{1}{2}})$ and $\mathcal{L}_1(F_{n,m}, F) = \mathcal{O}(m^{-\frac{1}{2}}) + \mathcal{O}(n^{-\frac{1}{2}})$.

Theorem 6 (The L_p distance between $F^{(m)}/F_{n,m}$ and F) *For a general F on $[0,1]$, it follows that*

$$\left(\int_0^1 |F^{(m)}(x) - F(x)|^p dx \right)^{\frac{1}{p}} \leq \frac{C(p)}{m^{\frac{1}{2p}}},$$

where $C(p)$ is a positive constant that depends on p. It indicates that

$$\mathbb{E}\left(\int_0^1 |F_{n,m}(x) - F(x)|^p dx \right)^{\frac{1}{p}} \leq \frac{C(p)}{m^{\frac{1}{2p}}} + \frac{K}{\sqrt{n}},$$

where K is another universal positive constant.

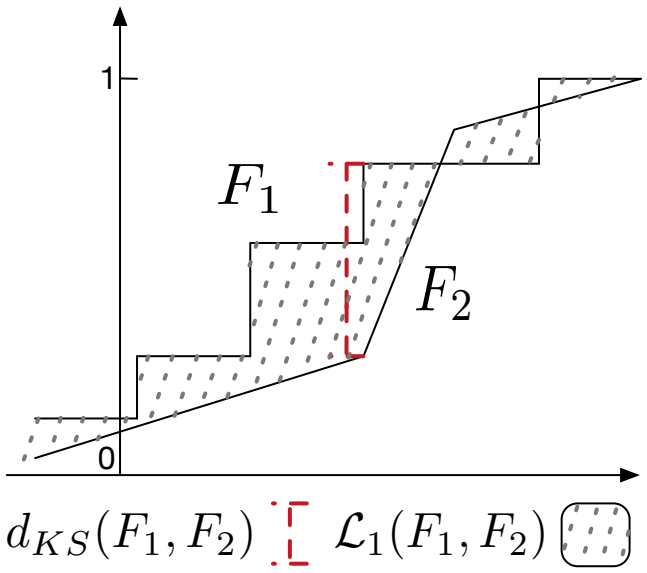


Figure 3.1: The length of the vertical shaded line in red represents the $d_{KS}(F_1, F2)$; the area of the grey shaded region represents $\mathcal{L}_1(F_1, F_2)$.

Proof By definition, it follows that

$$\begin{aligned} |F^{(m)}(x) - F(x)| &= |\mathbb{E}[\mathbb{I}(\hat{s} \leq x)] - \mathbb{E}[\mathbb{I}(s \leq x)]| \\ &= |\mathbb{E}[\mathbb{I}(\hat{s} \leq x < s)] - \mathbb{E}[\mathbb{I}(s \leq x < \hat{s})]|. \end{aligned}$$

Note that

$$\mathbb{E}[\mathbb{I}(\hat{s} \leq x < s)] = \mathbb{E}[\mathbb{E}[\mathbb{I}(\hat{s} \leq x < s)|s]] = \mathbb{E}[\mathbb{E}[\mathbb{I}(\hat{s}-s \leq x-s < 0)|s]] \leq \mathbb{E}[\exp\{-m(x-s)^2/2\}],$$

where $\hat{s} - s$ is bounded in $[-1, 1]$, and thus it is a sub-Gaussian random variable with the variance less or equal to 1 (Hoeffding, 1963). The same argument applies to $\mathbb{E}[\mathbb{I}(s \leq x < \hat{s})]$.

Therefore, we have

$$\begin{aligned}
\left(\int_0^1 |F^{(m)}(x) - F(x)|^p dx\right)^{\frac{1}{p}} &\leq 2\left(\int_0^1 \mathbb{E}[\exp\{-mp(x-s)^2/2\}]dx\right)^{\frac{1}{p}} \\
&= 2\left(\int_0^1\int_0^1 \exp\{-mp(x-s)^2/2\}dF(s)dx\right)^{\frac{1}{p}} \\
&\overset{(i)}{=} 2\left(\int_0^1\int_0^1 \exp\{-mp(x-s)^2/2\}dxdF(s)\right)^{\frac{1}{p}} \\
&\leq 2\left(\int_0^1 \frac{\sqrt{2\pi}}{\sqrt{mp}}dF(s)\right)^{\frac{1}{p}} \\
&= \frac{2(2\pi)^{\frac{1}{2p}}}{(mp)^{\frac{1}{2p}}},
\end{aligned}$$

where Equation (i) holds by the Fubini's theorem. Further, by noting that $F_{n,m}(x) - F(x) = F_{n,m}(x) - F^{(m)}(x) + F^{(m)}(x) - F(x)$ and using the DKW inequality, it follows that

$$\mathbb{E}\int_0^1 |F_{n,m}(x) - F(x)|dx \leq \frac{2(2\pi)^{\frac{1}{2p}}}{(mp)^{\frac{1}{2p}}} + \frac{\sqrt{2\pi}}{\sqrt{n}}.$$

■

Notwithstanding, Theorem 6 does not establish a useful bound for the K-S distance that corresponds to the L_∞ distance — there remains a non-negligible constant $\lim_{p\to\infty} \frac{2(2\pi)^{\frac{1}{2p}}}{(mp)^{\frac{1}{2p}}} = 2$ which does not depend on m. In fact, the K-S distance might evaluate the estimate of F from a too stringent perspective. Proposition 7 shows that no matter how small m is, there is an F with a point mass function and a point x_0 such that $|F^{(m)}(x_0) - F(x_0)|$ is larger than a constant that is independent of m. This result is attributed to the "bad" points with non-trivial masses like x_0. Such a "bad" point gives rise to a sharp jump in F, which $F^{(m)}$ cannot immediately catch up with due to the discretization nature of the binomial randomness. It leads to difficulty in recovering the underlying distribution F of the binomial mixture model.

On the other hand, the L_p distance with $p < \infty$ does not suffer from the issue of the K-S distance — it can be regarded as looking at an average of the absolute distance between $F^{(m)}$ and F when the support of F is finite. To be specific, even if there are "bad" points $x_1, x_2, \ldots$ such that $|F^{(m)}(x_i) - F(x_i)|$ has a non-trivial difference, $i = 1, 2, \ldots$, this difference will diminish outside a small neighbor of x_i of a width $\mathcal{O}(\frac{1}{m})$. Therefore, when taking the integral,

the averaging distance decreases as m grows. Furthermore, if F has a density, the issue of "bad" points no longer exists for the K-S distance either. In this case, the K-S distance is an appropriate choice to measure the difference between $F^{(m)}$ and F; see Section 3.3.2 for details.

Proposition 7 ($F^{(m)}$ can deviate in K-S far from F with a pmf) *There exists an F with a pmf, such that $\sup_x |F^{(m)}(x) - F(x)| \geq c > 0$, where c is a constant.*

Proof Let $F(x)$ be the delta function taking the mass at $\frac{1}{2} + \kappa$, where κ is an extremely small positive irrational number. Then by CLT, with probability about $1/2$, $\hat{s}$ is no larger than $\frac{\tilde{m}}{m}$, where $\tilde{m}$ is the largest integer such that $\frac{\tilde{m}}{m} < \frac{1}{2} + \kappa$. Take any x_0 in $(\frac{\tilde{m}}{m}, \frac{1}{2} + \kappa)$. It follows that $F(x_0) = 0$ but $F^{(m)}(x_0) \approx \frac{1}{2}$. It implies that $F^{(m)}(x_0) - F(x_0)$ is larger than a constant that is independent of m. ■

3.3.2 F with a density

In this section, we focus on F with a density so that the K-S distance can be employed. In addition, we stick to this metric partly because it is related to Grenander estimator for monotone density estimation (Grenander, 1956; Birge, 1989), which constitutes our method for the GeneFishing cutoff selection; see Section 3.3.4 and Section 3.5.

To bound the K-S distance between $F_{n,m}$ and F, i.e., $\sup_x |F_{n,m}(x) - F(x)|$, we just need to bound that between $F_{n,m} - F^{(m)}$ and that between $F^{(m)}$ and F by noticing that $F_{n,m} - F = F_{n,m} - F^{(m)} + F^{(m)} - F$. By the DKW inequality, we have a tight bound

$$\mathbb{P}[\sup_x |F_{n,m}(x) - F^{(m)}(x)| > t) \leq 2\exp\{-2nt^2\}, \quad \forall t > 0.$$

So it only remains to study the deviation of $F^{(m)}$ and F. In Proposition 8, we show that when F has a derivative bounded from both below and above, the K-S distance between $F^{(m)}$ and F is bounded by $\mathcal{O}(\frac{1}{m})$ from below and by $\mathcal{O}(\frac{1}{\sqrt{m}})$ from above.

Proposition 8 (Deviation of $F^{(m)}$ from F with a density) *Suppose f is a density function on $[0,1]$ with $0 < f_{\min} \leq f_{\max} < \infty$. Let $s \sim f$ and $m \cdot \hat{s} \sim Binomial(m, s)$. It follows that*

$$\frac{f_{\min}}{m+1} \leq \sup_x |F^{(m)}(x) - F(x)| \leq f_{\max} \cdot \frac{2\sqrt{2\pi}}{\sqrt{m}}.$$

Proof For the lower bound, we have

$$\mathbb{P}(\hat{s} \leq 0) - \mathbb{P}(s \leq 0) = \mathbb{P}(\hat{s} \leq 0) = \int_0^1 (1-u)^m f(u)du \geq f_{\min} \int_0^1 (1-u)^m du = \frac{f_{\min}}{m+1}.$$

For the upper bound, note that

$$\begin{aligned} F^{(m)}(x) - F(x) &= \mathbb{P}(\hat{s} \le x) - \mathbb{P}(s \le x) \\ &= \mathbb{E}(\mathbb{I}[\hat{s} \le x] - \mathbb{I}[s \le x]) \\ &= \mathbb{E}(\mathbb{I}[\hat{s} \le x < s] - \mathbb{I}[s \le x < \hat{s}]). \end{aligned}$$

We have

$$\begin{aligned} \mathbb{E}\mathbb{I}[\hat{s} \le x < s] &= \mathbb{P}(\hat{s} \le x < s) \\ &= \mathbb{P}(\hat{s} - s \le x - s < 0) \\ &= \mathbb{E}[\mathbb{P}(\hat{s} - s \le x - s < 0)|s] \\ &\le \mathbb{E}[\exp(-m(x-s)^2/2)] \\ &= \int_0^1 \exp(-m(x-u)^2/2) f(u) du \\ &\le f_{\max} \cdot \frac{\sqrt{2\pi}}{\sqrt{m}}, \end{aligned}$$

Similarly, we have $\mathbb{E}\mathbb{I}[s \le x < \hat{s}] \le f_{\max} \cdot \frac{\sqrt{2\pi}}{\sqrt{m}}$. So it follows that

$$\sup_x |\mathbb{P}(\hat{s} \le x) - \mathbb{P}(s \le x)| \le \sup_x \mathbb{E}\mathbb{I}[\hat{s} \le x < s] + \sup_x \mathbb{E}\mathbb{I}[s \le x < \hat{s}] \le f_{\max} \cdot \frac{2\sqrt{\pi}}{\sqrt{m}}.$$

■

Proposition 8 shows that the largest deviation of the binomial mixture CDF from the underlying CDF is at least the order $\mathcal{O}(m^{-1})$ and at most the order $\mathcal{O}(m^{-\frac{1}{2}})$. In fact, the condition that f is bounded can be relaxed to that f is L_p-integrable with $p > 1$ using the Hölder inequality, but the rate will be $\mathcal{O}(m^{-\frac{p}{2(p-1)}})$ correspondingly. Our result is the special case with $p = \infty$. Moreover, F with a density is a necessary condition for Proposition 8 — we have seen in Proposition 7 that if F has a point mass function, the deviation of $F^{(m)}$ from F cannot be controlled w.r.t m.

Proposition 9 shows that there exist two simple distributions that respectively attain the lower bound and the upper bound. However, Proposition 8 can be further improved: if the underlying density is assumed to be smooth, the lower bound is attained; see Proposition 12 in Section 3.3.3.

Proposition 9 (Tightness of Proposition 8) *The upper bound and the lower bound in Proposition 8 are tight. In other words, there exist an F_1 and F_2 such that $\sup_x |F_1^{(m)}(x) - F_1(x)| \le C_1 \cdot \frac{1}{m+1}$ and $\sup_x |F_2^{(m)}(x) - F_2(x)| \ge C_2 \cdot \frac{1}{\sqrt{m}}$, where C_1 and C_2 are two positive constants.*

Proof The lower bound of Proposition 8 can be attained by the uniform distribution. Specifically, if $f \equiv 1$, $\mathbb{P}(\hat{s} = k/m) = \frac{1}{m+1}$. So $|\mathbb{P}(\hat{s} \leq 0) - \mathbb{P}(s \leq 0)| = \frac{1}{m+1}$.

On the other hand, the upper bound can be attained by another simple distribution. Let

$$f(x) = 1.8 \cdot \mathbb{I}(x \in [0, 1/2]) + 0.2 \cdot \mathbb{I}(x \in (1/2, 1])$$

We can show that this density f leads to $|\mathbb{P}(\hat{s} \leq 1/2) - \mathbb{P}(s \leq 1/2)| \geq \frac{C}{\sqrt{m}}$, where C is a positive constant. It is a consequence of the central limit theorem for the binomial distribution. The detailed proof is delegated to Appendix B.1. ∎

Given Proposition 8, we can get the rate of of $\sup_x |F_{n,m}(x) - F(x)|$ along with the DKW inequality, which is $\mathcal{O}(n^{-1/2}) + \mathcal{O}(m^{-1/2})$ as shown in Theorem 10. By taking integral of $\mathbb{P}(\sup_x[F_{n,m}(x) - F(x)] > t)$ w.r.t t, we immediately have Corollary 11. It is easy to see that Theorem 10 can be generalized to a zero-inflation or one-inflation density.

Theorem 10 (The rate of K-S distance between $F_{n,m}$ and F with a density) *Suppose f is a density function on $[0, 1]$ with $f_{\max} < \infty$. The data is generated as Model (3.1.2). It follows that*

$$\mathbb{P}(\sup_x[F_{n,m}(x) - F(x)] > t) \leq \exp(-nt^2/2) + \mathbb{I}(f_{\max} \cdot \frac{2\sqrt{2\pi}}{\sqrt{m}} > t),$$

where $t \geq \sqrt{\frac{\log 2}{2n}}$. The two-side tail bound also holds as follow

$$\mathbb{P}(\sup_x |F_{n,m}(x) - F(x)| > t) \leq 2\exp(-nt^2/2) + \mathbb{I}(f_{\max} \cdot \frac{4\sqrt{2\pi}}{\sqrt{m}} > t),$$

where $t > 0$.

Proof Note that

$$\sup_x |F_{n,m}(x) - F(x)| \leq \sup_x |F_{n,m}(x) - F^{(m)}(x)| + \sup_x |F^{(m)}(x) - F(x)|.$$

The first term can be bounded by the original DKW inequality and the second term can be bound using the result of Proposition 8. Then we conclude the second result. The first result can be obtained in the same fashion. ∎

Corollary 11 *Under the same setup of Theorem 10, we have $\mathbb{E}\sup_x[F_{n,m}(x) - F(x)] \leq \frac{\sqrt{2\pi}}{\sqrt{n}} + \min\{1, \frac{f_{\max} \cdot 2\sqrt{2\pi}}{\sqrt{m}}\}$, and $\mathbb{E}\sup_x |F_{n,m}(x) - F(x)| \leq \frac{2\sqrt{2\pi}}{\sqrt{n}} + \min\{1, \frac{f_{\max} \cdot 4\sqrt{2\pi}}{\sqrt{m}}\}$.*

Proof By Theorem 10, it follows that

$$\begin{aligned}\mathbb{E}[\sup_x[F_{n,m}(x) - F(x)]] &= \int_0^1 \mathbb{P}(\sup_x[F_{n,m}(x) - F(x)] > t)dt \\ &\leq \int_0^1 [\exp(-nt^2/2) + \mathbb{I}(f_{\max} \cdot \frac{2\sqrt{2\pi}}{\sqrt{m}} > t)]dt \\ &\leq \frac{\sqrt{2\pi}}{\sqrt{n}} + \min\{1, \frac{f_{\max} \cdot 2\sqrt{2\pi}}{\sqrt{m}}\}.\end{aligned}$$

The two-side expectation can be proved in a similar manner. ■

3.3.3 F with A Smooth Density

In this section, we investigate the estimation of F with a smooth density. Under this condition, we first obtain a stronger result than Proposition 8 for $F^{(m)}$, based on a truncated K-S distance on the interval $[a, 1-a]$, where $0 < a < 1/2$. Proposition 12 shows that if the density of F is smooth, the truncated K-S distance decreases at $\mathcal{O}(\frac{1}{m})$. The proof is based on the fact the binomial distribution random variable $m \cdot \hat{s}_i$ behaves like a Gaussian random variable when the binomial probability s_i is bounded away from 0 and 1. When s_i is close to 0 and 1, the Gaussian approximation cannot be used since it has an unbounded variance $\frac{1}{ms_i(1-s_i)}$. The proof is deferred to Appendix B.2.

Proposition 12 (Deviation of $F^{(m)}$ from F with a smooth density) *Suppose f is a density function on $[0,1]$ with f' being absolutely continuous. Let $s \sim f$ and $m \cdot \hat{s} \sim Binomial(m, s)$. It follows for any $0 < a < 1/2$ that*

$$\sup_{x \in [a,1-a]} |F^{(m)}(x) - F(x)| \leq C \cdot \frac{1}{m},$$

where C is some constant that only depends on f and a.

Then, we investigate the histogram estimator, since it is the one of simplest nonparametric density estimators and it has a theoretical guarantee when the density is smooth. Let L be an integer and define bins

$$B_1 = [0, \frac{1}{L}), B_2 = [\frac{1}{L}, \frac{2}{L}), \ldots, B_L = [\frac{L-1}{L}, 1].$$

Define the bin-width $h = 1/L$, let Y_l be the number of observations in B_l, let $\hat{p}_l = Y_l/n$ and $p_l = \mathbb{P}(s_1 \in B_l)$. It is known that under certain smoothness conditions, the histogram

converges in the cubic root rate for the rooted mean squared error (MSE) (Wasserman, 2006). Next we study how the histogram behaves on the binomial mixture model. Denote

$$\begin{cases} \hat{f}_{n,m}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h}\mathbb{I}(\hat{s}_i \in B(x)) \\ \hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h}\mathbb{I}(s_i \in B(x)) \end{cases}$$

where h is the bandwidth, $B(x)$ denotes the the bin that x falls in. Theorem 13 shows that the histogram estimator based on $\hat{s}_i$'s has the same convergence rate in terms of the MSE metrics as the histogram estimator based on s_i's if $m = \Omega(n^{2/3})$ and $h = \mathcal{O}(n^{-\frac{1}{3}})$. This rate might not be improved even if f has higher order continuous derivatives since $\mathbb{E}|\hat{f}_{n,m}(x) - \mathbb{E}\hat{f}_n(x)|$ is bounded by $\mathcal{O}(\frac{1}{\sqrt{m}} + h + \frac{1}{mh})$, which dominates $\mathbb{E}|\hat{f}_n(x) - F(x)| = \mathcal{O}(h^\nu)$ when f has a ν-order continuous derivative. The proof of Theorem 13 is delegated to Appendix B.3.

Theorem 13 (Upper bound of the histogram risk for binomial mixture model) *Let $R(a,b) = \int_a^b \mathbb{E}(f(x) - \hat{f}_{n,m}(x))^2 dx$ be the integrated risk on the interval $[a,b]$. Assume that f' is absolutely continuous. It follows that*

$$R(a, 1-a) \leq C_1 \cdot (h^2 + \frac{1}{m} + \frac{1}{m^2h^2} + \frac{1}{nh}), \forall 0 < a < \frac{1}{2},$$

Furthermore, if $m \geq C_2 \cdot n^{\frac{2}{3}}$, $h = C_3 \cdot n^{-\frac{1}{3}}$, we have

$$R(a, 1-a) \leq C_4 \cdot n^{-\frac{2}{3}}, \forall 0 < a < \frac{1}{2}.$$

Here C_1, C_2 and C_4 are positive constants that only depend on a and f, $C_3 > 0$ only depends on f.

Finally, we conclude this section with a study on F whose density is discretized into a point mass function of K non-zero masses. In contrast to the existing results in Teicher (1963), we allow K to be as large as $\sqrt{n}$, and study the finite-sample rate of the histogram estimator. Let $p(x)$ be a point mass function such that

$$p(x) = \sum_{k=1}^{K} \alpha(k)\mathbb{I}(x = x_k), \tag{3.3.1}$$

where $\alpha(k) \geq 0$ and $\sum_{k=1}^{K} \alpha(k) = 1$, $s_k = \frac{(k-1)+1/2}{K}$. Denote by I_k the interval centered at x_k and of length $1/K$. We can make $(\alpha(1), \ldots, \alpha(K))$ "smooth" by letting $\alpha(k) = \int_{I_k} f(t)dt$, where f is a smooth function. Denote

$$\begin{cases} \hat{\alpha}_{n,m}(k) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(\hat{s}_i \in I_k) \\ \hat{\alpha}_n(k) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(s_i \in I_k) \end{cases}$$

Then the MSE can be defined as $R(\hat{\alpha}, \alpha) = \frac{1}{K}\sum_{k=1}^{K}(\hat{\alpha}(k) - \alpha(k))^2$. It is known that $R(\hat{\alpha}_n, \alpha) = \mathcal{O}(\frac{1}{n})$. Theorem 14 shows that the same rate can be achieved by $\hat{\alpha}_{n,m}$ when $m = \Omega(\sqrt{n}\max\{\sqrt{n}, K\})$. The proof is deferred to Appendix B.4.

Theorem 14 (Upper bound of the histogram risk for finite binomial mixture) *Let $\alpha(k) = \int_{I_k} f(t)dt$, where f' is absolutely continuous and $\int (f')^2 dx < \infty$. Let $R(a,b) = \frac{1}{\sum_{k=1}^{K}\mathbb{I}(aK \le k \le bK)}\sum_{aK\le k\le bK}(\hat{\alpha}_{n,m}(k) - \alpha(k))^2$ be the risk on the interval $[a,b]$. It follows that*

$$R(a, 1-a) \le C_1 \cdot (\frac{1}{n} + \frac{1}{m} + \frac{K^2}{m^2}), \forall 0 < a < \frac{1}{2}.$$

Furthermore, if $m \ge C_2 \cdot \sqrt{n}\max\{\sqrt{n}, K\}$, then

$$R(a, 1-a) \le C_3 \cdot \frac{1}{n}, \forall 0 < a < \frac{1}{2}.$$

Here C_1, C_2, C_3 are positive constants that only depend on a and f.

3.3.4 F with A Monotone Density

Next, we shift our attention to F with a monotone density f. It is motivated by the U shape in the histograms of the GeneFishing output, where we can decompose the U shape into a decreasing part, a flat part, and an increasing part. To estimate f, a natural solution is the Grenander estimator (Grenander, 1956; Jankowski and Wellner, 2009). Specifically, we construct the least concave majorant of the empirical CDF of F. And its left derivative is the desired estimator.

3.3.4.1 Review of Grenander Estimator

Monotone density models are often used in survival analysis and reliability analysis in economics—see Huang and Zhang (1994) and Huang and Wellner (1995). We can apply maximum likelihood for the monotone density estimation. Suppose that $X_1, \ldots, X_n$ is a random sample from a density f on $[0, \infty)$ that is known to be nonincreasing; the maximum likelihood estimator $\tilde{f}_n$ is defined as the nonincreasing density that maximizes the log-likelihood $\ell(f) = \sum_{i=1}^{n}\log f(X_i)$. Grenander (1956) first showed that this optimization problem has a unique solution under the monotone assumption — so the estimator is also called the Grenander estimator. The Grenander estimator is given explicitly by the left derivative of the least concave majorant of the empirical distribution function F_n. The least concave majorant of F_n is defined as the smallest concave function $\tilde{F}_n$ with $\tilde{F}_n \ge F_n$ for every x. Because $\tilde{F}_n$ is concave, its derivative is nonincreasing.

Marshall and Proschan (1965) showed that Grenander estimator is consistent when f is decreasing, as stated in Theorem 15.

Theorem 15 (Marshall and Proschan (1965)) *Suppose that $X_1, \ldots, X_n$ are i.i.d random variables with a decreasing density f on $[0, \infty)$. Then the Grenander estimator $\tilde{f}_n$ is uniformly consistent on closed intervals bounded away from zero: that is, for each $c > 0$, we have*

$$\sup_{c \leq x < \infty} |\tilde{f}_n(x) - f(x)| \to 0 \text{ a.s.}$$

The inconsistency of the Grenander estimator at 0, when $f(0)$ is bounded, was first pointed out by Woodroofe and Sun (1993). Balabdaoui et al. (2011) later extended this result to other situations, where they consider different behavior near zero of the true density. Theorem 16 explicitly characterizes the behavior of $\tilde{f}_n$ at zero.

Theorem 16 (Woodroofe and Sun (1993)) *Suppose that f is a decreasing density on $[0, \infty)$ with $0 < f(0) < \infty$, and let $N(t)$ denote a rate 1 Poisson process. Then*

$$\frac{\tilde{f}_n(0)}{f(0)} \xrightarrow{d} \sup_{t>0} \frac{N(t)}{t} \stackrel{d}{=} \frac{1}{U},$$

where U is a uniform random variable on the unit interval.

Birge (1989) proved that Grenander estimator has a cubic root convergence rate in the sense of L_1 norm, as in Theorem 17. Van der Vaart and Van der Laan (2003) pointed out that the rate of convergence of the Grenander estimator is slower than that of the monotone kernel density estimator when the underlying function is smooth enough.

Theorem 17 (Birge (1989)) *Suppose f is a decreasing density on $[0, \infty)$ with $0 < f(0) < \infty$. it follows that*

$$\mathbb{E}_f \int_0^1 |\tilde{f}_n(x) - f(x)| dx \leq C \cdot n^{-\frac{1}{3}},$$

where C is a constant that only depends on f.

Rao (1970) first obtained the limiting distribution of the Grenander estimator at a point. He has proved that $\sqrt[3]{n}(\tilde{f}_n(t_0) - f(t_0))$ converges to the location of the maximum of the process $\{B(x) - x^2, x \in \mathbb{R}\}$, where $f'(t_0) < 0$ and $B(x)$ is the standard two-sided Brownian motion on $\mathbb{R}$ such that $B(0) = 0$; see Rao (1970). Wang (1992) extends this result to the flat region and a higher order derivative, as stated in Theorem 18.

Theorem 18 (Wang (1992)) *Suppose f is a decreasing density on $[0, 1]$ and is smooth at $t_0 \in (0, 1)$. It follows that*

(A) If f is flat in a neighborhood of t_0. Let $[a, b]$ be the flat part containing t_0. Then,

$$\sqrt{n}(\tilde{f}_n(t_0) - f(t_0)) \xrightarrow{d} \hat{S}_{a,b}(t_0),$$

where $\hat{S}_{a,b}(t)$ is the slope at $F(t)$ of the least concave majorant in $[F(a), F(b)]$ of a standard Brownian Bridge in $[0, 1]$.

(B) If $f(t) - f(t_0) \sim f^{(k)}(t_0)(t - t_0)^k$ near t_0 for some k and $f^{(k)}(t_0) < 0$. Then,

$$n^{\frac{k}{2k+1}}[\frac{f^k(t_0)|f^{(k)}(t_0)|}{(k+1)!}]^{-\frac{1}{2k+1}}(\tilde{f}_n(t_0) - f(t_0)) \xrightarrow{d} V_k(0),$$

where $V_k(t)$ is the slope at t of the least concave majorant of the process $\{B(t)-|t|^{k+1}, t \in (-\infty, \infty)\}$, and $B(t)$ is a standard two-sided Brownian motion on $\mathbb{R}$ with $B(0) = 0$.

3.3.4.2 The Grenander Density Estimator for the Binomial Mixture Model

In this section, we establish similar results introduced in Section 3.3.4.1 for $\tilde{f}_{n,m}$ that is the Grenander estimator based on $\hat{s}_i$'s instead of s_i's. Theorem 19 states $\tilde{f}_{n,m}$ achieves the convergence rate of $\mathcal{O}(n^{-\frac{1}{3}})$ if $m = \Omega(n)$.

Theorem 19 (L_1 convergence of $\tilde{f}_{m,n}$) *Suppose f is a decreasing density on $[0, 1]$ with $f_{\max} < \infty$. It follows that*

$$\mathbb{E}_f \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \leq C_1 \cdot n^{-\frac{1}{3}} + C_2 \cdot m^{-\frac{1}{3}}.$$

Furthermore, if $m \geq C_3 \cdot n$, we have

$$\mathbb{E}_f \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \leq C_4 \cdot n^{-\frac{1}{3}}.$$

Here C_1, C_2, C_3, C_4 are positive constants that only depend on f.

Theorem 19 follows by Corollary 11 and the proof of Theorem 17. The details can be found in Appendix B.5.

Next, we study the local asymptotics of $\tilde{f}_{n,m}$. For the binomial mixture model, we yield a similar result for $\tilde{f}_{n,m}$ as Theorem 18 when m grows faster than n, as shown in Theorem 20.

Theorem 20 (Local Asymptotics of $\tilde{f}_{m,n}$) *Suppose f is a decreasing density on $[0, 1]$ and is smooth at $t_0 \in (0, 1)$. If $f_{\max} < \infty$ and $\frac{m}{n} \to \infty$ as $n \to \infty$, we have*

(A) If f is flat in a neighborhood of t_0. Let $[a, b]$ be the flat part containing t_0. Then,

$$\sqrt{n}(\tilde{f}_{m,n}(t_0) - f(t_0)) \xrightarrow{d} \hat{S}_{a,b}(t_0),$$

where $\hat{S}_{a,b}(t)$ is the slope at $F(t)$ of the least concave majorant in $[F(a), F(b)]$ of a standard Brownian Bridge in $[0, 1]$.

(B) If $f(t) - f(t_0) \sim f^{(k)}(t_0)(t - t_0)^k$ near t_0 for some k and $f^{(k)}(t_0) < 0$. Then,

$$n^{\frac{k}{2k+1}}[\frac{f^k(t_0)|f^{(k)}(t_0)|}{(k+1)!}]^{-\frac{1}{2k+1}}(\tilde{f}_{m,n}(t_0) - f(t_0)) \xrightarrow{d} V_k(0),$$

where $V_k(t)$ is the slope at t of the least concave majorant of the process $\{B(t)-|t|^{k+1}, t \in (-\infty, \infty)\}$, and $B(t)$ is a standard two-sided Brownian motion on $\mathbb{R}$ with $B(0) = 0$.

The proof of Theorem 20 relies on Proposition 8 and the Komlós-Major-Tusnády (KMT) approximation (Komlós, Major, and Tusnády, 1975). Given these two results, we can show that if f is upper bounded, there exists a sequence of Brownian bridges $\{B_n(x), 0 \le x \le 1\}$ such that

$$\mathbb{P}\left\{\sup_{0\le x\le 1} |\sqrt{n}(F_{n,m}(x) - F(x)) - B_n(F(x))| > \frac{\tilde{a}\sqrt{n}}{\sqrt{m}} + \frac{a\log n}{\sqrt{n}} + t\right\} \le be^{-c\sqrt{n}t},$$

where $\tilde{a} > 0$ only depends on f and a, b, c are universal positive constants. Together with the proof of Theorem 18, Theorem 20 follows. The details are deferred to Appendix B.6.

Remark 1 *Theorem 19 and Theorem 20 can be improved when f is smooth in the sense that we only need $m = \Omega(\sqrt{n})$ and m grows faster than $\sqrt{n}$, respectively. This results from the fact that $\sup_{0\le x\le 1} |F^{(m)}(x) - F(x)| = \mathcal{O}(m^{-\frac{1}{2}})$ when f is bounded (Proposition 8) can be improved to $\sup_{a\le x\le 1-a} |F^{(m)}(x) - F(x)| = \mathcal{O}(m^{-1})$ when f has an absolutely continuous derivative (Proposition 12).*

Finally, we conclude this section with a discussion on the histogram estimator and the Grenander estimator (for a density). Both of them are bin estimators but differ in the choice of the bin width. One can pick the bin width for the histogram to attain optimal convergence rates (Wasserman, 2006). On the other hand, the bin widths of the Grenander estimator are chosen completely automatically by the estimator and are naturally locally adaptive (Birge, 1989). The consequence is that the Grenander estimator can guarantee monotonicity, but the histogram estimator cannot. If the underlying model is monotone, the Grenander estimator has a better convergence rate than the histogram estimator. Notably, the convergence theory of the histogram estimator cannot be established unless the density is smooth. In contrast,

that of the Grenander estimator only requires the density is monotone and L_p integrable ($p > 2$) (Birge, 1989).

In our setup, we show that both the Grenander estimator with $m = \Omega(n)$ and the histogram estimator with $m = \Omega(n^{2/3})$, based on $\{\hat{s}_i\}_{i=1}^n$, have the same rate at $\mathcal{O}(n^{-1/3})$ in L_1 distance (the L_2 convergence of the histogram can imply the L_1 convergence). It seems that the histogram estimator is more favorable than the other because it requires a smaller binomial size. Nonetheless, we mention that the conditions for the convergence of the two estimators are different. The Grenander estimator requires a bounded monotone density, while the histogram requires a smooth density. If the density is monotone and has an absolutely continuous derivative, the Grenander estimator requires $m = \Omega(n^{1/2})$ less than the histogram estimator, which is illustrated in Remark 1.

3.4 The U-shape Model

Now we have sufficient insights into the estimations of F under various conditions in the binomial mixture model (3.1.2). We are ready to cast our attention back to the cutoff selection problem for the GeneFishing method, i.e., distinguishing the relevant genes from the irrelevant ones. To answer this question, we leverage the observation that the histogram appears to have a U shape for the number of times a gene is fished out in Figure 1.2. We decompose the density or the pmf of F into three parts: the first part decreases, the second part remains flat, and the last part increases. The first part is assumed to be purely related to the irrelevant genes; the second part is associated with the mixture of the irrelevant and the relevant genes; the last part is purely corresponding to the relevant genes. Denote by c_l and c_r the transition points from the first part to the second part, and the second part to the third part, respectively. Then the question is reduced to identifying c_r and getting an upper confidence bound on c_r. In the sequel, we formally write this assumption when F is associated with a continuous random variable. The corresponding mathematical formulations for the pmf are similar, so we omit them here.

Let f be the derivative of F, i.e., the probability density function. We assume f consists of three parts:

$$f(x) = \begin{cases} f_l(x) = \alpha_l \cdot g_l(x), & \text{if } x \in [0, c_l] \\ \frac{\alpha_{mid}}{c_r - c_l}, & \text{if } x \in (c_l, c_r] \\ f_r(x) = \alpha_r \cdot g_r(x), & \text{if } x \in (c_r, 1] \end{cases}, \tag{3.4.1}$$

where $0 < c_l < c_r < 1$, g_l is a decreasing function, g_r is an increasing function such that $\int_0^{c_l} g_l(x)dx = 1$, $\int_1^{c_r} g_r(x)dx = 1$, and $\alpha_l + \alpha_r + \alpha_{mid} = 1$ with $\alpha_l, \alpha_r, \alpha_{mid} > 0$. For the U-shaped constraint, we also need

$$\min\{f_l(c_l^-), f_r(c_r^+)\} \geq \frac{\alpha_{mid}}{c_r - c_l}.$$

The shape constraint (3.4.1) is determined by six parameters $\{\alpha_l, \alpha_r, c_l, c_r, g_l, g_r\}$, but they are not identifiable. Below is an example of such unidentifiability.

Example 1 (Identifiability Issue for (3.4.1)**)**

$$\begin{array}{ll} \tilde{\alpha}_l = \alpha_l + \frac{\alpha_{mid}}{c_r - c_l} \cdot \tau; & \tilde{\alpha}_r = \alpha_r \\ \tilde{c}_l = c_l + \tau; & \tilde{c}_r = c_r \\ \tilde{g}_l = \begin{cases} g_l \cdot \alpha_l / \tilde{\alpha}_l, & \text{if } x \in [0, c_l] \\ \frac{\alpha_{mid}}{(c_r - c_l)\cdot \tilde{\alpha}_l}, & \text{if } \in (c_l, c_l + \tau] \end{cases} ; & \tilde{g}_r = g_r \end{array}$$

The parameters $\{\tilde{\alpha}_l, \tilde{\alpha}_r, \tilde{c}_l, \tilde{c}_r, \tilde{g}_l, \tilde{g}_r\}$ *yield the same model as* $\{\alpha_l, \alpha_r, c_l, c_r, g_l, g_r\}$ *if* $\tau < c_r - c_l$.

The identifiability issue results from the vague transitions from one part to the next adjacent part in Model (3.4.1). To tackle it, we need to introduce some assumptions to sharpen the transitions. For example, if f is smooth, then a sharp transition means the first derivative of f, i.e., the slope, significantly changes at this point. In general, we do not impose the smoothness on f and use the finite difference as the surrogate of the slope. To be specific, suppose there exist $\delta_l, \delta_r > 0$ and neighborhoods around c_l and c_r with sizes τ_l and τ_r such that

$$\begin{array}{ll} f_l(x) \geq \frac{\alpha_{mid}}{c_r - c_l} + \delta_l \cdot (c_l - x), & \text{if } x \in [c_l - \tau_l, c_l) \\ f_r(x) \geq \frac{\alpha_{mid}}{c_r - c_l} + \delta_r \cdot (x - c_r), & \text{if } x \in (c_r, c_r + \tau_r]. \end{array}$$

For the sake of convenience, we consider a stronger condition that drop off the factors $c_l - x$ and $x - c_r$, which is called **Assumption 2**. It indicates that the density jumps at the transition points c_l and c_r.

Assumption 2 *There are two positive parameters* δ_l *and* δ_r *such that*

$$\begin{array}{l} f_l(c_l^-) \geq \frac{\alpha_{mid}}{c_r - c_l} + \delta_l \\ f_r(c_r^+) \geq \frac{\alpha_{mid}}{c_r - c_l} + \delta_r. \end{array}$$

Together, we call the constraint (3.4.1) with Assumption 2 the gapped U-shape constraint. And we refer to the Binomial mixture with the gapped U-shape constraint as the **BMU** model.

3.5 Method

Let $c_l^{(0)}$ and $c_r^{(0)}$ be the underlying ground truth of the two cutoffs in BMU. Our goal is to identify the cutoff that separates the relevant genes and the irrelevant genes in the

GeneFishing method. Specifically, we want to find an estimator $\hat{c}_r$ for $c_r^{(0)}$ and study the behavior of $\mathbb{P}[\hat{c}_r \geq c_r^{(0)}]$.

Since we are working on $\hat{s}_i$'s rather than on s_i's, we denote $\alpha_l(x) = \mathbb{P}[\hat{s} \leq x]$ and $N_l(x) = N_l(x; \{\hat{s}_i\}_{i=1}^n)$ for simplicity. Similarly, we can get simplified notation $\alpha_{mid}(x, y)$, $\alpha_r(y)$, $N_{mid}(x, y)$, $N_r(y)$, $N(x)$; see Section 3.2 for details. In the rest of the chapter, we sometimes use α_l, α_r, α_{mid} for $\alpha_l(c_l)$, $\alpha_r(c_r)$ and $\alpha_{mid}(c_l, c_r)$ respectively if no confusion arises.

3.5.1 The Non-parametric Maximum Likelihood Estimation

To estimate the parameters in BMU, we consider the non-parametric maximum likelihood estimation (NPMLE). We first solve the problem given c_l and c_r, then searching for optimal c_l and c_r using grid searching. The NPMLE problem is:

$$\begin{aligned} H_{full}(c_l, c_r) := \max \quad & \sum_{\hat{s}_i \leq c_l} \log g_l(\hat{s}_i) + \sum_{\hat{s}_i > c_r} \log g_r(\hat{s}_i) \qquad (3.5.1) \\ & + N_l(c_l) \log \alpha_l + N_{mid}(c_l, c_r) \log \frac{\alpha_{mid}}{c_r - c_l} + N_r(c_r) \log \alpha_r \\ s.t. \quad & \int_0^{c_l} g_l = 1, \int_{c_r}^1 g_r = 1, g_l \text{ decreasing }, g_r \text{ increasing} \\ & \alpha_l, \alpha_r, \alpha_{mid} > 0, \alpha_l + \alpha_r + \alpha_{mid} = 1 \\ & \left.\begin{array}{l} \alpha_l g_l(c_l^-) \geq \frac{\alpha_{mid}}{c_r - c_l} + d_l \\ \alpha_r g_r(c_r^+) \geq \frac{\alpha_{mid}}{c_r - c_l} + d_r \end{array}\right\} \qquad (3.5.2) \end{aligned}$$

Here d_l and d_r are two parameters to tune, and we call the inequalities (3.5.2) the **change-point-gap** constraint. Given c_l and c_r, the variables to optimize over are

$$S := \{\alpha_l, \alpha_r, g_l(\hat{s}_1), \ldots, g_l(\hat{s}_{i_l}), g_r(\hat{s}_{i_r}), \ldots, g_r(\hat{s}_n)\},$$

where $i_l := \max_i \cdot \mathbb{I}(\hat{s}_i < c_l)$, $i_r := \min_i \cdot \mathbb{I}(\hat{s}_i \geq c_r)$. Since $\log x$ is continuous and concave w.r.t x, and the feasible set is convex (it is easy to check that $\{(x, y, z) : xy \geq z; x, y, z \geq 0\}$ is a convex set), the problem (3.5.1) is one of convex optimizations with a unique optimizer.

There are mainly two difficulties for the optimization problem (3.5.1). First, the change-point-gap constraint (3.5.2) complicates the monotone density estimation. Furthermore, it is not easy to optimize over α_l, α_r and g_l, g_r simultaneously.

3.5.2 Ucut: A simplified Estimator

Fortunately, we have the below observations that motivate us to think of a simplified optimization problem.

- Note that α_l and α_r are the population masses for $x \le c_l$ and $x \ge c_r$, which can be well estimated by the empirical masses $\hat{\alpha}_l(c_l) = N_l(c_l)/n$, $\hat{\alpha}_{mid}(c_l, c_r) = N_{mid}(c_l, c_r)/n$ and $\hat{\alpha}_r(c_r) = N_r(c_r)/n$.
- If BMU is true with $\delta_l \ge d_l$ and $\delta_r \ge d_r$, and the solution to the optimization (3.5.1) without (3.5.2) at $c_l = c_l^{(0)}$, $c_r = c_r^{(0)}$ is good enough, then the change-point-gap constraint (3.5.2) is satisfied with high probability.
- From Figure 1.2, we can see that the flat region is wide. We can easily pick an interior point within the flat region.

Inspired by these observations, we replace the population masses with the empirical masses and drop off the change-point-gap constraint. We obtain the simplified objective function as follows:

$$\begin{aligned} H_{simplified}(c_l, c_r) := \max \quad & \sum_{\hat{s}_i \le c_l} \log g_l(\hat{s}_i) + \sum_{\hat{s}_i > c_r} \log g_r(\hat{s}_i) \\ & + N_l(c_l) \log \hat{\alpha}_l(c_l) + N_{mid}(c_l, c_r) \log \frac{\hat{\alpha}_{mid}(c_l, c_r)}{c_r - c_l} + N_r(c_r) \log \hat{\alpha}_r(c_r) \\ s.t. \quad & \int_0^{c_l} g_l = 1, \int_{c_r}^1 g_r = 1, g_l \text{ decreasing }, g_r \text{ increasing} \end{aligned} \tag{3.5.3}$$

where

$$\begin{aligned} \hat{\alpha}_l(c_l) &= N_l(c_l)/n = \frac{\#\{i|\hat{s}_i \le c_l\}}{n} \\ \hat{\alpha}_{mid}(c_l, c_r) &= N_{mid}(c_l, c_r)/n = \frac{\#\{i|c_l < \hat{s}_i \le c_r\}}{n} \\ \hat{\alpha}_1(c_r) &= N_r(c_r)/n = \frac{\#\{i|\hat{s}_i > c_r\}}{n}. \end{aligned}$$

The problem (3.5.3) is reduced to two monotone density estimations, which the Grenander estimator can solve. As we point out in the above observations, we can easily identify an interior point μ in the flat region. We fit an Grenander estimator for the decreasing g_l on $[0, \mu]$ and an Grenander estimator for the increasing g_r on $(\mu, 1]$. There are three advantages of using the interior point μ. First, it significantly reduces the computational cost by estimating the two Grenander estimators just once, regardless of the choices of c_l and c_r. Second, it bypasses the boundary issue of the Grenander estimators since we are mainly concerned with the behaviors of the estimators at the points $c_l < \mu$ and $c_r > \mu$. Moreover, the usage of μ disentangles the mutual influences of the left decreasing part and the right increasing part; thus, it makes the analysis of the estimators simple.

Once we fit the Grenander estimator, we check whether the change-point-gap constraint (3.5.2) holds for different pairs of c_l and c_r. Finally, we pick the feasible pair with the maximal likelihood. We call this algorithm **Ucut** (U-shape cutoff), which is summarized in Algorithm 5.

Algorithm 5 Ucut: estimation of the BMU model by grid-searching the optimal cutoff pair.

Require: Data: $\{\hat{s}_1, \ldots, \hat{s}_n\}$;
The density gaps: d_l, d_r;
The interior point μ of the flat region;
The searching interval: $[0, c_l^{(\max)}]$, and $(c_r^{(\min)}, 1]$, where $c_l^{(\max)} < \mu$ and $c_r^{(\min)} > \mu$;
The unit for grid searching: γ.

Initiate $c_l^* = \text{NULL}$, $c_r^* = \text{NULL}$; $\ell(c_l^*, c_r^*) = -\infty$.
Estimate $\hat{\alpha}_l(\mu) = N_l(\mu)/n$.
Fit the Grenander estimator on $[0, \mu]$ to get $\tilde{g}_l$ and on $(\mu, 1]$ to get $\tilde{g}_r$.
for $c_l \in \{0, \gamma, 2\gamma, \ldots, c_l^{(\max)}\}$ **do**
  for $c_r \in \{c_r^{(\min)}, c_r^{(\min)} + \gamma, c_r^{(\min)} + 2\gamma, \ldots, 1\}$ **do**
    Estimate $\hat{\alpha}_{mid}(c_l, \mu)$, and $\hat{\alpha}_{mid}(\mu, c_r)$.
    Let $\tilde{d}_l = \frac{\hat{\alpha}_{mid}(c_l,\mu)}{\hat{\alpha}_l(\mu)\cdot(\mu - c_l)} + \frac{d_l}{\hat{\alpha}_l(\mu)}$, $\tilde{d}_r = \frac{\hat{\alpha}_{mid}(\mu,c_r)}{(1-\hat{\alpha}_l(\mu))\cdot(c_r-\mu)} + \frac{d_r}{1-\hat{\alpha}_l(\mu)}$.
    Let $\ell(c_l, c_r)$ be the corresponding $H_{simplified}(c_l, c_r)$ defined in the problem (3.5.3).
    Let $flag = \mathbb{I}[\tilde{g}_l(c_l) \geq \tilde{d}_l \text{ and } \tilde{g}_r(c_r) \geq \tilde{d}_r]$.
    if $flag$ and $\ell(c_l, c_r) > \ell(c_l^*, c_r^*)$ **then**
      $(c_l^*, c_r^*) \leftarrow (c_l, c_r)$.
    end if
  end for
end for
if $\ell(c_l^*, c_r^*) > -\infty$ **then**
  Return: c_l^*, c_r^*, $\tilde{g}_l, \tilde{g}_r$, $\hat{\alpha}_l(\mu)$, $\ell(c_l^*, c_r^*)$.
else
  Return: $False$.
end if

3.5.3 Analysis of the Simplified Estimator

For Algorithm 5, the question arises whether $(c_l^{(0)}, c_r^{(0)})$ is a feasible pair for the change-point-gap constraint (3.5.2). Theorem 21 answers this question by claiming that there exist c_l in a small neighborhood of $c_l^{(0)}$ and c_r in a small neighborhood of $c_r^{(0)}$ such that $\tilde{g}_l(c_l) > \tilde{d}_l$ and $\tilde{g}_r(c_r) > \tilde{d}_r$ for appropriate choices of d_l and d_r. This implies that we can safely set aside

the constraint (3.5.2) when solving the problem (3.5.3). The proof of Theorem 21 is deferred to Appendix B.7.

Theorem 21 (Feasibility of Gap Constraint for $(c_l^{(0)}, c_r^{(0)})$) *Suppose f is a distribution satisfying the constraint* (3.4.1) *and Assumption 2, with $\alpha_{mid}(c_l^{(0)}, c_r^{(0)}) > 0$, $f_{\max} < \infty$. If $m \geq C_1 \cdot \max\{N_l(c_l^{(0)}), N_r(c_r^{(0)})\}$, and $d_l < \delta_l$, $d_r < \delta_r$, there exist c_l and c_r such that*

$$c_l \leq c_l^{(0)} \text{ with } |c_l - c_l^{(0)}| \leq C_2 \cdot N_l(c_l^{(0)})^{-\frac{1}{3}}$$

and

$$c_r \geq c_r^{(0)} \text{ with } |c_r - c_r^{(0)}| \leq C_3 \cdot N_r(c_r^{(0)})^{-\frac{1}{3}}$$

such that $\tilde{g}_l$, $\tilde{g}_r$, $\tilde{d}_l$ and $\tilde{d}_r$ produced by Algorithm 5 satisfy $\tilde{g}_l(c_l) > \tilde{d}_l$ and $\tilde{g}_r(c_r) > \tilde{d}_r$, provided the input $\mu \in (c_l^{(0)}, c_r^{(0)})$. Furthermore, the resulting density estimator $\tilde{f}_{m,n}$ satisfies

$$\int_0^1 |\tilde{f}_{m,n} - f(x)|dx \leq C_4 \cdot \left\{ N_l(c_l^{(0)})^{-\frac{1}{3}} + N_r(c_r^{(0)})^{-\frac{1}{3}} \right\}.$$

Here C_1, C_2, C_3, C_4 are positive constants that only depend on f.

Besides knowing there are some feasible points near $c_l^{(0)}$ and $c_r^{(0)}$, we want to have a clear sense of the optimal cutoff pair produced by Algorithm 5. Theorem 22 says that the optimal cutoff for the left (right) part is smaller (larger) than $c_l^{(0)}$ ($c_r^{(0)}$) with high probability.

Theorem 22 (Tail Bounds of Identified Cutoffs) *Suppose $(\hat{c}_l, \hat{c}_r)$ is the identified optimal cutoff pair produced by Algorithm 5, provided an input $\mu \in (c_l^{(0)}, c_r^{(0)})$. Under the same assumptions as Theorem 21, particularly $n \to \infty$, $m/\max\{N_l(c_l^{(0)}), N_r(c_r^{(0)})\} \to \infty$,*

$$\mathbb{P}[\hat{c}_l > c_l^{(0)}] \leq \mathbb{P}[\hat{S}_{c_l^{(0)},\mu}(c_l^{(0)}) \geq \sqrt{N_l(c_l^{(0)})} \cdot \frac{d_l}{\alpha_l(\mu)} - C_1],$$

and

$$\mathbb{P}[\hat{c}_r < c_r^{(0)}] \leq \mathbb{P}[\hat{S}_{\mu,c_r^{(0)}}(c_r^{(0)}) \geq \sqrt{N_r(c_r^{(0)})} \cdot \frac{d_r}{1 - \alpha_l(\mu)} - C_2],$$

where C_1, C_2 are positive constants, and C_1 only depends on $\alpha_l(\mu)$, d_l, C_2 only depends on $\alpha_r(\mu)$, d_r; $\hat{S}_{a,b}(t)$ is the slope at $F(t)$ of the least concave majorant in $[F(a), F(b)]$ of a standard Brownian Bridge in $[0, 1]$.

The proof of Theorem 22 is in fact reduced to proving any cutoff pair (c_l, c_r) with $c_l > c_l^{(0)}$ or $c_r < c_r^{(0)}$ does not satisfy the change-point-constraint with high probability. Since $\tilde{g}_l$ and $\tilde{g}_r$

estimated in Algorithm 5 are decreasing and increasing respectively, if $c_l > c_l^{(0)}$ (or $c_r < c_r^{(0)}$) violates the constraint, then c_l' (or c_r') will violate it with high probability if $c_l' > c_l$ (or $c_r' < c_r$). So it is reduced to considering the smallest $c_l > c_l^{(0)}$ and the largest $c_r < c_r^{(0)}$ in the grid searching space of Algorithm 5. Then the result can be concluded using Theorem 20. The detail is deferred to Appendix B.8.

Finally, we show in Theorem 23 that the identified $\hat{c}_r$ converges to $c_r^{(0)}$ at the rate of $\mathcal{O}([N_r(c_r^{(0)})]^{-\frac{1}{3}})$ if $m = \Omega(n)$. And the estimated density also converges to the true one at the rate of $\mathcal{O}(\max\{[N_r(c_r^{(0)})]^{-\frac{1}{3}}, [N_l(c_l^{(0)})]^{-\frac{1}{3}}\})$. The proof can be found in Appendix B.9.

Theorem 23 (L_1 Convergence of Identified Cutoff) *Suppose f is a distribution satisfying the constraint (3.4.1) and Assumption 2, with $\alpha_{mid}(c_l^{(0)}, c_r^{(0)}) > 0$, $f_{\max} < \infty$. Let $\Delta_l = c_l - c_l^{(0)}$, $\Delta_r = c_r - c_r^{(0)}$. If we have*

$$m \geq C_1 \cdot \max\{N_l(c_l^{(0)}), N_r(c_r^{(0)})\},$$

then

$$|\widehat{\Delta}_l| \leq C_2 \cdot N_l(c_l^{(0)})^{-\frac{1}{3}},\ |\widehat{\Delta}_r| \leq C_3 N_r(c_r^{(0)})^{-\frac{1}{3}},$$

where $\widehat{\Delta}_l$ and $\widehat{\Delta}_r$ are associated with $\hat{c}_l$ and $\hat{c}_r$ output by Algorithm 5. Furthermore, the resulting $\tilde{f}_{n,m}$ satisfies

$$\mathbb{E}_f \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \leq C_4 \cdot \left\{ N_l(c_l^{(0)})^{-\frac{1}{3}} + N_r(c_r^{(0)})^{-\frac{1}{3}} \right\}.$$

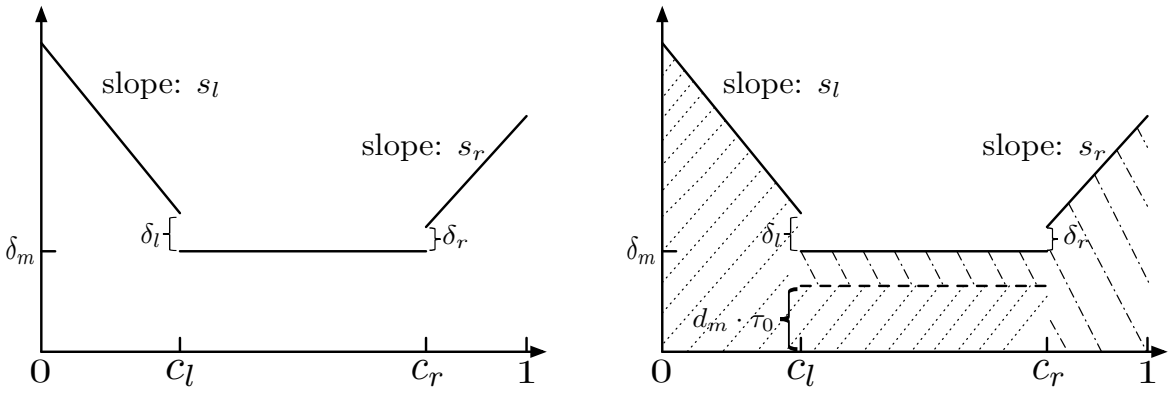
Here C_1, C_2, C_3, C_4 are positive constants that do not depend on n, $N_l(c_l^{(0)})$ and $N_r(c_r^{(0)})$.

Remark 2 *We want to point out that Theorem 21, Theorem 22 and Theorem 23 are based on the assumption that the underlying density f in BMU is bounded, thus we require $m = \Omega(n)$. If we further assume f is smooth, we might relax the condition on $m = \Omega(n)$ to $m = \Omega(\sqrt{n})$. The reason is that we can get a $\frac{1}{m}$ rate rather than a $\frac{1}{\sqrt{m}}$ rate on the truncated K-S distance of the CDF of $\hat{s}_i$'s and the CDF of s_i's when f is assumed to be smooth; see Proposition 12. This is an improved version of Proposition 8, which is the foundation of Theorem 19 and Theorem 20 that are used to prove the three theorems in this section.*

3.6 Numerical Experiments

3.6.1 Data generating process

To confirm and complement our theory, we use extensive numerical experiments to examine the finite performance of Ucut on the estimation of c_r. We study a U-shape model that is comprised of linear components. Specifically, the model consists of three parts with boundaries c_l and c_r. The middle part is a flat region of height δ_m. The left part is a segment with the right end at $(c_l, \delta_m + \delta_l)$ and slope $s_l < 0$ while the right part is a segment with the left end at $(c_r, \delta_m + \delta_r)$ and slope $s_r > 0$; see Figure 3.2 (a) for illustration. We call it the **linear valley** model. We normalize the linear valley model to produce the density of interest. We call the normalized gaps $\tilde{\delta}_l$ and $\tilde{\delta}_r$.



(a) original model

(b) two-group model

Figure 3.2: The linear valley model.

The linear valley model depicts the mixture density of the null distribution and the alternative distribution. We only assume that the left part ahead of c_l purely belongs to the null distribution while the right part ahead of c_r purely belongs to the alternative distribution. The null and the alternative distributions can hardly be distinguished in the flat middle part. To understand the linear valley model from the perspective of the two-group model, we assume that $\tau_0 \times 100\%$ of the middle part belongs to the null distribution while the remaining belongs to the alternative model. In Figure 3.2 (b), the part in left slash corresponds to the null density f_0 while the part in right slash corresponds to the alternative density f_1. Let π_0 be the area in the left slash divided by the total area. Then the marginal density can be

written as $f = \pi_0 f_0 + (1 - \pi_0) f_1$. Since any $\tau_0 \in [0, 1]$ gives the same f, the middle part is not identifiable. So it is necessary to estimate and infer the right cutoff c_r, so that we can safely claim all the samples beyond c_r are from the alternative distribution.

By default, we set $c_l = 0.3$, $c_r = 0.9$, $\delta_m = 1$, $\delta_l = 0.5$, $\delta_r = 0.5$, $s_l = -3$, $s_r = 1$. We sample $n = 10,000$ samples $\{s_1, \ldots, s_n\}$ from the linear valley model. Then for each $i \in \{1, \ldots, n\}$, we get the observations $\hat{s}_i \sim Binom(m, s_i)$ independently, where $m = 1,000$ if it is not specified particularly. The value of τ_0 does not affect the data generation but it affects the FDR and the power of any method that yields discoveries.

Finally, the left part and the right part are not necessary to be linear. To investigate the effect of general monotone cases and misspecified cases (e.g., unimodal densities), we replace the left part and the right part with other functions; see Section 3.6.5.

3.6.2 Robustness to model parameters

When using Algorithm 5, we use the middle point $\mu = 0.5$, the left gap parameter $d_l = 0.8 \cdot \tilde{\delta}_l$, the right gap parameter $d_r = 0.8 \cdot \tilde{\delta}_r$, the searching unit $\gamma = 0.001$. We first investigate how the binomial size m affects the estimation of c_r. Using the default setup as described in Section 3.6.1, we vary the binomial size $m \in \{10^2, 10^3, 2 \times 10^3, 5 \times 10^3, 10^4, Inf\}$, where Inf refers to the case that there is no binomial randomness and we observe s_i's directly. As shown in Figure 3.3, $\hat{c}_r$ converges to the true $c_r^{(0)}$ as m grows. When $m = 10^3$, the estimated c_r is as good as that of using s_i's directly. Note that in the linear setup, even with $m = 10^2$, $\hat{c}_r$ is larger than true $c_r^{(0)}$ with large probability. It implies that Ucut is safe in the sense that it will make few false discoveries by using $\hat{c}_r$ as the cutoff.

In the sequel, we stick to $m = 10^3$ since it works well enough for the linear valley model. We investigate whether the width of the middle flat region affects the estimation of c_r. We consider $c_l = 0.5 - w/2$, $c_r = 0.5 + w/2$ with $w \in \{0.6, 0.4, 0.2, 0.1, 0.\}$ while other model parameters are set by default. In Figure 3.4, the estimation of c_r is quite satisfying when the width is no smaller than 0.2. When the width drops to 0.1 or smaller, the estimation is not stable but still conservative in the sense that $\hat{c}_r > c_r$ in most cases.

Finally, we examine how the gap size influences the estimation of c_r. We take $\delta_l = \{0.5, 0.3, 0.20.1, 0.01\}$ and $\delta_r = \{0.5, 0.3, 0.20.1, 0.01\}$. Figure 3.5 shows that the estimation of c_r is robust to the gap sizes as long as the input d_l and d_r are smaller than the true gaps. This gives us confidence in applying Ucut to identify the cutoff even when there is no gap, which is more realistic.

3.6.3 Sensitivity of the algorithm hyper-parameters

Algorithm 5 (Ucut) mainly have three tuning parameters: the middle point μ, the left gap d_l and the right gap d_r. For practical use, the three tuning parameters may be misspecified.

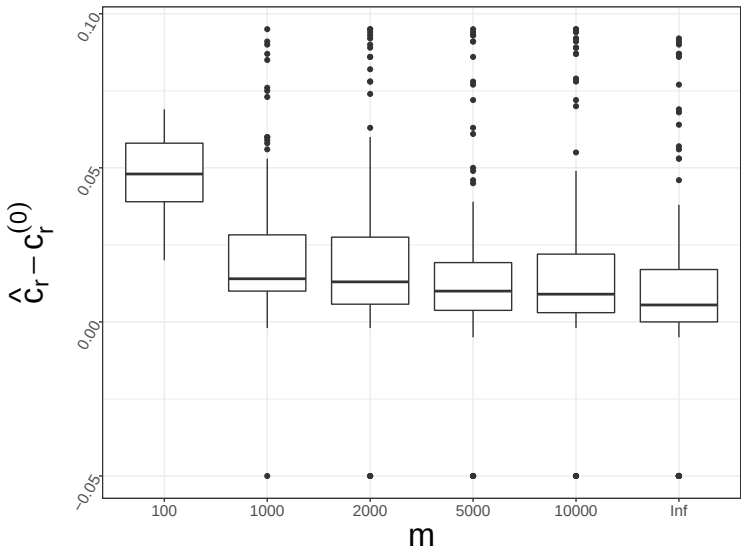


Figure 3.3: The convergence of $\hat{c}_r$ with respect to m.

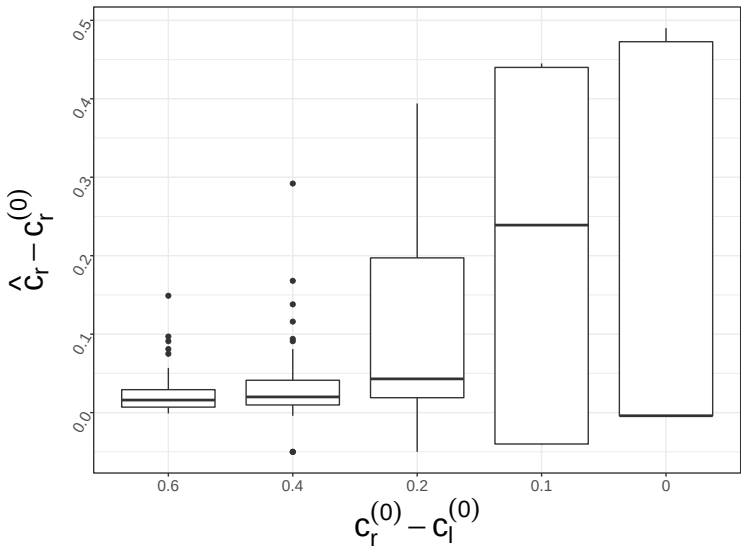


Figure 3.4: The estimation of $\hat{c}_r$ with respect to the width of the middle flat region.

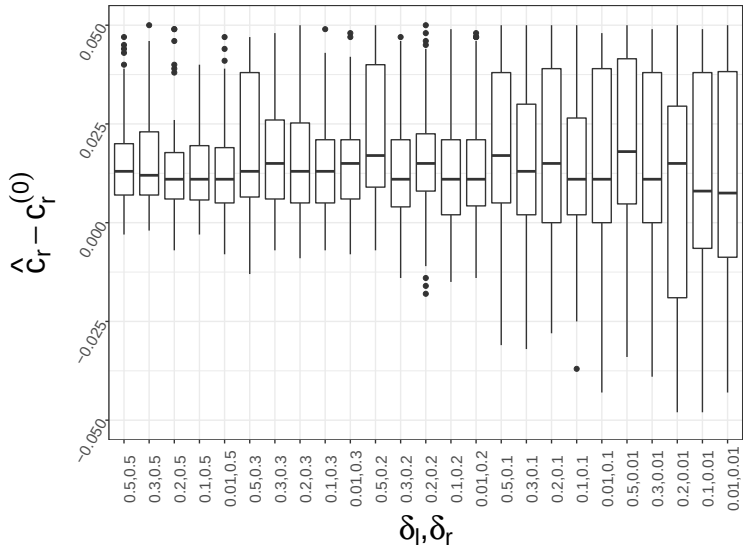


Figure 3.5: The estimation of $\hat{c}_r$ with respect to the gap sizes.

For example, the middle point is not easy to spot, or the left gap and the right gap are too small. We use the default model parameters as specified in Section 3.6.1. Let d_l and d_r be 0.8 times the true normalized gaps $\tilde{\delta}_l$ and $\tilde{\delta}_r$ respectively. We vary the choice of the middle point μ. Figure 3.6 shows that the estimation of c_r is not sensitive to the choice of μ as long as it is picked within the flat region $[0.3, 0.9]$. If μ is picked left to the flat region, the $\hat{c}_r$ has a larger variance but it is more conservative in the sense that $\hat{c}_r > c_r^{(0)}$ in most cases. If μ is picked right to the flat region, the $\hat{c}_r$ tends to be $\min\{\mu, c_r^{(0)}\}$.

Next, we fix $\mu = 0.5$ but consider $d_l = \kappa \times \tilde{\delta}_l$ and $d_r = \kappa \times \tilde{\delta}_r$, where $\kappa \in \{1, 0.9, 0.8, 0.5, 0.2, 0.1, 0.01\}$. We do not consider $\kappa > 1$ because there might not exist feasible (c_l, c_r) that satisfies the gap constraint. Figure 3.7 shows that when κ is within $[0.5, 1]$ the estimation is satisfying. The estimated c_r can be slightly smaller than the true $c_r^{(0)}$ when $\kappa < 0.5$ but in a tolerable range.

In a nutshell, the choices of μ, d_l and d_r are crucial to Algorithm 5. But the sensitivity analysis indicates that it is not necessary to be excessively cautious. In practice, picking these parameters by eyeballs can give a safe estimation in most cases.

3.6.4 Comparison to other methods

To examine the power and the capacity of controlling FDR of Ucut, we consider the two-group model as specified in Figure 3.2 (b). The middle part is not identifiable, which means that the samples of the alternative distribution can not be distinguished from those

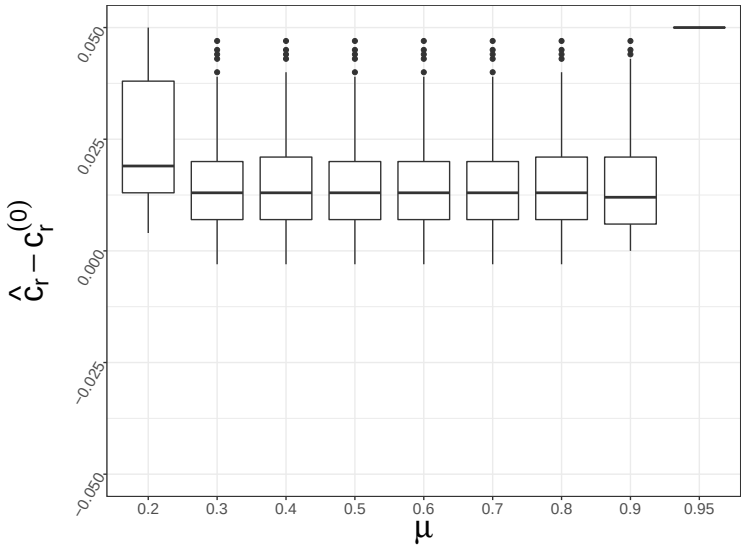


Figure 3.6: The estimation of $\hat{c}_r$ with respect to the choice of the middle point μ.

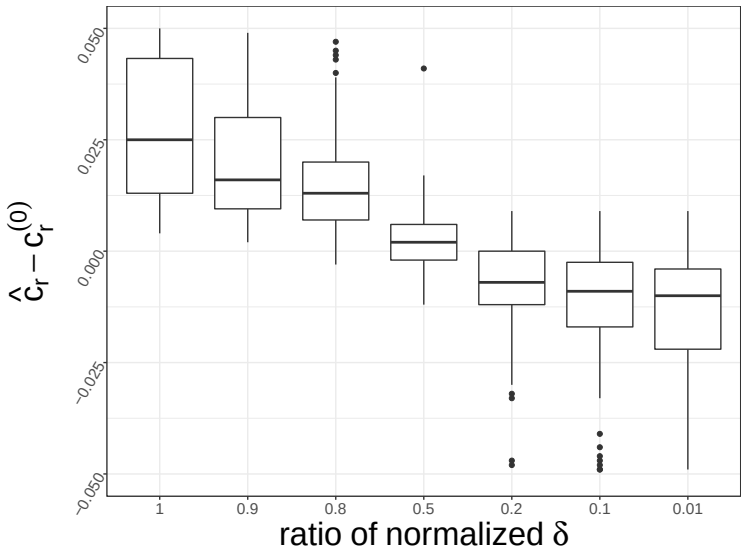


Figure 3.7: The estimation of $\hat{c}_r$ with respect to the choice of the input d_l and d_r. Here $d_l = \kappa \times \tilde{\delta}_l$ and $d_r = \kappa \times \tilde{\delta}_r$, where κ is a ratio of the normalized δ's.

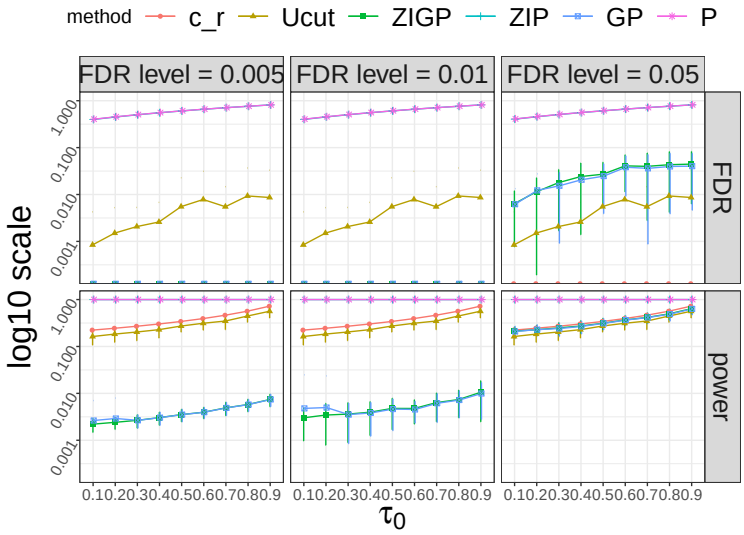


Figure 3.8: FDR and power of Ucut and other competing methods.

of the null distribution. To reflect this point, we arbitrarily set the proportion of the null distribution in the middle part τ_0. The goal is to identify the right cutoff c_r but not τ_0 because it is impossible to infer τ_0.

We compare Ucut to the other four methods that are studied in Gauran et al. (2018), i.e., ZIGP (Zero-inflated Generalized Poisson), ZIP (Zero Inflated Poisson), GP (Generalized Poisson) and P (Poisson). These four methods are used to make decisions on the cutoff for zero-inflated discrete mixture distributions.

Figure 3.8 shows that GP and P use rather small cutoffs and have too large FDRs. ZIGP and ZIP are over-conservative if the target FDR level is too low at 0.005 or 0.01, thus having quite low power. They perform better when the target FDR level is set to be 0.05. On the other hand, Ucut can control FDR at 0.01 if we directly use $\hat{c}_r$ as the cutoff. The associated power is better than those of ZIGP and ZIP. In order to loosen the FDR control and get higher power, it is fine to use a slightly smaller cutoff than $\hat{c}_r$. From this result, we confirm that Ucut is a better fit for the scenario where the middle part is not distinguishable.

3.6.5 More simulation studies

Besides the linear valley model specified in Section 3.6.1, we also consider a non-linear model and a misspecified model.

For the non-linear model, we replace the left linear part in the linear valley model with an unnormalized decreasing function $f_l = Beta(x/c_l; 0.5, 1.5)/c_l \cdot 3/20$, $x \in [0, c_l]$. We replace the

right linear part with an unnormalized increasing function $f_r = Beta(x/(1-c_r); 2, 0.8)/(1-c_r) \cdot 1/20$, $x \in (c_r, 1]$. Here $Beta(x; \alpha, \beta)$ is a density of Beta distribution with parameters α and β. In this case, we use $m = 10^4$. Applying Ucut to the synthetic data generated by this model, we observe similar results as those from the linear valley model. It indicates that Ucut can detect the cutoff in a satisfying range as long as the underlying model satisfies the gapped U-shape constraint. Ucut is particularly useful when the the middle part is "uniform" and not easy to tell apart the samples of the alternative distribution from those of the null distribution.

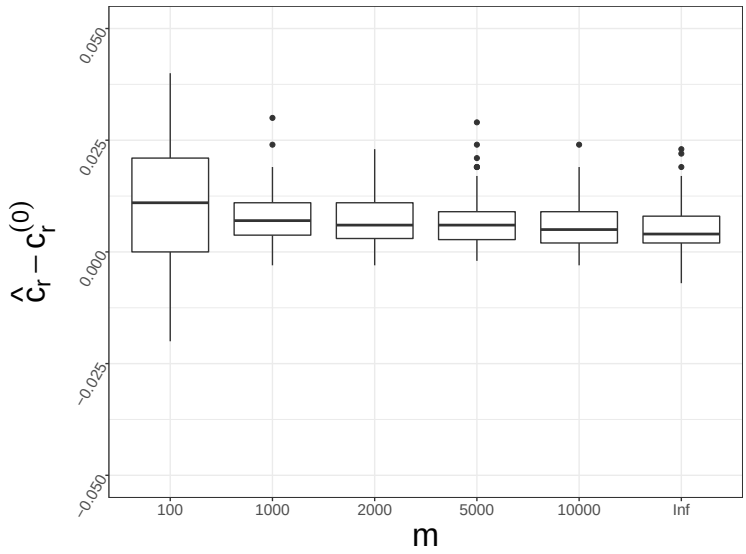


Figure 3.9: The convergence of $\hat{c}_r$ with respect to m on the non-linear decreasing-uniform-increasing model.

For the misspecified model, we replace the left linear part with an unnormalized unimodal function $f_l = Beta(x/c_l; 1.5, 5)/c_l \cdot 3$, $x \in [0, c_l]$. We replace the right linear part with an unnormalized unimodal function $f_r = Beta(x/(1-c_r); 2.5, 1.5)/(1-c_r)$, $x \in (c_r, 1]$. In this case, the estimated c_r is not satisfying until m attains 10^4; see Figure 3.15. So in the other experiments, we use $m = 10^4$. We find that although the variance of $\hat{c}_r$ becomes larger than the estimate for the correctly specified model, the detected cutoff tends to be larger than the truth. It implies that if the model does not align with the gapped U-shape constraint, Ucut is still useful because it is conservative.

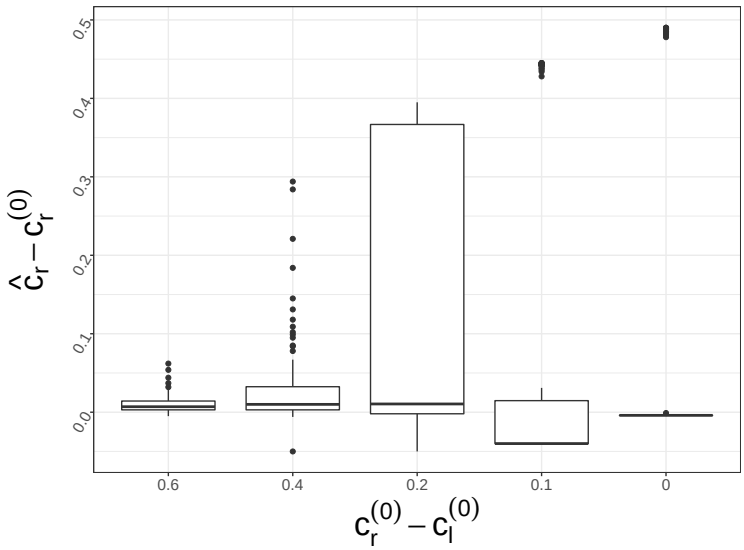


Figure 3.10: The estimation of $\hat{c}_r$ with respect to the width of the middle flat region on the non-linear decreasing-uniform-increasing model.

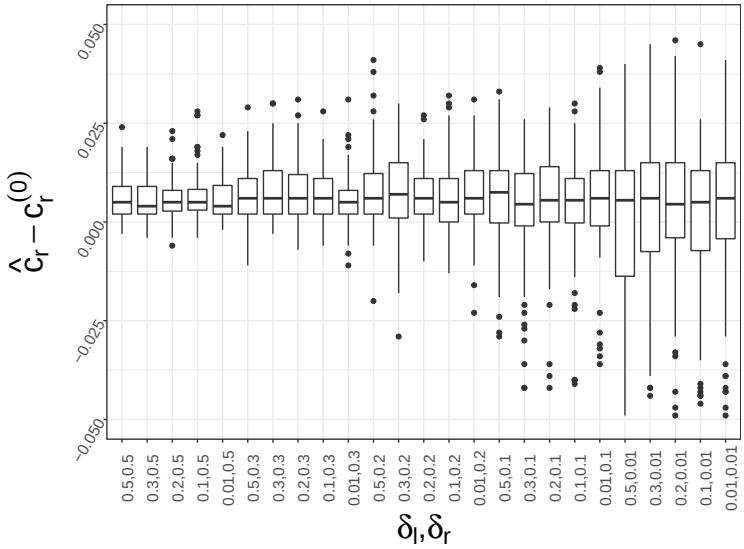


Figure 3.11: The estimation of $\hat{c}_r$ with respect to the gap sizes on the non-linear decreasing-uniform-increasing model.

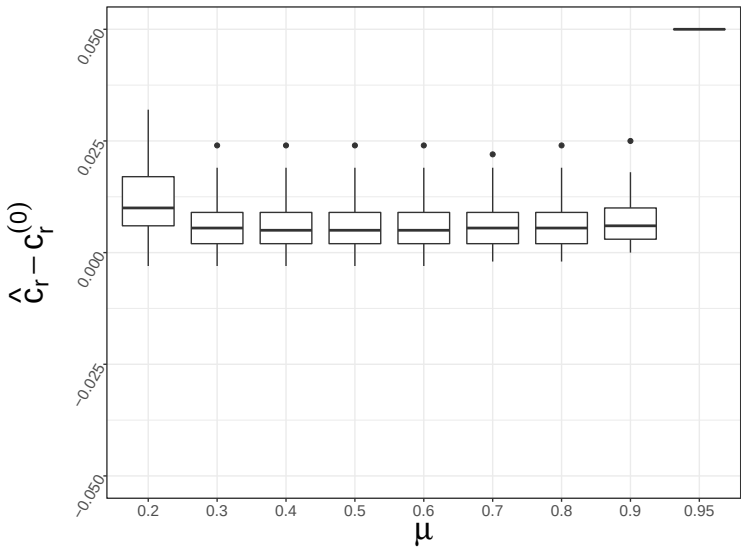


Figure 3.12: The estimation of $\hat{c}_r$ with respect to the choice of the middle point μ on the non-linear decreasing-uniform-increasing model.

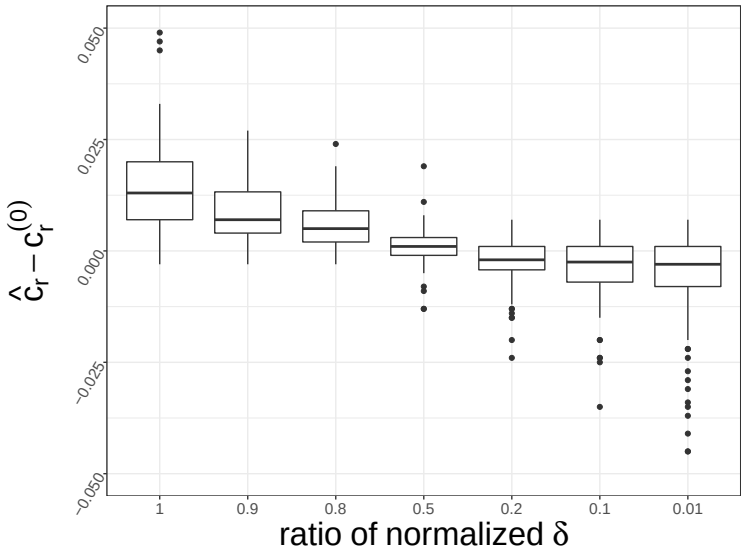


Figure 3.13: The estimation of $\hat{c}_r$ with respect to the choice of the input d_l and d_r on the non-linear decreasing-uniform-increasing model.. Here $d_l = \kappa \times \tilde{\delta}_l$ and $d_r = \kappa \times \tilde{\delta}_r$, where κ is a ratio of the normalized δ's.

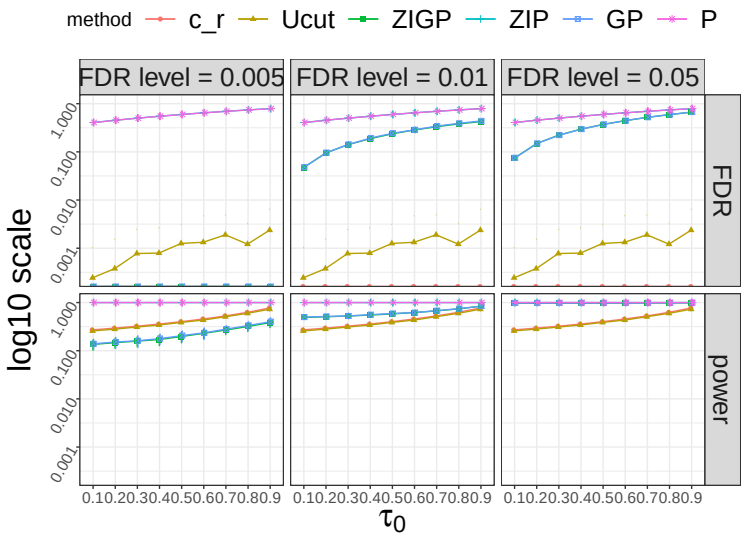


Figure 3.14: FDR and power of Algorithm 5 and other competing methods on the non-linear decreasing-uniform-increasing model.

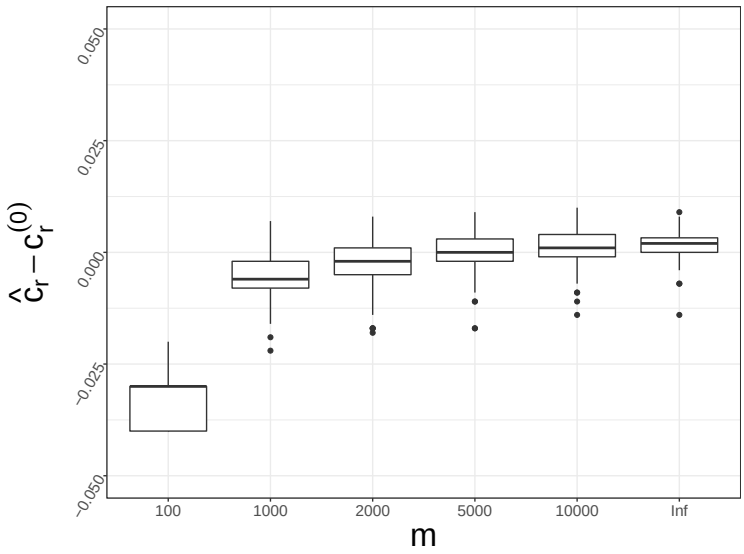


Figure 3.15: The convergence of $\hat{c}_r$ with respect to m on the misspecified model.

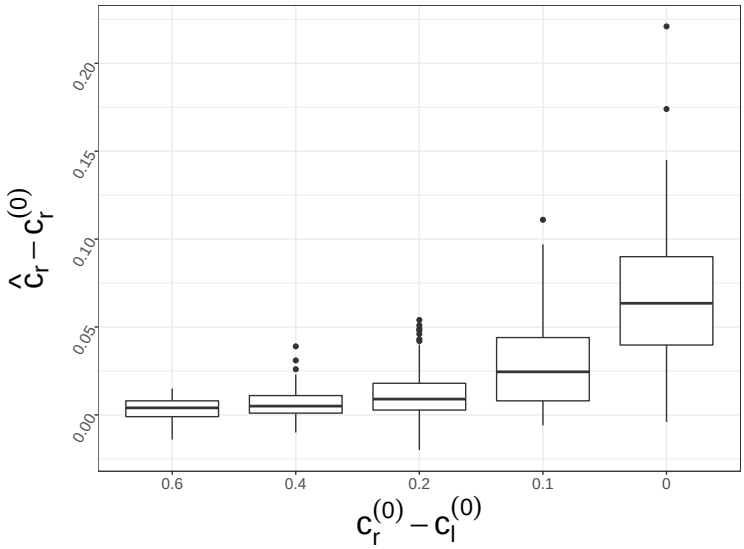


Figure 3.16: The estimation of $\hat{c}_r$ with respect to the width of the middle flat region on the misspecified model.

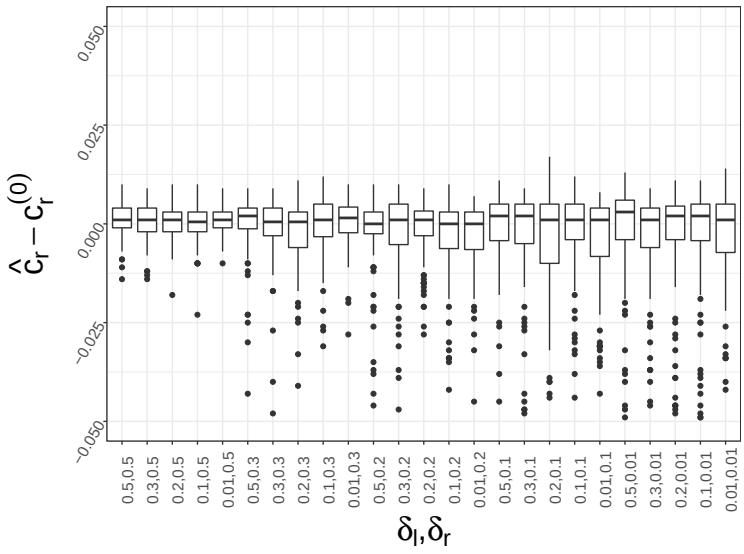


Figure 3.17: The estimation of $\hat{c}_r$ with respect to the gap sizes on the misspecified model.

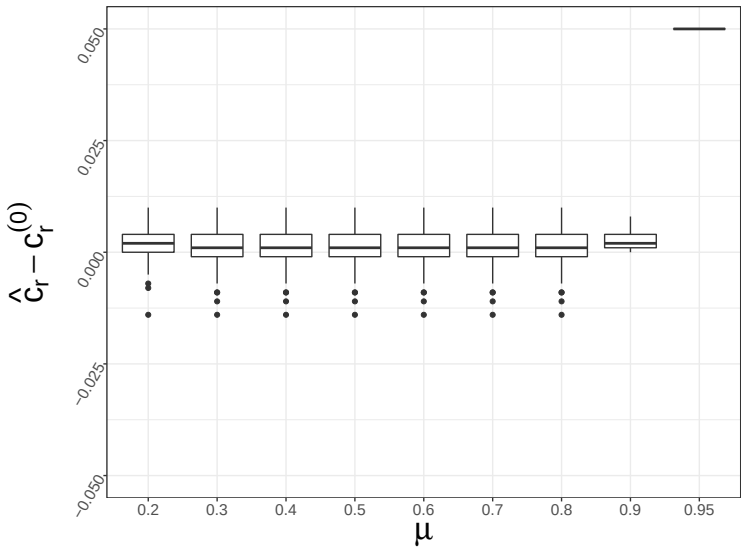


Figure 3.18: The estimation of $\hat{c}_r$ with respect to the choice of the middle point μ on the misspecified model.

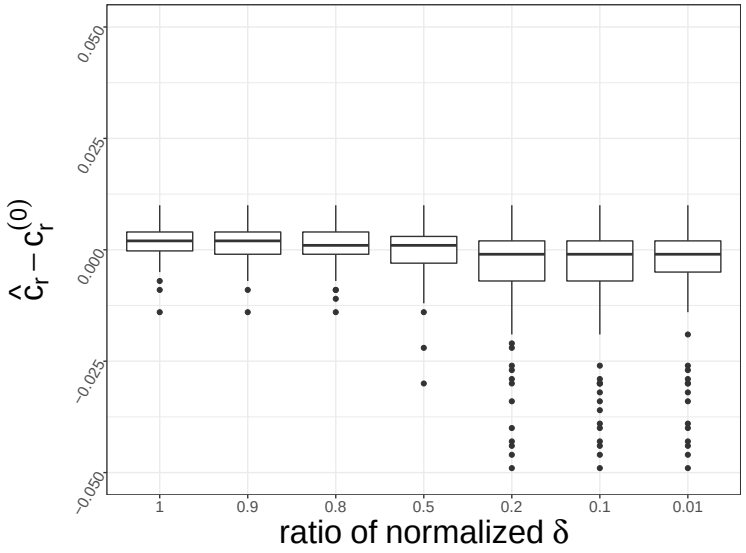


Figure 3.19: The estimation of $\hat{c}_r$ with respect to the choice of the input d_l and d_r on the misspecified model. Here $d_l = \kappa \times \tilde{\delta}_l$ and $d_r = \kappa \times \tilde{\delta}_r$, where κ is a ratio of the normalized δ's.

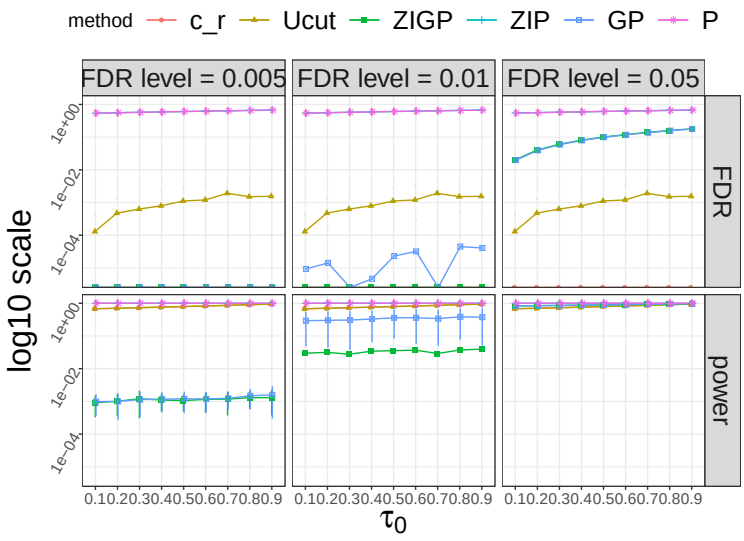


Figure 3.20: FDR and power of Algorithm 5 and other competing methods on the misspecified model.

3.7 Application to Real Data

We further demonstrate the performance of Ucut on the real datasets used in Liu et al. (2019). To be specific, we apply the GeneFishing method to four GTEx RNAseq datasets, liver, Artery Coronary, Transverse Colon, and Testis; see Table 3.1 for details. We leverage the same set of 21 bait genes used in Liu et al. (2019). The number of fishing rounds is set to be $m = 10,000$.

Table 3.1: Details of GTEx RNAseq datasets.

	# samples	# genes
Liver	119	18,845
Artery-Coronary	133	20,597
Colon-Transverse	196	21,695
Testis	172	31,931

Once the CFRs are generated, we apply Algorithm 5 with the middle point $\mu = 0.5$, $d_l = 0.1$ and $d_r = 0.01$. We take d_l to be ten times d_r because there are much more zeros than ones in CFRs. As shown in Section 3.6, Ucut is not sensitive to the three parameters. The change of these parameters lays little influence on the results. Table 3.2 shows that for each tissue Ucut gives the estimator of c_r that yields 50 to 80 discoveries. We estimate the false discovery rate using the second approach in Liu et al. (2019). Note that for Artery-Coronary, the estimated $c_r = 0.972$ by Ucut (which gives $\widehat{FDR} \approx 10^{-3}$) is less extreme than simply using 0.990 (which gives $\widehat{FDR} \approx 10^{-4}$). It implicates that Ucut adapts to the tissue and can pick a cutoff with a reasonable false discovery rate.

Table 3.2: Estimation of c_r by Algorithm 5 on four tissues, where $\mu = 0.5$, $d_l = 0.1$ and $d_r = 0.01$. The second column is the estimated c_r using bootstrap by sampling 70% of the CFRs.

	$\hat{c}_r$	bootstrapping estimation	# discovery (use $\hat{c}_r$)	$\widehat{FDR}$
Liver	0.995	0.993(0.005)	52	1.4×10^{-3}
Artery-Coronary	0.972	0.976(0.009)	85	5.7×10^{-3}
Colon-Transverse	0.989	0.991(0.049)	57	1.2×10^{-4}
Testis	0.993	0.992(0.001)	73	0.010

In addition, we also apply the GeneFishing method to a single-cell data of the pancreas cells from Tabula Muris[1]. It contains 849 cells from mice and 5,220 genes expressed in enough

[1]https://tabula-muris.ds.czbiohub.org

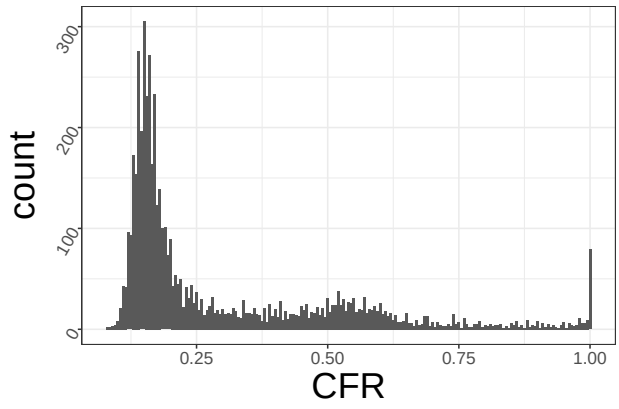


Figure 3.21: The CFRs of the single cell data of the pancreas tissue.

cells out of about $20,000$ genes. We find out 9 bait genes based on the pancreas insulin secretion gene ontology (GO) term.

Unlike Figure 1.2, the CFRs of this data set do not appear in a U shape. Instead, we observe a unimodal pattern around zero and an increasing pattern around one (Figure 3.21). Nonetheless, it does not hinder us from using Ucut to determine the cutoff, since we are mainly concerned about the right cutoff and Section 3.6 demonstrates that Ucut is conservative even if the model is misspecified. By using $\mu = 0.5$, $d_l = 0.1$ and $d_r = 0.01$, Ucut gives 0.994 as the estimation of the right cutoff, which discovers 77 genes. By doing the GO enrichment analysis, we find out that these identified genes are enriched for the GO of response to ethanol with p-value 0.0021, the GO of positive regulation of fatty acid biosynthesis with p-value 0.0055, and the GO of eating behavior with p-value 0.0079. These GOs have been shown to relate to insulin secretion in literature (Huang and Sjöholm, 2008; Nolan et al., 2006; Tanaka et al., 2003), which indicates the effectiveness of Ucut.

3.8 Discussion

In this work, we analyze the binomial mixture model (3.1.2) under the U-shape constraint, which is motivated by the results of the GeneFishing method (Liu et al., 2019). The contributions of this work are two-fold. First, to the best of our knowledge, this is the pioneering work that investigates the relationship between the binomial size m and the sample size n for the binomial mixture model under various conditions for F. Second, we provide a convenient tool to help the downstream decision-making of the GeneFishing method.

Despite the identifiability issue of the binomial mixture model, we show that the estimator of the underlying distribution deviates from the true distribution, in some distance, at most

a small quantity that depends on m. The implication is that to have the same convergence rate as if there is no binomial randomness, we need m to be at the same order as n for the empirical CDF and the Grenander estimator when the underlying density is bounded. However, when the underlying density is smooth, the simulation studies and the theoretical results imply that the condition can be relaxed to $m \asymp n^{1/2}$ while the histogram estimator requires $m \asymp n^{2/3}$. It is of great interest to further investigate how the minimal m hinges on the smoothness of the underlying distribution, e.g., studying the kernel density estimator and the smoothing spline estimator under the binomial mixture model.

To answer the motivating question of how large the CFR should be so that the associated sample can be regarded as a discovery in the GeneFishing method, we propose a U-shape model to depict the underlying distribution and an NPMLE method Ucut to determine the cutoff. This estimator comprises two Grenander estimators, thus having a cubic convergence rate as the Grenander estimator when m is large enough. We also show that the estimated cutoff is larger than the true cutoff with high probability. The simulation studies indicate that Ucut is robust to the three hyper-parameters, even if the model is misspecified. Therefore, we recommend Ucut as a cost-effective and robust tool for the downstream analysis of the GeneFishing method.

Bibliography

Alves, Roberto Teixeira, MR Delgado, and Alex Alves Freitas (2010). "Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions". In: *Fuzzy Systems (FUZZ), 2010 IEEE International Conference*. IEEE, pp. 1–8.

Ananpiriyakul, Thanawut, Piyapan Poomsirivilai, and Peerapon Vateekul (2014). "Label correction strategy on hierarchical multi-label classification". In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, pp. 213–227.

Bahadur, Raghu Raj (1960). "Some approximations to the binomial distribution function". In: *The Annals of Mathematical Statistics*, pp. 43–54.

Balabdaoui, Fadoua et al. (2011). "On the Grenander estimator at zero". In: *Statistica Sinica* 21.2, p. 873.

Baraniuk, Richard G (1999). "Optimal tree approximation with wavelets". In: *Wavelet Applications in Signal and Image Processing VII*. Vol. 3813. International Society for Optics and Photonics, pp. 196–208.

Baraniuk, Richard G and Douglas L Jones (1994). "A signal-dependent time-frequency representation: Fast algorithm for optimal kernel design". In: *Signal Processing, IEEE Transactions* 42.1, pp. 134–146.

Baraniuk, Richard G et al. (2010). "Model-based compressive sensing". In: *IEEE Transactions on Information Theory* 56.4, pp. 1982–2001.

Barndorff-Nielsen, O (1965). "Identifiability of mixtures of exponential families". In: *Journal of Mathematical Analysis and Applications* 12.1, pp. 115–121.

Barutcuoglu, Zafer, Robert E Schapire, and Olga G Troyanskaya (2006). "Hierarchical multi-label prediction of gene function". In: *Bioinformatics* 22.7, pp. 830–836.

Bi, Wei and Jame T Kwok (2015). "Bayes-Optimal Hierarchical Multilabel Classification". In: *IEEE Transactions on Knowledge and Data Engineering* 27.11, pp. 2907–2918.

Bi, Wei and James T Kwok (2011). "Multi-label classification on tree-and dag-structured hierarchies". In: *Proceedings of the 28th International Conference on Machine Learning*, pp. 17–24.

Birge, Lucien (1989). "The grenander estimator: A nonasymptotic approach". In: *The Annals of Statistics*, pp. 1532–1549.

Blockeel, Hendrik et al. (2002). "Hierarchical multi-classification". In: *Proceedings of the ACM SIGKDD 2002 Workshop on Multi-relational Data Mining (MRDM 2002)*, pp. 21–35.

Blockeel, Hendrik et al. (2006). "Decision trees for hierarchical multilabel classification: A case study in functional genomics". In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, pp. 18–29.

Buckner, Randy L, Fenna M Krienen, and BT Thomas Yeo (2013). "Opportunities and limitations of intrinsic functional connectivity MRI". In: *Nature neuroscience* 16.7, pp. 832–837.

Cai, Li and Yangyong Zhu (2015). "The challenges of data quality and data quality assessment in the big data era". In: *Data science journal* 14.

Cesa-Bianchi, Nicolò, Claudio Gentile, and Luca Zaniboni (2006a). "Hierarchical classification: combining Bayes with SVM". In: *Proceedings of the 23rd International Conference on Machine Learning*. ACM, pp. 177–184.

— (2006b). "Incremental algorithms for hierarchical classification". In: *Journal of Machine Learning Research* 7.Jan, pp. 31–54.

Clare, Amanda (2003). "Machine learning and data mining for yeast functional genomics". PhD thesis. The University of Wales, Aberystwyth.

Costa, E et al. (2007). "A review of performance evaluation measures for hierarchical classifiers". In: *Evaluation Methods for Machine Learning II: Papers from the AAAI-2007 Workshop, AAAI Technical Report WS-07-05*. AAAI Press, pp. 182–196.

Davis, Jesse and Mark Goadrich (2006). "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. ACM, pp. 233–240.

DeCoro, Christopher, Zafer Barutcuoglu, and Rebecca Fiebrink (2007). "Bayesian Aggregation for Hierarchical Genre Classification." In: *International Society for Music Information Retrieval*, pp. 77–80.

Dimitrovski, Ivica et al. (2011). "Hierarchical annotation of medical images". In: *Pattern Recognition* 44.10, pp. 2436–2449.

Dunbar, Steven R (2011). "The de Moivre-Laplace Central Limit Theorem". In: *Technical Report*.

Dvoretzky, Aryeh, Jack Kiefer, and Jacob Wolfowitz (1956). "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator". In: *The Annals of Mathematical Statistics*, pp. 642–669.

Efron, Bradley (2012). *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*. Vol. 1. Cambridge University Press.

Fisher, Ronald A (1922). "On the mathematical foundations of theoretical statistics". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 222.594-604, pp. 309–368.

Fraser, Martin D, Yu Sheng Hsu, and Walker JJ (1981). "Identifiability of finite mixtures of von Mises distributions". In: *The Annals of Statistics* 9.5, pp. 1130–1134.

Gauch, Susan, Aravind Chandramouli, and Shankar Ranganathan (2009). "Training a hierarchical classifier using inter document relationships". In: *Journal of the Association for Information Science and Technology* 60.1, pp. 47–58.

Gauran, Iris Ivy M et al. (2018). "Empirical null estimation using zero-inflated discrete mixture distributions and its application to protein domain data". In: *Biometrics* 74.2, pp. 458–471.

Grenander, Ulf (1956). "On the theory of mortality measurement: part ii". In: *Scandinavian Actuarial Journal* 1956.2, pp. 125–153.

Grilli, Leonardo, Carla Rampichini, and Roberta Varriale (2015). "Binomial mixture modeling of university credits". In: *Communications in Statistics-Theory and Methods* 44.22, pp. 4866–4879.

Hand, David J (2009). "Measuring classifier performance: a coherent alternative to the area under the ROC curve". In: *Machine learning* 77.1, pp. 103–123.

Haque, Ashraful et al. (2017). "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications". In: *Genome medicine* 9.1, pp. 1–12.

Hasin, Yehudit, Marcus Seldin, and Aldons Lusis (2017). "Multi-omics approaches to disease". In: *Genome biology* 18.1, pp. 1–15.

Heart, Tsipi, Ofir Ben-Assuli, and Itamar Shabtai (2017). "A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy". In: *Health Policy and Technology* 6.1, pp. 20–25.

Herskovic, Jorge R, M Sriram Iyengar, and Elmer V Bernstam (2007). "Using hit curves to compare search algorithm performance". In: *Journal of Biomedical Informatics* 40.2, pp. 93–99.

Hoeffding, Wassily (1963). "Probability Inequalities for Sums of Bounded Random Variables". In: *Journal of the American Statistical Association* 58.301, pp. 13–30.

Holden, Nicholas and Alex Alves Freitas (2005). "A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data". In: *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE, pp. 100–107.

Hotelling, Harold (1933). "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6, p. 417.

Huang, Haiyan, Chun-Chi Liu, and Xianghong Jasmine Zhou (2010). "Bayesian approach to transforming public gene expression repositories into disease diagnosis databases". In: *Proceedings of the National Academy of Sciences* 107.15, pp. 6823–6828.

Huang, Jian and Jon A Wellner (1995). "Estimation of a monotone density or monotone hazard under random censoring". In: *Scandinavian Journal of Statistics*, pp. 3–33.

Huang, Youping and Cun-Hui Zhang (1994). "Estimating a monotone density from censored observations". In: *The Annals of Statistics*, pp. 1256–1274.

Huang, Zhen and Åke Sjöholm (2008). "Ethanol acutely stimulates islet blood flow, amplifies insulin secretion, and induces hypoglycemia via nitric oxide and vagally mediated mechanisms". In: *Endocrinology* 149.1, pp. 232–236.

Jankowski, Hanna K and Jon A Wellner (2009). "Estimation of a discrete monotone distribution". In: *Electronic journal of statistics* 3, p. 1567.

Jiang, Ci-Ren et al. (2014). "Optimal Ranking in Multi-label Classification Using Local Precision Rates". In: *Statistica Sinica* 24.4, pp. 1547–1570.

Kent, John T (1983). "Identifiability of finite mixtures for directional data". In: *The Annals of Statistics*, pp. 984–988.

Kéry, Marc (2008). "Estimating abundance from bird counts: binomial mixture models uncover complex covariate relationships". In: *The Auk* 125.2, pp. 336–345.

Kiritchenko, Svetlana, Stan Matwin, and Fazel Famili (2005). "Functional annotation of genes using hierarchical text categorization". In: *BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05)*, pp. 1–4.

Koller, Daphne and Mehran Sahami (1997). "Hierarchically Classifying Documents Using Very Few Words". In: *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., pp. 170–178.

Kolodziejczyk, Aleksandra A et al. (2015). "The technology and biology of single-cell RNA sequencing". In: *Molecular cell* 58.4, pp. 610–620.

Komlós, János, Péter Major, and Gábor Tusnády (1975). "An approximation of partial sums of independent RV'-s, and the sample DF. I". In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 32.1-2, pp. 111–131.

Lee, Brian K, Justin Lessler, and Elizabeth A Stuart (2011). "Weight trimming and propensity score weighting". In: *PLOS One* 6.3.

Lee, Wayne T. (2013). "Bayesian Analysis in Problems with High Dimensional Data and Complex Dependence Structure". PhD thesis. University of California, Berkeley.

Lewis, David D et al. (2004). "Rcv1: A new benchmark collection for text categorization research". In: *Journal of Machine Learning Research* 5.Apr, pp. 361–397.

Liu, Ke et al. (2019). "GeneFishing to reconstruct context specific portraits of biological processes". In: *Proceedings of the National Academy of Sciences* 116.38, pp. 18943–18950.

Lord, Frederic M (1969). "Estimating true-score distributions in psychological testing (an empirical Bayes estimation problem)". In: *Psychometrika* 34.3, pp. 259–299.

Lord, Frederic M and Noel Cressie (1975). "An empirical Bayes procedure for finding an interval estimate". In: *Sankhyā: The Indian Journal of Statistics, Series B*, pp. 1–9.

Lüxmann-Ellinghaus, U (1987). "On the identifiability of mixtures of infinitely divisible power series distributions". In: *Statistics & probability letters* 5.5, pp. 375–378.

Maaten, Laurens Van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.86, pp. 2579–2605.

Marshall, Albert W and Frank Proschan (1965). "Maximum likelihood estimation for distributions with monotone failure rate". In: *The annals of mathematical statistics* 36.1, pp. 69–77.

Massart, Pascal (1990). "The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality". In: *The annals of Probability*, pp. 1269–1283.

McDonald, Trent et al. (2020). "Evidence of Absence Regression: A Binomial N-Mixture Model for Estimating Bird and Bat Fatalities at Wind Power Facilities". In: *bioRxiv*. DOI: 10.1101/2020.01.21.914754.

Mead, Al (1992). "Review of the development of multidimensional scaling methods". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 41.1, pp. 27–39.

Ng, Andrew and Michael Jordan (2001). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in Neural Information Processing Systems*, pp. 841–848.

Nolan, Christopher J et al. (2006). "Fatty acid signaling in the β-cell and insulin secretion". In: *Diabetes* 55.Supplement 2, S16–S23.

Nowak, Stefanie et al. (2010). "Performance measures for multilabel evaluation: a case study in the area of image classification". In: *Proceedings of the International Conference on Multimedia Information Retrieval*. ACM, pp. 35–44.

Nuwaysir, Emile F et al. (2002). "Gene expression analysis using oligonucleotide arrays produced by maskless photolithography". In: *Genome research* 12.11, pp. 1749–1755.

O'Donnell, Katherine M, Frank R Thompson III, and Raymond D Semlitsch (2015). "Partitioning detectability components in populations subject to within-season temporary emigration using binomial mixture models". In: *PLoS One* 10.3, e0117216.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *International Conference on Machine Learning*, pp. 1310–1318.

Patil, GP and Sheela Bildikar (1966). "Identifiability of countable mixtures of discrete probability distributions using methods of infinite matrices". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 62. 3. Cambridge University Press, pp. 485–494.

Pearson, Karl (1901). "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572.

Pillai, Ignazio, Giorgio Fumera, and Fabio Roli (2013). "Threshold optimisation for multi-label classifiers". In: *Pattern Recognition* 46.7, pp. 2055–2065.

Ploner, Alexander et al. (2006). "Multidimensional local false discovery rate for microarray studies". In: *Bioinformatics* 22.5, pp. 556–565.

Rao, BLS Prakasa (1970). "Estimation for distributions with monotone failure rate". In: *The annals of mathematical statistics*, pp. 507–519.

Rousu, Juho et al. (2006). "Kernel-based learning of hierarchical multilabel classification models". In: *Journal of Machine Learning Research* 7.Jul, pp. 1601–1626.

Royle, J Andrew (2004). "N-mixture models for estimating population size from spatially replicated counts". In: *Biometrics* 60.1, pp. 108–115.

Sapatinas, Theofanis (1995). "Identifiability of mixtures of power-series distributions and related characterizations". In: *Annals of the Institute of Statistical Mathematics* 47.3, pp. 447–459.

Shen, Jiayi et al. (2019). "Artificial intelligence versus clinicians in disease diagnosis: systematic review". In: *JMIR medical informatics* 7.3, e10010.

Silla, Carlos N and Alex A Freitas (2009). "Novel top-down approaches for hierarchical classification and their application to automatic music genre classification". In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference*. IEEE, pp. 3499–3504.

Silva, Michael A et al. (2018). "Challenges and techniques for presurgical brain mapping with functional MRI". In: *NeuroImage: Clinical* 17, pp. 794–803.

Sivaganesan, S and James Berger (1993). "Robust Bayesian analysis of the binomial empirical Bayes problem". In: *Canadian Journal of Statistics* 21.1, pp. 107–119.

Sporns, Olaf (2013). "Structure and function of complex brain networks". In: *Dialogues in clinical neuroscience* 15.3, pp. 247–262.

Sun, Aixin and Ee-Peng Lim (2001). "Hierarchical text classification and evaluation". In: *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference*. IEEE, pp. 521–528.

Tallis, GM (1969). "The identifiability of mixtures of distributions". In: *Journal of Applied Probability* 6.2, pp. 389–398.

Tallis, GM and Peter Chesson (1982). "Identifiability of mixtures". In: *Journal of the Australian Mathematical Society* 32.3, pp. 339–348.

Tanaka, Muneki et al. (2003). "Eating pattern and the effect of oral glucose on ghrelin and insulin secretion in patients with anorexia nervosa". In: *Clinical endocrinology* 59.5, pp. 574–579.

Teicher, Henry (1963). "Identifiability of finite mixtures". In: *The annals of Mathematical statistics*, pp. 1265–1269.

Teicher, Henry et al. (1961). "Identifiability of mixtures". In: *The annals of Mathematical statistics* 32.1, pp. 244–248.

Thomas, Hoben (1989). "A binomial mixture model for classification performance: A commentary on Waxman, Chambers, Yntema, and Gelman (1989)". In: *Journal of Experimental Child Psychology* 48.3, pp. 423–430.

Triguero, Isaac and Celine Vens (2016). "Labelling strategies for hierarchical multi-label classification techniques". In: *Pattern Recognition* 56, pp. 170–183.

Valentini, Giorgio (2009). "True path rule hierarchical ensembles". In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 232–241.

— (2011). "True path rule hierarchical ensembles for genome-wide gene function prediction". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8.3, pp. 832–847.

Van der Vaart, Aad and Mark Van der Laan (2003). "Smooth estimation of a monotone density". In: *Statistics: A Journal of Theoretical and Applied Statistics* 37.3, pp. 189–203.

Vens, Celine et al. (2008). "Decision trees for hierarchical multi-label classification". In: *Machine Learning* 73.2, pp. 185–214.

Verspoor, Karin et al. (2006). "A categorization approach to automated ontological function annotation". In: *Protein Science* 15.6, pp. 1544–1549.

Vilanova, Cristina and Manuel Porcar (2016). "Are multi-omics enough?" In: *Nature microbiology* 1.8, pp. 1–2.

Wainwright, Martin J and Michael Irwin Jordan (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Wang, Y. (1992). *Nonparametric Estimation Subject to Shape Restrictions*. University of California, Berkeley. URL: `https://books.google.com/books?id=QmVMAQAAMAAJ`.

Wang, Zhong, Mark Gerstein, and Michael Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature reviews genetics* 10.1, pp. 57–63.

Wasserman, Larry (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Weston, Jason et al. (2003). "Kernel dependency estimation". In: *Advances in Neural Information Processing Systems*, pp. 897–904.

Wood, GR et al. (1999). "Binomial mixtures: geometric estimation of the mixing distribution". In: *The Annals of Statistics* 27.5, pp. 1706–1721.

Woodroofe, Michael and Jiayang Sun (1993). "A penalized maximum likelihood estimate of f (0+) when f is non-increasing". In: *Statistica Sinica*, pp. 501–515.

Wu, Feihong, Jun Zhang, and Vasant Honavar (2005). "Learning classifiers using hierarchically structured class taxonomies". In: *International Symposium on Abstraction, Reformulation, and Approximation*. Springer, pp. 313–320.

Wu, Guohui et al. (2015). "Bayesian binomial mixture models for estimating abundance in ecological monitoring studies". In: *Annals of Applied Statistics* 9.1, pp. 1–26.

Yakowitz, Sidney J and John D Spragins (1968). "On the identifiability of finite mixtures". In: *The Annals of Mathematical Statistics*, pp. 209–214.

Zhang, Min-Ling and Zhi-Hua Zhou (2013). "A review on multi-label learning algorithms". In: *IEEE Transactions on Knowledge and Data Engineering* 26.8, pp. 1819–1837.

Appendix A

Appendix of Chapter 2

A.1 Proof of Theorem 4

Proof The simplest case is a graph with only one chain, and the theorem obviously holds for this case. Next, suppose there are two chain branches, $X_{(1)} \to \ldots \to X_{(m)}$ and $Y_{(1)} \to \ldots \to Y_{(m')}$, both of which share the same parent node in $\mathcal{P}_2$. Directly merging these two chain branches by Algorithm 1 yields an ordering denoted as o_{XY}. Now given an arbitrary ordering o_A of all the n nodes in the entire graph, which respects the tree hierarchy. Denote by p_1 the first position of the nodes among $X'_{(i)}s$ and $Y'_{(j)}s$ within o_A, and denote all the nodes, other than $X'_{(i)}s$ and $Y'_{(j)}s$ and located after the p_1 position, by $W_{(1)}, \ldots, W_{(n')}$ (the position of $W_{(l)}$ is ahead of that of $W_{(l')}$ in o_A if $l < l'$). We want to show that

Lemma 24 *There exists a topological ordering $o_{A'}$ of $\mathcal{G}$ that is at least as good as o_A, in terms of CATCH, such that* $\left(\star\right)$ Node B is located ahead of Node C in o_A if it is the case in o_{XY}, where B, C are two distinct nodes among $X'_{(i)}s$ and $Y'_{(i)}s$.

Lemma 24 implies that in order to figure out the optimal ordering of the original tree structure, it boils down to replacing the two branches $X_{(1)}, \ldots, X_{(m)}$ and $Y_{(1)}, \ldots, Y_{(m')}$ by a single chain characterized by o_{XY} and seeking the optimal ordering of the new structure. The ordering $o_{A'}$ mentioned above can be constructed easily with two constraints:

(i) Fixing the nodes located ahead of the position p_1 as well as their ordering as in o_A.

(ii) The position of $W_{(l)}$ is ahead of $W_{(l')}$ if $l < l'$, as in o_A.

The first constraint is straightforward, and the second constraint can be satisfied by applying Algorithm 1 to $X_{(1)}, \ldots, X_{(m)}$, $Y_{(1)}, \ldots, Y_{m'}$ and $W_{(1)}, \ldots, W_{(n')}$. Here, we take $W_{(1)}, \ldots, W_{(n')}$ as a chain, regardless of their original structure. Without loss of generality, we assume the

first chain branch with the maximal average score value figured out by Algorithm 1 is $X_{(1)}, \ldots, X_{(t)}$, $t \leq m$ (we can skip the case that Algorithm 1 first picks a part of $W_{(l)}$'s since it does not affect $\big(\star\big)$). In order to prove Lemma 24, it is reduced to showing that

Lemma 25 Conditional on (i) and (ii), the ordering with the maximal CATCH puts $X_{(1)}, \ldots, X_{(t)}$ at the position $p_1, \ldots, p_1 + t - 1$ respectively.

The detailed proof of Lemma 25 is deferred to Appendix A.1.1. Note that $X_{(1)}, \ldots, X_{(t)}$ must be located in the first place in the ordering o_{XY}. Therefore, by applying Lemma 25 in an inductive way (exclude $X_{(1)}, \ldots, X_{(t)}$ and apply the same argument on the remaining nodes), we can conclude the constructed $o_{A'}$ is at least as good as o_A and satisfies $\big(\star\big)$. Here, we need to clarify the point that putting $X_{(1)}, \ldots, X_{(t)}$ in such place does not violate the tree hierarchy since $X'_{(i)}s$ and $Y'_{(j)}s$ are the children chains of the same node, and none of $W'_{(l)}s$ can be an ancestor of $X'_{(i)}s$ or $Y'_{(j)}s$ (otherwise o_A is not a valid ordering). Thus the proof is completed. ■

A.1.1 Proof of Lemma 25

Proof
Let a denote the average of these t values, i.e., $a := \sum_{k=1}^{t} X_{(k)}$. For the sake of simplicity, we further assume $p_1 = 1$ and simply denote by $Z_{(1)}, \ldots, Z_{(n-t)}$ the combination of $X_{(t+1)}, \ldots, X_{(m)}$, $Y'_{(j)}s$ and $W'_{(l)}s$. Let the ordering o_A be as follows:

$$\begin{array}{cccccccccc} Z_{(1)} & \ldots & Z_{(i_1-1)} & X_{(1)} & Z_{(i_1)} & \ldots & Z_{(i_t-t)} & X_{(t)} & Z_{(i_t-t+1)} & \ldots & Z_{(n-t)} \\ 1 & \ldots & i_1-1 & i_1 & i_1+1 & \ldots & i_t-1 & i_t & i_t+1 & \ldots & n \end{array}$$

where i_c is the position of $X_{(c)}$, $c = 1, \ldots, t$. Note that $i_{c+1} \geq i_c + 1$. Denote by $o_{A'_1}$ the ordering of $(X_{(1)}, \ldots, X_{(t)}) + o_A/(X_{(1)}, \ldots, X_{(t)})$, that is, move $(X_{(1)}, \ldots, X_{(t)})$ to the head of o_A. The difference in the value of the objective function (OF) between $o_{A'_1}$ and o_A can be written as follows:

$$\begin{aligned} \text{OF of } o_{A'_1} =& \sum_{i=1}^{t}(n-i+1)X_{(i)} + \sum_{j=1}^{n-t}(n-t-j+1)Z_{(j)} \\ \text{OF of } o_A =& (n-i_1+1)X_{(1)} + \ldots + (n-i_t+1)X_{(t)} \\ &+ \sum_{j=1}^{i_1-1}(n-j+1)Z_{(j)} + \ldots + \sum_{j=i_t-t+1}^{n-t}(n-(j+t)+1)Z_{(j)} \end{aligned}$$

$$\begin{aligned}
\text{OF of } o_{A_1'} - \text{OF of } o_A &= \left[(i_1-1)X_{(1)} - \sum_{k=1}^{i_1-1} Z_{(k)}\right] + \ldots + \left[(i_t-t)X_{(t)} - \sum_{k=1}^{i_t-t} Z_{(k)}\right] \\
&= (i_1-1)\left[X_{(1)} - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Z_{(k)}\right] + \ldots + (i_t-t)\left[X_{(t)} - \frac{1}{i_t-t}\sum_{k=1}^{i_t-t} Z_{(k)}\right] \\
&= (i_1-1)\left[\big(X_{(1)}-a\big) + \left(a - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Z_{(k)}\right)\right] + \ldots \\
&\qquad + (i_t-t)\left[\big(X_{(t)}-a\big) + \left(a - \frac{1}{i_t-t}\sum_{k=1}^{i_t-t} Z_{(k)}\right)\right] \\
&= \big[(i_1-1)\big(X_{(1)}-a\big) + \ldots + (i_t-t)\big(X_{(t)}-a\big)\big] \\
&\qquad + \left[(i_1-1)\left(a - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Z_{(k)}\right) + (i_t-t)\left(a - \frac{1}{i_t-t}\sum_{k=1}^{i_t-t} Z_{(k)}\right)\right]
\end{aligned}$$

It remains to prove both the first term and the second term on the right side are nonnegative:

- *The first term.* We can rewrite the sum

$$(i_1-1)\big(X_{(1)}-a\big) + \ldots + (i_t-t)\big(X_{(t)}-a\big)$$

as follows

$$(i_1-1)\sum_{k=1}^{t}(X_{(k)}-a) \;+\; (i_2-i_1-1)\sum_{k=2}^{t}\big(X_{(k)}-a\big) + \ldots + (i_t-i_{t-1}-1)(X_{(t)}-a).$$

The first sum $\sum_{k=1}^{t}(X_{(k)}-a) = 0$ since a is the average. The other sums being nonnegative follows from the fact that a must be at least as large as the smaller averages in the chain, i.e. $a \geq \frac{1}{c}\sum_{k=1}^{c} X_{(k)}$ where $1 \leq c \leq t$. In detail, we know that

$$\begin{aligned}
X_{(c+1)} + \ldots + X_{(t)} &= ta - [X_{(1)} + \ldots + X_{(c)}], \quad 1 \leq c \leq t-1 \\
&\geq ta - ca = (t-c)a.
\end{aligned}$$

So $(X_{(c+1)}-a) + \ldots + (X_{(t)}-a) \geq 0$.

Therefore, each sum

$$\sum_{k=c}^{t} \big(X_{(k)} - a\big) \geq 0, \quad c = 1, \ldots, t. \tag{A.1.1}$$

It is clear that the expression

$$(i_1 - 1)\sum_{k=1}^{t}(X_{(k)} - a) \; + \; (i_2 - i_1 - 1)\sum_{k=2}^{t}\big(X_{(k)} - a\big) + \ldots + (i_t - i_{t-1} - 1)(X_{(t)} - a)$$

is exactly zero only when each $X_{(c)} = a$.

- *The second term.* We claim that each term in the expression

$$\left[(i_1 - 1)\left(a - \frac{1}{i_1 - 1}\sum_{k=1}^{i_1 - 1} Z_{(k)}\right) + (i_t - t)\left(a - \frac{1}{i_t - t}\sum_{k=1}^{i_t - t} Z_{(k)}\right)\right]$$

must be nonnegative, and equality holds only if there is a tie. To see this, we notice that $\sum_{k=1}^{i_c - c} Z_{(k)}$ can be separated as three sums: $\sum_{k=t+1}^{t_X} X_{(k)}$, $\sum_{k=1}^{t_Y} Y_{(k)}$ and $\sum_{k=1}^{t_W} W_{(k)}$, where $t_X \leq m$, $t_Y \leq m'$, $t_W \leq n'$ and $c = t_X - t + t_Y + t_W$. In terms of the procedure of Algorithm 1, it follows that

$$\sum_{k=1}^{t_Y} Y_{(k)} \leq t_Y \cdot a \qquad \text{and} \qquad \sum_{k=1}^{t_W} W_{(k)} \leq t_W \cdot a.$$

For the same reason, $\sum_{k=t+1}^{t_X} X_{(k)} \leq (t_X - t)a$, otherwise we have $\frac{1}{t_X}\sum_{k=1}^{t_X} X_{(k)} > a$ and it violates the condition that a is the average of the chain branch with the largest average score . So we have

$$\frac{1}{i_c - c}\sum_{k=1}^{i_c - c} Z_{(k)} = \frac{1}{i_c - c}[\sum_{k=t+1}^{t_X} X_{(k)} + \sum_{k=1}^{t_Y} Y_{(k)} + \sum_{k=1}^{t_W} W_{(k)}] \leq a.$$

■

A.2 Proof of Theorem 5

Proof To establish the bridge between HierRank (Algorithm 2) and Algorithm 4, we start from a simple case, i.e., a tree consisting of multiple chains with the same root (the root is in

$\mathcal{P}_2$). Specifically, denote by R the root and these children chain by $C_s := \{X_1^{(s)}, \ldots, X_{k_s}^{(s)}\}$, $s = 1, \ldots, \nu$, where ν is the number of chains, and k_s is the length of the s_{th} chain. Without loss of generality, suppose $T_1 := C_1(h_1) = \{X_1^{(1)}, \ldots, X_{h_1}^{(1)}\}$ is the first supernode that has been condensed to R, if R has not been taken off, or the first to be taken off after R. We claim that

Lemma 26 *When merging $C_1, \ldots, C_\nu$, Algorithm 1 puts T_1 in the first place.*

To show Lemma 26, we only need to show $\frac{1}{h_1}\sum_{k\in C_1(h_1)} S_k \geq \frac{1}{h}\sum_{k\in C_s(h)} S_k$, $\forall\, h \in \{1, \ldots, k_s\}$, $s \in \{1, \ldots, \nu\}$. The detailed proof is deferred to Appendix A.2.1. Inductively, it implies that the ordering given by Algorithm 4 on such simple case is the same as HierRank. Furthermore, any complicated structure boils down to the above simple case, since we can inductively merge the sub-chains starting from a root in $\mathcal{P}_2$ using Algorithm 1. This completes the proof showing that the results of HierRank and Algorithm 4 are the same. ■

A.2.1 Proof of Lemma 26

Proof We show the proof in three steps:

(i) Along the chain C_1, all the sub-chains starting from $X_1^{(1)}$ with larger length than T_1 have at most as large average score as T_1, that is, $\bar{\ell}_{1,h} \leq \bar{\ell}_{1,h_1}$, $\forall h_1 < h \leq k_1$. In terms of the procedure of Algorithm 4, all the mean score values in the supernodes following T_1 is no larger than ℓ_{1,h_1}.

(ii) Along the chain C_1, all the sub-chains starting from $X_1^{(1)}$ with smaller length than T_1 have at most as large average score as T_1, that is, $\bar{\ell}_{1,h} \leq \bar{\ell}_{1,h_1}$, $\forall 1 \leq h < h_1$. Otherwise, suppose $h_1' < h_1$ s.t. $X_1^{(1)}, \ldots, X_{h_1'}^{(1)}$ is the sub-chain with the largest average score and $\bar{\ell}_{1,h_1'} > \bar{\ell}_{1,h_1}$. By Eq. (A.1.1), we know that $\sum_{i=c}^{h_1'}(X_i^{(1)} - \ell_{1,h_1'}) \geq 0$, $c = 1, \ldots, h_1'$. So to make any supernode right behind the one ending with $X_{h_1'}^{(1)}$ merged with its former supernode, the average score value of this supernode must be at least $\ell_{1,h_1'}$. Thus, we can inductively conclude that $\bar{\ell}_{1,h_1} \geq \bar{\ell}_{1,h_1'}$, which is a contradiction.

(iii) $\bar{\ell}_{s,h} \leq \bar{\ell}_{1,h_1}$, $\forall h \in \{1, \ldots, k_s\}, s \in \{1, \ldots, \nu\}$. Otherwise, without loss of generality, suppose $X_1^{(2)}, \ldots, X_{h_2}^{(2)}$ is the sub-chain with the largest average score and $\bar{\ell}_{2,h_2} > \bar{\ell}_{1,h_1}$. By Eq. (A.1.1), any super node ending with $X_{h_2}^{(2)}$ has an average score of at least $\bar{\ell}_{2,h_2}$. Then it contradicts with the assumption that T_1 is the first supernode that will be merged with R, if R has not been taken off, or by the time T_1 is taken off.

■

Appendix B

Appendix for Chapter 3

B.1 Proof of Proposition 9

Proof Suppose

$$f(x) = 1.8 \cdot \mathbb{I}(x \in [0, 1/2]) + 0.2 \cdot \mathbb{I}(x \in (1/2, 1]).$$

Note that

$$\begin{aligned} & \mathbb{P}(s \le k/m) - \mathbb{P}(\hat{s} \le k/m) \\ = & \int_0^{k/m} f(u)du - \sum_{r \le k} \int_0^1 \binom{m}{r} u^r (1-u)^{m-r} f(u)du \\ = & \int_0^1 \left(\mathbb{I}[u \le k/m] - \sum_{r \le k} \binom{m}{r} u^r (1-u)^{m-r} \right) f(u)du \end{aligned} \tag{B.1.1}$$

Decompose $f(x) = f_1(x) + f_2(x)$, where $f_1(x) = 1.6 \cdot \mathbb{I}(x \in [0, 1/2])$, $f_2(x) \equiv 0.2$. The previous example shows that the difference for the f_2 part in Equation (B.1.1) is at most $\frac{0.2}{m+1}$. So we only need to take care of the f_1 part in Equation (B.1.1), i.e.,

$$1.6 \times \int_0^{1/2} \left(\mathbb{I}[u \le k/m] - \sum_{r \le k} \binom{m}{r} u^r (1-u)^{m-r} \right) du = 1.6 \times \int_0^{1/2} B_k(m, u)du,$$

provided $k/m \le 1/2$. Here $B_k(m, x) = \sum_{r=k}^m \binom{m}{k} x^r (1-x)^{m-r}$. Define

$$A_k(m, x) = [\binom{m}{k} x^k (1-x)^{m-k+1}] \cdot [(k+1)/(k+1-(m+1)x)].$$

Bahadur (1960)[Theorem 1] indicates that $1 \leq A_k(m,x)/B_k(m,x) \leq 1+z^{-2}$, where $z = (k-mx)/(mx(1-x))^{\frac{1}{2}}$. Let $x = 1/2-\epsilon$. By Stirling's formula and Taylor expansion on $\log(1+\epsilon)$, we can obtain

$$A_{m/2}(m, 1/2-\epsilon) \sim \frac{1}{\sqrt{2\pi m}} e^{-2m\epsilon^2},$$

and we can rewrite

$$z = \frac{\sqrt{m}\epsilon}{\sqrt{1/4-\epsilon^2}}.$$

So we have

$$B_{m/2}(m, 1/2-\epsilon) \geq \frac{A_{m/2}(m,1/2-\epsilon)}{1+z^{-2}} \sim \frac{2\sqrt{m}\epsilon e^{-2m\epsilon^2}}{4(m-1)\epsilon^2+1} \geq \frac{2\sqrt{m}\epsilon e^{-2m\epsilon^2}}{4m\epsilon^2+1}.$$

Then it follows that

$$\int_0^{1/2} B_{m/2}(m,1/2-\epsilon)d\epsilon \geq \int_0^{1/\sqrt{m}} \frac{2\sqrt{m}\epsilon e^{-2m\epsilon^2}}{4m\epsilon^2+1} d\epsilon = \frac{1}{\sqrt{m}} \int_0^1 \frac{2e^{-2u^2}}{4u^2+1} du \approx \frac{0.81}{\sqrt{m}}.$$

Together, we have $\mathbb{P}(s \leq 1/2) - \mathbb{P}(\hat{s} \leq 1/2) \geq \frac{C}{\sqrt{m}} + \varepsilon \cdot m^{-1}$, where ε is a residual term with $|\varepsilon| \leq K$, C and K are positive constants. ■

B.2 Proof of Proposition 12

Proof The proof of this theorem follows from that of Theorem 13, so we defer most details to the proof of the latter.

From Equation (B.3.5), we know that

$$\begin{aligned}
& \mathbb{P}(\hat{s}_1 \in B(x)) - \mathbb{P}(s_1 \in B(x)) \\
= & \sum_{d=1}^{D} [\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x+d\cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x+d\cdot h))] \\
& + \sum_{d=1}^{D} [\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x-d\cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x-d\cdot h))] \\
& + \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x+d\cdot h) : d \geq D+1 \text{ or } d \leq -D-1) \\
& + \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x+d\cdot h) : d \geq D+1 \text{ or } d \leq -D-1),
\end{aligned}$$

where $B(x)$ is still defined as the bin that contains x among $[0,h],(h,2h],\ldots,(1-h,1]$. Note that

$$\begin{aligned}
& \mathbb{P}(\hat{s}_1 \le x) - \mathbb{P}(s_1 \le x) \\
= \;& \mathbb{P}(\hat{s}_1 \le x, s_1 > x) - \mathbb{P}(\hat{s}_1 > x, s_1 \le x) \\
= \;& \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 \le x, s_1 > x) - \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 > x, s_1 \le x) \\
& + \sum_{\substack{d=1,2,\ldots \\ d'=1,2,\ldots}} [\mathbb{P}(\hat{s}_1 \in B(x-dh), s_1 \in B(x+d'h)) - \mathbb{P}(\hat{s}_1 \in B(x+dh), s_1 \in B(x-d'h))]. \\
= \;& \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 \le x, s_1 > x) - \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 > x, s_1 \le x) \\
& + (\sum_{|d-d'|\le D} + \sum_{|d-d'|>D})[\mathbb{P}(\hat{s}_1 \in B(x-dh), s_1 \in B(x+d'h)) \\
& \qquad - \mathbb{P}(\hat{s}_1 \in B(x+dh), s_1 \in B(x-d'h))].
\end{aligned}$$

Following the proof of bounding $\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x+d\cdot h) : d \ge D+1 \text{ or } d \le -D-1)$ in Theorem 13, we can bound

$$\begin{aligned}
& |\sum_{|d-d'|>D} [\mathbb{P}(\hat{s}_1 \in B(x-dh), s_1 \in B(x+d'h)) - \mathbb{P}(\hat{s}_1 \in B(x+dh), s_1 \in B(x-d'h))| \\
\le \;& 2f_{\max} \cdot \exp(-2mD^2h^2) = \frac{2f_{\max}}{m},
\end{aligned}$$

where $D = \lceil \sqrt{\frac{\log m}{2mh^2}} \rceil$.
Following the proof of bounding $|\sum_{d=1}^{D}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x+d\cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x+d\cdot h))]|$ in Theorem 13, we can bound

$$\begin{aligned}
& |\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 \le x, s_1 > x) - \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x), \hat{s}_1 > x, s_1 \le x)| \\
\le \;& K_1 \cdot (\frac{f_{\max}\cdot h}{\sqrt{m}} + \frac{f_{\max}}{m} + \frac{h\cdot f'_{\max}}{\sqrt{m}}) + |\mathcal{E}|,
\end{aligned}$$

where $|\mathcal{E}| \le K_2 \cdot (\frac{h\cdot f_{\max}}{\sqrt{m}} + h^2 f_{\max} + \frac{f_{\max}}{m})$ for some constant K_2 that only depends on a.
To bound

$$|\sum_{|d-d'|\le D} [\mathbb{P}(\hat{s}_1 \in B(x-dh), s_1 \in B(x+d'h)) - \mathbb{P}(\hat{s}_1 \in B(x+dh), s_1 \in B(x-d'h))|,$$

we still follow the proof of bounding $|\sum_{d=1}^{D}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x+d\cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x+d\cdot h))]|$ in Theorem 13. The problem is that there are D^2 terms instead D terms. By

carefully arranging the D^2 terms, and using the fact that $\sum_{d=1}^{\infty}\exp(-C_0d^2)<\infty$ for any positive C_0, it can be easily seen that we still get the same bound as above. In total, we have

$$|\mathbb{P}(\hat{s}_1\le x)-\mathbb{P}(s_1\le x)|\le K_3\cdot(f_{\max}+f'_{\max})\cdot(\frac{h}{\sqrt{m}}+h^2+\frac{1}{m}),$$

where K_3 is some constant that only hinges on a. To minimize the upper bound, take $h=\frac{1}{\sqrt{m}}$. Thus, it follows that

$$\sup_{x\in[a,1-a]}|F^{(m)}(x)-F(x)|\le\frac{C}{m},$$

where C is some constant that only depends on f and a. ■

B.3 Proof of Theorem 13

Theorem 13 relies on the the local limit theorem of binomial distribution, as follows.

Lemma 27 *Suppose $X\sim Binom(m,s)$, with $0<s<1$. For any $a<b$ such that $\Delta:=\max\{|\frac{a}{m}-s|,|\frac{b}{m}-s|\}\to 0$ as $m\to\infty$, then*

$$\mathbb{P}(a\le X\le b)=\left[\int_{\frac{a-ms}{\sqrt{ms(1-s)}}}^{\frac{b-ms}{\sqrt{ms(1-s)}}}\frac{1}{\sqrt{2\pi}}\exp(-t^2/2)dt\right](1+\varepsilon_1)+\varepsilon_2,$$

where ε_1 is the error from the Gaussian approximation to Binomial point mass function, and ε_2 is the error from the summation series approximating the integral. Specifically, we have

$$|\varepsilon_1|\le K\cdot\Delta,$$

$$|\varepsilon_2|\le C\cdot[\frac{\exp(-m\delta^2)}{\sqrt{m}}+\frac{b-a}{\sqrt{m}}\cdot\Delta\cdot\exp(-m\delta^2)],$$

where $\delta:=\min_{x\in[\frac{a}{m},\frac{b}{m}]}|s-x|$, K and C are two positive constants that depend on s via $\frac{1}{s(1-s)}$.

The detailed proof of Lemma 27 can be easily obtained by adapting that of Dunbar (2011). Now we prove Theorem 13:

Proof Denote by $R_x=\mathbb{E}(f(x)-\hat{f}_{n,m}(x))^2$ the risk at a point x. We decompose the risk into the variance and the bias square as follows.

$$R_x=var(\hat{f}_{n,m}(x))+(\mathbb{E}\hat{f}_{n,m}(x)-f(x))^2. \tag{B.3.1}$$

For the variance part, denote by $p_l = \mathbb{P}(m \cdot \hat{s}_1 \in B_l)$. We have

$$\int_a^{1-a} var(\hat{f}_{n,m}(x))dx = \sum_{aL \le l \le (1-a)L} \int_{B_l} \frac{p_l(1-p_l)}{nh^2} \le K_1 \cdot \frac{1}{nh}, \tag{B.3.2}$$

where K_1 is a positive constant that relies on $a > 0$. For the bias part, since

$$(\mathbb{E}\hat{f}_{n,m}(x) - f(x))^2 \le 2(\mathbb{E}\hat{f}_{n,m}(x) - \mathbb{E}\hat{f}_n(x))^2 + 2(\mathbb{E}\hat{f}_n(x) - f(x))^2, \tag{B.3.3}$$

and it is well known that

$$\int_0^1 (\mathbb{E}\hat{f}_n(x) - f(x))^2 dx \le K_2 \cdot h^2, \tag{B.3.4}$$

where K_2 is positive constant that only relies on f. We only need to consider $\mathbb{E}\hat{f}_{n,m}(x) - \mathbb{E}\hat{f}_n(x)$. By definition,

$$\begin{aligned}
& \mathbb{E}\hat{f}_{n,m}(x) - \mathbb{E}\hat{f}_n(x) \\
= \; & \frac{1}{h}[\mathbb{P}(\hat{s}_1 \in B(x)) - \mathbb{P}(s_1 \in B(x))] \\
= \; & \frac{1}{h}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \notin B(x)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \notin B(x))] \\
= \; & \frac{1}{h}\{\sum_{d=1}^{D}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h))] \\
& + \sum_{d=1}^{D}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x - d \cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x - d \cdot h))] \\
& + \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h) : d \ge D + 1 \text{ or } d \le -D - 1) \\
& + \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h) : d \ge D + 1 \text{ or } d \le -D - 1)\}.
\end{aligned} \tag{B.3.5}$$

By McDiarmid's inequality, it follows that

$$\begin{aligned}
& \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h) : d \ge D + 1 \text{ or } d \le -D - 1) \\
= \; & \int_{s_1 \in B(x)} [\mathbb{E}\mathbb{I}(\hat{s}_1 \in B(x + d \cdot h) : d \ge D + 1 \text{ or } d \le -D - 1 | s_1)] f(s_1) ds_1 \\
\le \; & \int_{s_1 \in B(x)} \mathbb{P}(|\hat{s}_1 - s_1| \ge D \cdot h | s_1) f(s_1) ds_1 \\
\le \; & \int_{s_1 \in B(x)} \exp(-2mD^2h^2) f(s_1) ds_1 \\
\le \; & f_{\max} \cdot h \cdot \exp(-2mD^2h^2),
\end{aligned}$$

where $f_{\max}$ is the maximal value of f in $[0,1]$. Take $D = \lceil \sqrt{\frac{\log m}{4mh^2}} \rceil$, i.e., the least integer that is not smaller than $\sqrt{\frac{\log m}{4mh^2}}$. Then we have

$$\mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h) : d \geq D + 1 \text{ or } d \leq -D - 1) \leq K_3 \cdot \frac{f_{\max} \cdot h}{\sqrt{m}}, \tag{B.3.6}$$

where K_3 is a universal positive constant. Similarly, we can show that $\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h) : d \geq D + 1 \text{ or } d \leq -D - 1) \leq K_4 \cdot \frac{f_{\max}}{m}$ for some positive constant K_4. Next, we investigate $\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h))$. Denote by $l(x)$ and $r(x)$ the left boundary and the right boundary of the interval $B(x)$. By Lemma 27, it follows that

$$\begin{aligned}
& \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h)) \\
= & \int_{s_1 \in B(x)} \left[\sum_{k: \frac{k}{m} \in B(x + d \cdot h)} \binom{m}{k} (s_1)^k (1 - s_1)^{m-k} \right] f(s_1) ds_1 \\
= & \int_{s \in B(x)} \Bigg\{ \underbrace{\left[\int_{t \in B(x + d \cdot h)} \frac{\sqrt{m}}{\sqrt{2\pi s(1-s)}} \exp(-\frac{m(t-s)^2}{2s(1-s)}) dt \right] \cdot (1 + \varepsilon_5 \cdot (d+1)h)}_{\text{(I)}} \\
& \qquad + \underbrace{\varepsilon_6 \cdot \frac{\exp(-m[r(x) + (d-1)h - s]^2)}{\sqrt{m}}}_{\text{(II)}} \\
& \qquad + \underbrace{\varepsilon_7 \cdot \sqrt{m} h \cdot [r(x) + dh - s] \cdot \exp(-m[r(x) + (d-1)h - s]^2}_{\text{(III)}} \Bigg\} f(s) ds,
\end{aligned}$$

where $|\varepsilon_5| \leq K_5$, $|\varepsilon_6| \leq K_6$, $|\varepsilon_7| \leq K_7$ and K_5, K_6, K_7 are positive constants that only depend on a. We consider the summation of the D error terms in $\mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h))$, $d = 1, \ldots, D$:

(I)

$$
\begin{aligned}
& \sum_{d=1}^{D} \int_{s\in B(x)} \left[\int_{t\in B(x+d\cdot h)} \frac{\sqrt{m}}{\sqrt{2\pi s(1-s)}} e - \frac{m(t-s)^2}{2s(1-s)} dt \right] K_5 (d+1) h f(s) ds \\
\leq\ & K_8 h^2 f_{\max} + \sum_{d=3}^{D} K_9 \int_{s\in B(x)} \left[\int_{t\in B(x+d\cdot h)} \sqrt{m}(d+1)h \cdot e^{-\frac{m(t-s)^2}{2s(1-s)}} dt \right] f(s) ds \\
\leq\ & K_8 h^2 f_{\max} + \sum_{d=3}^{D} K_9 \int_{s\in B(x)} \left[\int_{t\in B(x+d\cdot h)} \sqrt{m}(d+1)h \cdot e^{-m(t-s)^2} dt \right] f(s) ds \\
\leq\ & K_8 h^2 f_{\max} + \sum_{d=3}^{D} K_9 \int_{s\in B(x)} \left[\int_{t\in B(x+d\cdot h)} 2\sqrt{m}(d-1)h \cdot e^{-m(t-s)^2} dt \right] f(s) ds \\
\leq\ & K_8 h^2 f_{\max} + \sum_{d=3}^{D} K_9 \int_{s\in B(x)} \left[\int_{t\in B(x+d\cdot h)} 2\sqrt{m}(t-s) \cdot e^{-m(t-s)^2} dt \right] f(s) ds \\
=\ & K_8 h^2 f_{\max} \\
& + \sum_{d=3}^{D} \frac{2K_9}{\sqrt{m}} \int_{s\in B(x)} \left[e^{-m(r(x)-s+(d-1)h)^2} - e^{-m(r(x)-s+dh)^2} \right] f(s) ds \\
=\ & K_8 h^2 f_{\max} + \tilde{K}_9 \frac{h f_{\max}}{\sqrt{m}},
\end{aligned}
$$

where $K_8, K_9, \tilde{K}_9$ are positive constants that only depend on a.

(II)

$$
\begin{aligned}
& \sum_{d=1}^{D} \int_{s\in B(x)} K_6 \cdot \frac{\exp(-m[r(x)+(d-1)h-s]^2)}{\sqrt{m}} \cdot f(s) ds \\
\leq\ & K_6 \cdot \frac{f_{\max}}{\sqrt{m}} \cdot \int_0^{(D-1)h} \exp(-mt^2) dt \\
\leq\ & \tilde{K}_6 \cdot \frac{f_{\max}}{m},
\end{aligned}
$$

where $\tilde{K}_6$ is a positive constant that only depends on a.

(III)

$$\begin{aligned}
& \sum_{d=1}^{D} \int_{s\in B(x)} K_7 \cdot \sqrt{m}h \cdot [r(x) + dh - s] \cdot e^{-m[r(x)+(d-1)h-s]^2} \cdot f(s)ds \\
= & \sum_{d=1}^{D} \int_{s\in B(x)} K_7 \cdot \sqrt{m}h \cdot [r(x) + (d-1)h - s + h] \cdot e^{-m[r(x)+(d-1)h-s]^2} \cdot f(s)ds \\
\leq & \tilde{K}_7 \cdot \left\{ h^2 \cdot f_{\max} + \frac{h \cdot f_{\max}}{\sqrt{m}} \cdot \sum_{d=1}^{D} [e^{-m(d-1)^2h^2} - e^{-md^2h^2}] \right\} \\
= & \tilde{K}_7 \cdot \left\{ h^2 \cdot f_{\max} + \frac{h \cdot f_{\max}}{\sqrt{m}} \cdot [1 - e^{-mD^2h^2}] \right\} \\
\leq & \tilde{K}_7 \cdot (h^2 \cdot f_{\max} + \frac{h \cdot f_{\max}}{\sqrt{m}}),
\end{aligned}$$

where $\tilde{K}_7 > 0$ only depends on a.

We call the summation of the D error terms by $\mathcal{E}$, which satisfies $|\mathcal{E}| \leq K_{10} \cdot (\frac{h \cdot f_{\max}}{\sqrt{m}} + h^2 f_{\max} + \frac{f_{\max}}{m})$, where $K_{10} > 0$ only depends on a. Similarly, for the summation of the D error terms

$\tilde{\mathcal{E}}$ in $\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h))$, $d = 1, \ldots, D$, we have the same rate. Now we consider

$$
\begin{aligned}
& |\sum_{d=1}^{D} \mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x + d \cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x + d \cdot h))| \\
\overset{(i)}{=} & \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| \frac{\sqrt{m}}{\sqrt{2\pi s(1-s)}} e^{-\frac{m(t-s)^2}{2s(1-s)}} f(s) \right. \\
& \qquad \left. - \frac{\sqrt{m}}{\sqrt{2\pi t(1-t)}} e^{-\frac{m(s-t)^2}{2t(1-t)}} f(t) \right| dtds + |\mathcal{E}| + |\tilde{\mathcal{E}}| \\
\overset{(ii)}{\leq} & \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| \frac{|2\tilde{s} - 1|}{(1-\tilde{s})^{\frac{5}{2}} \tilde{s}^{\frac{5}{2}}} \cdot \frac{m^{\frac{3}{2}}(t-s)^3}{2\sqrt{2\pi}} e^{-\frac{m(t-s)^2}{2\tilde{s}(1-\tilde{s})}} f(\tilde{s}) \right| dtds \\
& + \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| \frac{|2\tilde{s} - 1|}{(1-\tilde{s})^{\frac{3}{2}} \tilde{s}^{\frac{3}{2}}} \cdot \frac{m^{\frac{1}{2}}(t-s)}{2\sqrt{2\pi}} e^{-\frac{m(t-s)^2}{2\tilde{s}(1-\tilde{s})}} f(\tilde{s}) \right| dtds \\
& + \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| \frac{1}{(1-\tilde{s})^{\frac{1}{2}} \tilde{s}^{\frac{1}{2}}} \cdot \frac{m^{\frac{1}{2}}(t-s)}{\sqrt{2\pi}} e^{-\frac{m(t-s)^2}{2\tilde{s}(1-\tilde{s})}} f'(\tilde{s}) \right| dtds + |\mathcal{E}| + |\tilde{\mathcal{E}}| \\
\overset{(iii)}{\leq} & \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| K_{11} \cdot f_{\max} \cdot m^{\frac{3}{2}}(t-s)^3 e^{-m(t-s)^2} \right| dtds \\
& + \sum_{d=1}^{D} \int_{s \in B(x)} \int_{t \in B(x+d \cdot h)} \left| K_{12} \cdot (f_{\max} + f'_{\max}) \cdot m^{\frac{1}{2}}(t-s) e^{-m(t-s)^2} \right| dtds + |\mathcal{E}| + |\tilde{\mathcal{E}}| \\
\overset{(iv)}{=} & K_{13} \cdot \left(\frac{f_{\max} \cdot h}{\sqrt{m}} + \frac{f_{\max}}{m} + \frac{h \cdot f'_{\max}}{\sqrt{m}}\right) + |\mathcal{E}| + |\tilde{\mathcal{E}}|, \qquad \text{(B.3.7)}
\end{aligned}
$$

where K_{11}, K_{12}, K_{13} are positive constants that only depend on a. Equation (i) uses the Fubini's theorem; Inequality (ii) applies the mean value theorem to the function $g(s) = \frac{1}{s(1-s)} \exp(-\frac{A}{2s(1-s)}) f(s)$, where A is a constant; Inequality (iii) holds since $0 < a \leq s \leq \tilde{s} \leq t \leq 1 - a < 1$, thus $\frac{1}{(1-\tilde{s})\tilde{s}}$ is bounded, and $\exp(-\frac{m(t-s)^2}{2\tilde{s}(1-\tilde{s})})$ attains the maximal when $\tilde{s} = 1/2$; Inequality (iv) is obtained via integral by part. Similarly, $\sum_{d=1}^{D}[\mathbb{P}(\hat{s}_1 \in B(x), s_1 \in B(x - d \cdot h)) - \mathbb{P}(s_1 \in B(x), \hat{s}_1 \in B(x - d \cdot h))]$ has the same rate as (B.3.7). Putting (B.3.5)(B.3.6)(B.3.7) together, we have

$$
|\mathbb{E}\hat{f}_{n,m}(x) - \mathbb{E}\hat{f}_n(x)| \leq K_{14} \cdot (f_{\max} + f'_{\max}) \cdot \left(\frac{1}{\sqrt{m}} + h + \frac{1}{mh}\right), \qquad \text{(B.3.8)}
$$

where K_{14} is some constant that only depends on a. Combining Inequalities (B.3.1)(B.3.2) (B.3.3)(B.3.4) and Inequality (B.3.8), it follows that

$$R(a, 1-a) \leq C_1 \cdot (h^2 + \frac{1}{m} + \frac{1}{m^2 h^2} + \frac{1}{nh}),$$

The minimal risk is no larger than $C_4 \cdot n^{-\frac{2}{3}}$, which is attained when $h = C_3 \cdot n^{-\frac{1}{3}}$, $m \geq C_2 \cdot n^{\frac{2}{3}}$. Here C_1, C_2, C_3, C_4 are positive constants that only depend on a and f. ■

B.4 Proof of Theorem 14

Proof Note for the point mass function (3.3.1), we have an additional information that only hold for the discrete case but not for the density case,

$$\max_{x \in I_{k+d}} |x - x_k| = (d + \frac{1}{2}) \cdot \frac{1}{K}.$$

We follow the proof of Theorem 13 and can show that

$$R(a, 1-a) \leq C_1 \cdot (\frac{1}{n} + \frac{1}{m} + \frac{K^2}{m^2}).$$

When $m \geq C_2 \cdot \sqrt{n} \max\{K, \sqrt{n}\}$, we have $R(a, 1-a) \leq C_3 \cdot \frac{1}{n}$. Here where $C_1, C_2, C_3 > 0$ do not depend on n, m and K. ■

B.5 Proof of Theorem 19

We first define a few concepts. Let J denote the interval $[a, b)$, where $a = 0$ and $b = 1$ in our setup. We set

$$l(J) = b - a, f(J) = \int_J f(t)dt, \Delta f(J) = f(b) - f(a),$$

$$bf(J) = \int_J |f(J)/l(J) - f(t)|dt.$$

Any finite increasing sequences $\{x_i\}_{0 \leq x_i \leq q}$ with $x_0 = a$, $x_q = b$ generates a partition $\mathcal{P}$ of J into intervals $J_i = [x_{i-1}, x_i)$, $1 \leq i \leq q$. When no confusion arises, we put f_i for $f(J_i)$, Δf_i for $\Delta f(J_i)$ and so on. Set a functional $L(\mathcal{P}, f, z)$ defined for positive z by

$$L(\mathcal{P}, f, z) = \sum_{i=1}^{q} [bf(J_i) + z(f(J_i))^{1/2}] = \sum_{i=1}^{q} [bf_i + z f_i^{1/2}].$$

Before proving Theorem 19, we state the needed lemma, which is adapted from Lemma 1 in Birge (1989). The proof also follows that of Lemma 1 in Birge (1989).

Lemma 28 *Let $\mathcal{P} = \{J_i\}_{1\leq i\leq q}$ be some partition of J, F an absolutely continuous distribution function, $F_{n,m}$ the corresponding empirical c.d.f based on $\hat{s}_i$ and $\tilde{F}_{n,m} = \tilde{F}^J_{n,m}$, $\tilde{F}^i_{n,m} = \tilde{F}^{J_i}_{n,m}$ the related Grenander estimators defined on the associative intervals. Define*

$$\bar{F}_{n,m}(x) = \sum_{i=1}^{q} \tilde{F}^i_{n,m}(x)\mathbb{I}_{x\in J_i},$$

with f and $\bar{f}_{n,m}$ to be the respective derivatives of F and $\bar{F}_{n,m}$. Then

$$\mathbb{E}\left[\int_J |\tilde{f}_{n,m}(x) - f(x)|dx\right] \leq \mathbb{E}\left[\int_J |\bar{f}_{n,m}(x) - f(x)|dx\right].$$

In the proof, there are many similar notations that might be confusing. I list all of them below for clarity and the convenience of reference.

- Let $F^{(m)}(x) := \mathbb{P}(\hat{s}_i \leq x)$, i.e. the c.d.f of $\hat{s}_i$. Let $f^{(m)}$ be the derivative of $F_{n,m}$.
- For any interval J, $F^J_{n,m}(x) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}(\hat{s}_i \leq x; \hat{s}_i \in J)$, $F^J_n(x) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}(s_i \leq x; s_i \in J)$. $F_{n,m}$ and F_n correspond to $F^J_{n,m}(x)$ and $F^J_n(x)$ with $J = [0,1]$ in our setup.
- $\tilde{F}^J_{n,m}$ and $\tilde{F}^J_n$ are the respective least concave majorants of $F^J_{n,m}$ and F^J_n condition on the interval J. Let $\tilde{f}^J_{n,m}$ and $\tilde{f}^J_n$ be the derivatives of $\tilde{F}^J_{n,m}$ and $\tilde{F}^J_n$. $\tilde{F}_{n,m}$, $\tilde{F}_n$ and $\tilde{f}_{n,m}$, $\tilde{f}_n$ correspond to $\tilde{F}^J_{n,m}(x)$, $\tilde{F}^J_n(x)$ and $\tilde{f}^J_{n,m}$, $\tilde{f}^J_n$.
- For any partition $\mathcal{P} = \{J_i\}_{1\leq i\leq q}$, let $\bar{f}_{n,m}$ be the derivative of $\bar{F}_{n,m}(x) = \sum_{i=1}^q \mathbb{I}(x \in J_i)\tilde{F}^{J_i}_{n,m}(x)$.
- $l(J) = b - a$, $f(J) = \int_J f(t)dt$, $\Delta f(J) = f(b) - f(a)$, $bf(J) = \int_J |f(J)/l(J) - f(t)|dt$.
- For any partition $\mathcal{P} = \{J_i\}_{1\leq i\leq q}$, $L^J(\mathcal{P}, f, z) = \sum_{i=1}^q [bf(J_i) + z(f(J_i))^{1/2}] = \sum_{i=1}^q [bf_i + zf_i^{1/2}]$. $L^J(f, z) = \inf_{\mathcal{P}} L^J(\mathcal{P}, f, z)$.
- $M := \int_0^b f^p(t)dt$ for some $p > 2$; $H := \lim_{x\to b-} f(x)$.
- For an interval $I := [a^I, b^I)$,
  - Let N and N_m be the number of s_i's and $\hat{s}_i$'s that fall in I, respectively.
  - Define G and $G^{(m)}$ to be the respective conditional c.d.f's of the s_i's and $\hat{s}_i$'s that fall in I, g and $g^{(m)}$ their derivatives.

– Define $G_N(x) = \sum_{i=1}^n \mathbb{I}[s_i \leq x; s_i \in I]$, $G_{N_m}(x) := \sum_{i=1}^n \mathbb{I}[\hat{s}_i \leq x; \hat{s}_i \in I]$.
– Define $\tilde{G}_{N_m,m}$ and $\tilde{G}_N$ be the respective least concave majorants of G_{N_m} and G_N conditional on I. Let $\tilde{g}_{N_m,m}$ and $\tilde{g}_N$ be derivatives of $\tilde{G}_{N_m,m}$ and $\tilde{G}_N$ respectively.

Proof

We want to show that

$$\mathbb{E}_f\left[\int_J |f(x) - \tilde{f}_{n,m}(x)|dx\right] \leq 3L^J(f, \tilde{K}\cdot n^{-\frac{1}{2}} + \tilde{C}\cdot m^{-\frac{1}{2}}),$$

where $\tilde{K}, \tilde{C}$ are universal constants. Then, the L_1 convergence of $\tilde{f}_{n,m}$ only hinges on the characteristics of f. For example, when f is a decreasing function on $J = [0,1]$ such that $M := \int_0^b f^p(t)dt < +\infty$ for some $p > 2$ and $H = \lim_{x\to b-} f(x) > 0$, then Proposition 4 of Birge (1989) shows that

$$z^{-2/3}L^J(f,z) \leq 3/2(H/(H-h))^{p-1}(bMH^{2-p}/(p-2))^{1/3}, \tag{B.5.1}$$

where $h^3 = z^2b^{-2}MH^{2-p}/(p-2)$ and $Mz^2 < (p-2)b^2H^{p+1}$. It implies that $\tilde{f}_{n,m}$ has an L_1 convergence rate at $L^J(f, \tilde{K}\cdot n^{-\frac{1}{2}} + \tilde{C}\cdot m^{-\frac{1}{2}}) \leq (C_1\cdot n^{-\frac{1}{2}} + C_2\cdot m^{-\frac{1}{2}})^{2/3}$, where C_1, C_2 are some positive constants.

By Lemma 28 it is sufficient to prove that for any partition $\mathcal{P} = \{J_i\}_{1\leq i\leq q}$ of J, we have

$$\mathbb{E}_f\left[\int_J |f(x) - \bar{f}_{n,m}(x)|dx\right] \leq 3\sum_{i=1}^q [bf_i + \sqrt{f_i}\cdot(\tilde{K}\cdot n^{-\frac{1}{2}} + \tilde{C}\cdot m^{-\frac{1}{2}})],$$

where $\bar{f}_{n,m}$ is the derivative of $\bar{F}_{n,m}(x) = \sum_{i=1}^q \tilde{F}_{n,m}^{J_i}(x)\mathbb{I}(x \in J_i)$. This is certainly true if for any arbitrary sub-interval $I = [a^I, b^I)$ of J, the below inequality holds

$$\mathbb{E}_f\left[\int_I |f(x) - \tilde{f}_{n,m}^I|dx\right] \leq 3\left[bf(I) + \sqrt{f(I)}\cdot(\tilde{K}\cdot n^{-\frac{1}{2}} + \tilde{C}\cdot m^{-\frac{1}{2}})\right].$$

In order to prove this inequality, we assume there are N s_i's and N_m $\hat{s}_i$'s falling in the interval I respectively. Here, N has a binomial distribution $Binomial(n, f(I))$ and N_m has a binomial distribution $Binomial(n, f^{(m)}(I))$, where $f^{(m)}$ is the derivative of the c.d.f $F^{(m)} = \mathbb{P}[\hat{s}_i \leq x]$. Then with $\tilde{f}_{n,m} = \tilde{f}_{n,m}^I$,

$$\int_I |f(x) - \tilde{f}_{n,m}(x)|dx \leq bf(I) + |f(I) - N/n| + |N/n - N_m/n| + b\tilde{f}_{n,m}(I). \tag{B.5.2}$$

The only difficulty comes from the last term. Define G and $G^{(m)}$ to be the respective conditional c.d.f's of the s_i's and $\hat{s}_i$'s that fall in I, g and $g^{(m)}$ their derivatives. Then

$$\mathbb{E}_{f^{(m)}}[b\tilde{f}_{n,m}(I)|N_m] = N_m/n\mathbb{E}_{g^{(m)}}[b\tilde{g}_{N_m,m}(I)|N_m]$$

because the joint distribution of the N_m $\hat{s}_i$'s falling in I given N_m is the same as the distribution of N_m i.i.d variables from $G^{(m)}$. If $U(x)$ is the uniform c.d.f on I, then

$$\begin{aligned} 1/2b\tilde{g}_{N_m,m}(I) &\overset{(a)}{=} \sup_{x\in I}[\tilde{G}_{N_m,m}(x) - U(x)] \\ &\overset{(b)}{=} \sup_{x\in I}[G_{N_m}(x) - U(x)] \\ &\leq \sup_{x\in I}[G_{N_m}(x) - G(x)] + \sup_{x\in I}[G(x) - U(x)] \\ &\leq \sup_{x\in I}[G_{N_m}(x) - G(x)] + 1/2bg(I). \end{aligned}$$

Here Equation (a) holds because this is an equivalent expression of the total variation for $\tilde{G}_{N_m,m}(x)$ with a non-increasing derivative and $U(x)$ with a flat density. Equation (b) holds because

- $\tilde{G}_{N_m,m}(x) \geq G_{N_m}(x)$ for any x and the equality occurs when the derivative of $\tilde{G}_{N_m,m}$ changes.
- $\tilde{G}_{N_m,m}(x) - U(x)$ attains the maximum at a point which corresponds to a change of the derivative of $\tilde{G}_{N_m,m}$.

Since $bf(I) = f(I)bg(I)$, we get

$$\mathbb{E}_{f^{(m)}}[b\tilde{f}_{n,m}(I)|N_m] \leq N_m/n \left[2\mathbb{E}_{g^{(m)}}[\sup_{x\in I}(G_{N_m(x)} - G(x))|N_m] + bf(I)/f(I)\right],$$

and using Corollary 11,

$$\mathbb{E}_{f^{(m)}}[b\tilde{f}_{n,m}(I)|N_m] \leq N_m/n \left[K \cdot (\frac{1}{\sqrt{N_m}} + \frac{1}{\sqrt{m}}) + bf(I)/f(I)\right],$$

where $K > 1$. Plug in this result into the inequality (B.5.2), and with the Cauchy-Schwarz inequality, we have

$$\begin{aligned} &\mathbb{E}_f \int_I |f(x) - \tilde{f}_{n,m}(x)|dx \\ \leq\ & bf(I) + \mathbb{E}|f(I) - N/n| \\ &+K\sqrt{f^{(m)}(I)/n} + K \cdot f^{(m)}/\sqrt{m} + bf(I) \cdot f^{(m)}(I)/f(I) + \mathbb{E}|N/n - N_m/n|. \end{aligned}$$

By the Cauchy-Schwarz inequality, it follows that

$$\mathbb{E}|f(I) - N/n| \leq \sqrt{\frac{f(I)(1-f(I))}{n}}.$$

Note that

$$\begin{aligned} f^{(m)}(I) &= \mathbb{E}N_m/n \\ &= \mathbb{E}N/n + \mathbb{E}[N_m - N]/n \\ &\leq f(I) + \mathbb{E}|N_m - N|/n \\ &= f(I) + \mathbb{E}[\mathbb{E}[|N_m - N|/n|N]] \\ &\overset{(c)}{\leq} f(I) + \mathbb{E}[(\frac{C}{\sqrt{m}}) \cdot N/n] \\ &= f(I)(1 + C \cdot m^{-\frac{1}{2}}) \end{aligned}$$

where the Inequality (c) holds because of Proposition8 with $C > 0$ (note that $N_m/n = F_{n,m}(b^I) - F_{n,m}(a^I)$, and $N/n = F_n(b^I) - F_n(a^I)$). Thus, for $m \geq C$, it follows that

$$\mathbb{E}_f \int_I |f(x) - \tilde{f}_{n,m}(x)|dx \leq 3[bf(I) + \sqrt{f(I)} \cdot (\frac{\tilde{K}}{\sqrt{n}} + \frac{\tilde{C}}{\sqrt{m}})],$$

where $\tilde{K}$ and $\tilde{C}$ are positive constants. Finally, as Proposition 4 in Birge (1989), we construct the partition $\mathcal{P} = \{J_i\}_{j \leq i \leq q}$ of J where j is the integer such that $jh \leq H < (j+1)h$, $J_q = \{x|f(x) \geq q\}$, $J_j = \{x|f(x) < (j+1)h\}$, and $J_i = \{x|ih \leq f(x) < (i+1)h\}$ for $q > i > j$. Since $f_{\max} < \infty$, there is only finite number of intervals. It can be shown that when q is the smallest integer that is larger than $f_{\max}$, this partition can give the inequality (B.5.1). ■

B.6 Proof of Theorem 20

B.6.1 Local Inference of $F_{n,m}$ when F is absolutely continuous

Theorem 10 shows that the empirical CDF $F_{n,m}$ is a consistent estimator of the population CDF. We also want to understand the uncertainty of the empirical CDF. The Komlós-Major-Tusnády (KMT) approximation shows that $\sqrt{n}(F_n(x) - F(x))$ can be approximated by a sequence of Brownian bridges $\{B_n(x), 0 \leq x \leq 1\}$ (Komlós, Major, and Tusnády, 1975). This result can be extended to the empirical CDF based on $\hat{s}_i$'s; see Theorem 29. The proof is similar to Theorem 10 by splitting $F_{n,m}(x) - F(x)$ into $F_{n,m}(x) - F^{(m)}(x)$ and $F^{(m)}(x) - F(x)$. The former can be bounded by the original KMT approximation and the latter one can be bounded using Proposition 8.

Theorem 29 (Local inference of $F_{n,m}$) *Suppose F corresponds to a density f on $[0,1]$ with $f_{\max} < \infty$. There exists a sequence of Brownian bridges $\{B_n(x), 0 \le x \le 1\}$ such that*

$$\mathbb{P}\left\{\sup_{0\le x\le 1} |\sqrt{n}(F_{n,m}(x) - F(x)) - B_n(F(x))| > \frac{2\sqrt{2\pi} f_{\max}\cdot\sqrt{n}}{\sqrt{m}} + \frac{a\log n}{\sqrt{n}} + t\right\} \le be^{-c\sqrt{n}t},$$

for all positive integers n and all $t > 0$, where a, b and c are positive constants.

B.6.2 Proof of the local asymptotics

This proof is adapted from Wang (1992). Define

$$U_{n,m}(a) = \sup\{x : F_{n,m}(x) - ax \text{ is maximal }\}.$$

Then with probability one, we have the switching relation

$$\tilde{f}_{n,m}(t) \le a \Leftrightarrow U_{n,m}(a) \le t. \tag{B.6.1}$$

By the relation (B.6.1), we have

$$\mathbb{P}(\sqrt{n}(\tilde{f}_{n,m}(t_0) - f(t_0)) \le x) = \mathbb{P}(U_{n,m}(f(t_0) + n^{-\frac{1}{2}}x) \le t_0).$$

From the definition of $U_{n,m}$, it follows that

$$\begin{aligned} U_{n,m}(f(t_0) + n^{-\frac{1}{2}}x) &= \sup\{s : F_{n,m}(s) - (f(t_0) + n^{-\frac{1}{2}}x)s \text{ is maximal }\} \\ &= \sup\{s : \sqrt{n}(F_{n,m}(s) - F(s)) + \sqrt{n}(F(s) - f(t_0)) - xs \text{ is maximal }\} \end{aligned}$$

By Theorem 29,

$$\sqrt{n}(F_{n,m} - F(s)) = B_n(F(s)) + \mathcal{O}(\frac{\sqrt{n}}{\sqrt{m}}) + \mathcal{O}_p(\frac{\log n}{\sqrt{n}}),$$

where $\{B_n, n \in N\}$ is a sequence of Brownian Bridges, constructed on the same space as the F_n. So the limit distribution of $U_{n,m}(f(t_0) + n^{-\frac{1}{2}}x)$ is the same as that of the location of the maximum of the process $\{B_n(F(s)) + \sqrt{n}(F(s) - f(t_0)(s) - xs, s \ge 0\}$. Note that $F(s)$ is concave and linear in $[a,b]$, then

$$F(s) = F(a) + f(t_0)(s-a) \text{ for } s \in [a,b],$$

and

$$F(s) - f(t_0)(s-a) < F(a) \text{ for } s \notin [a,b].$$

Hence the location of the maximum of $\{(B_n(F(s)) + \sqrt{n}(F(s) - f(t_0)s) - xs, s \geq 0\}$ behaves asymptotically as that of

$$\{B(F(s)) - xs, a \leq s \leq b\} = \{B(F(a) + f(t_0)(s-a)) - xs, a \leq s \leq b\},$$

where B is a standard Brownian bridge in $[0,1]$. Thus,

$$\begin{aligned} & \mathbb{P}(\sqrt{n}(\tilde{f}_{n,m}(t_0) - f(t_0)) \leq x) \to \\ & \mathbb{P}(\text{ the location of the maximum of } \{B(F(s)) - xs, a \leq s \leq b \leq t_0\}) \\ = \ & \mathbb{P}(\hat{S}_{a,b}(t_0) \leq x), \end{aligned}$$

by the definition of $\hat{S}$. That completes the proof of Part (A). The proof of Part (B) follows in a similar manner.

B.7 Proof of Theorem 21

Proof Let $\hat{\alpha}_l(c_l)$, $\hat{\alpha}_r(c_r)$, $\hat{\alpha}_{mid}(c_l, c_r)$, $\tilde{g}_l(c_l)$, $\tilde{g}_r(c_r)$ be the output of Algorithm 5 with the input c_l, c_r and d_l, d_r. The corresponding estimator of f is termed as $\tilde{f}_{n,m}$. For simplicity, we consider $c_r^{(0)} = 1$, i.e., the case where there is only the decreasing part and the flat part. For a general case where $c_r^{(0)} < 1$, we just need to focus on $[0, \mu]$.

By Theorem 19, we know that $\mathbb{E}_f \int_0^{c_l^{(0)}} |\tilde{f}_{n,m}(x) - f(x)|dx \leq K_1 \cdot N_l(c_l^{(0)})^{-\frac{1}{3}}$ when $m \geq C_1 \cdot N_l(c_l^{(0)})$ for some positive constants K_1 and C_1 that only depend on f. For $c_l > c_l^{(0)}$, it is easy to see that $\lim_{x \to (c_l)_-} \tilde{f}_{m,n}(x) - \lim_{x \to (c_l)_+} \tilde{f}_{m,n}(x) = \varepsilon \cdot N_l(c_l^{(0)})^{-1/2}$, where ε is a residual term with $|\varepsilon| \leq K_2$ for some positive constant K_2, because the estimator of the flat region converges at a square-root rate (Wang, 1992). In other words, it is unlikely to find a desired gap beyond $c_l^{(0)}$.

If $\lim_{x \to (c_l^{(0)})_-} \tilde{f}_{n,m}(x) \geq \frac{\hat{\alpha}_{mid}(c_l^{(0)}, 1)}{1 - c_l^{(0)}} + d_l$, we select the desired $c_l = c_l^{(0)}$. Otherwise, define

$$t \text{ be the maximal } c_l \text{ such that } \begin{cases} c_l < c_l^{(0)} \\ \tilde{f}_{n,m}(c_l) \geq \frac{\hat{\alpha}_{mid}(c_l^{(0)}, 1)}{1 - c_l^{(0)}} + d_l. \end{cases}$$

Then $\forall t < c_l \leq c_l^{(0)}$, it follows that

$$\begin{aligned} f(c_l) - \tilde{f}_{n,m}(c_l) &= f(c_l) - \lim_{x \to (c_l^{(0)})_+} f(x) - (\tilde{f}_{n,m}(c_l) - \frac{\hat{\alpha}_{mid}(c_l^{(0)}, 1)}{1 - c_l^{(0)}}) \\ &\quad + \lim_{x \to (c_l^{(0)})_+} f(x) - (\frac{\hat{\alpha}_{mid}(c_l^{(0)}, 1)}{1 - c_l^{(0)}}) \\ &\geq \delta_l - d_l - K_2 \cdot (N_{mid}(c_l^{(0)}, 1)^{-\frac{1}{2}}), \end{aligned}$$

When $d_l < \delta_l$, it implies that

$$\mathbb{E}_f \int_t^{c_l^{(0)}} |f(x) - \tilde{f}_{n,m}(x)|dx \geq (\delta_l - d_l - K_2 \cdot N_{mid}(c_l^{(0)}, 1)^{-\frac{1}{2}}) \cdot (c_l^{(0)} - t).$$

Since the L_1 distance between f and $\tilde{f}_{n,m}$ reduces at a cubic-root rate, it follows that $c_l^{(0)} - t \leq C_2 \cdot N_l(c_l^{(0)})^{-1/3}$ for some positive constant C_2. So $c_l = t$ is the desired cutoff. Finally, we have that

$$\begin{aligned}
\int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx &= \int_0^t |\tilde{f}_{n,m}(x) - f(x)|dx + \int_t^{c_l^{(0)}} |\frac{\hat{\alpha}_{mid}(t,1)}{1-t} - f(x)|dx \\
&\quad + \int_{c_l^{(0)}}^1 |\frac{\hat{\alpha}_{mid}(t,1)}{1-t} - f(x)|dx \\
&\leq \int_0^t |\tilde{f}_{n,m}(x) - f(x)|dx + (\frac{1}{1-t} + f_{\max}) \cdot |c_l^{(0)} - t| \\
&\quad + |\frac{\hat{\alpha}_{mid}(t,1)}{1-t} - \frac{\alpha_{mid}(c_l^{(0)}, 1)}{1 - c_l^{(0)}}| \cdot |1 - c_l^{(0)}| \\
&\leq C_4 \cdot N_l(c_l^{(0)})^{-1/3},
\end{aligned}$$

where the last inequality holds because $|c_l^{(0)} - t| \leq C_2 \cdot N_l(c_l^{(0)})^{-1/3}$ and f is bounded can imply $|\frac{\hat{\alpha}_{mid}(t,1)}{1-t} - \frac{\alpha_{mid}(c_l^{(0)},1)}{1-c_l^{(0)}}| \leq K_3 \cdot N_l(c_l^{(0)})^{-1/3}$. Here C_4, K_3 are two positive constants that only depend on f. ■

B.8 Proof of Theorem 22

Proof By considering the interior point μ in the flat region, the left decreasing part and the right increasing part are disentangled. Therefore, we only need to consider the left side, and the right side can be proven in the same way. A necessary condition for c_l being identified as feasible for the change-point-gap constraint in Algorithm 5 is that

$$\tilde{g}_l(c_l) \geq \frac{\hat{\alpha}_{mid}(c_l, \mu)}{\hat{\alpha}_l(\mu) \cdot (\mu - c_l)} + \frac{d_l}{\hat{\alpha}_l(\mu)},$$

where $\hat{\alpha}_{mid}(c_l, \mu) = N_{mid}(c_l, \mu)/n$ and $\hat{\alpha}_l(\mu) = N_l(\mu)/n$. It is easy to see that

$$\hat{\alpha}_l(\mu) = \alpha_l(\mu) + \varepsilon_1 \cdot n^{-\frac{1}{2}},$$

and

$$\hat{\alpha}_{mid}(c_l,\mu) = \alpha_{mid}(c_l,\mu) + \varepsilon_2 \cdot n^{-\frac{1}{2}},$$

where $\alpha_l(\mu) = \mathbb{E}\hat{\alpha}_l(\mu)$ and $\alpha_{mid}(c_l,\mu) = \mathbb{E}\hat{\alpha}_{mid}(c_l,\mu)$; ε_1 and ε_2 are residual terms with $\max(|\varepsilon_1|,|\varepsilon_2|) \leq K$ for some universal positive constant K by McDiarmid inequality. So the necessary condition is

$$\tilde{g}_l(c_l) \geq \frac{\alpha_{mid}(c_l,\mu)}{\alpha_l(\mu)\cdot(\mu-c_l)} + \frac{d_l}{\alpha_l(\mu)} + \varepsilon_3 \cdot n^{-\frac{1}{2}}, \tag{B.8.1}$$

with $|\varepsilon_3| \leq C$ for some constant C that only depends on $\alpha_l(\mu)$, $\mu - c_l$ and d_l. If $c_l > c_l^{(0)}$ and the constraint is violated at c_l, then for any $c_l' > c_l$ the constraint is violated at c_l' with high probability since $\tilde{g}_l(c_l) \geq \tilde{g}_l(c_l')$ and $\frac{\alpha_{mid}(c_l,\mu)}{\alpha_l(\mu)\cdot(\mu-c_l)} = \frac{\alpha_{mid}(c_l',\mu)}{\alpha_l(\mu)\cdot(\mu-c_l')}$ (c_l and c_l' are both in the flat region). Therefore, to see if $\hat{c}_l > c_l^{(0)}$, we only need to investigate the smallest c_l with $c_l > c_l^{(0)}$ that is in the searching space of Algorithm 5. By Theorem 20, when $m/\cdot N_l(c_l^{(0)}) \to \infty$, it follows that

$$\sqrt{N_l(\mu)}(\tilde{g}_l(c_l) - \frac{\alpha_{mid}(c_l,\mu)}{\alpha_l(\mu)\cdot(\mu-c_l)}) \xrightarrow{d} \hat{S}_{[c_l^{(0)},\mu]}(c_l).$$

By the necessary condition (B.8.1), then asymptotically we have

$$\begin{aligned}
\mathbb{P}[c_l > c_l^{(0)}] &\leq \mathbb{P}\left(\sqrt{N_l(\mu)}(\tilde{g}_l(c_l) - \frac{\alpha_{mid}(c_l,\mu)}{\alpha_l(\mu)\cdot(\mu-c_l)}) \geq \sqrt{N_l(\mu)}\cdot\frac{d_l}{\alpha_l(\mu)} + \varepsilon_3\cdot\sqrt{\frac{N_l(\mu)}{n}}\right) \\
&\approx \mathbb{P}\left(\hat{S}_{c_l^{(0)},\mu}(c_l) \geq \sqrt{N_l(\mu)}\cdot\frac{d_l}{\alpha_l(\mu)} + \varepsilon_3\cdot\sqrt{\frac{N_l(\mu)}{n}}\right) \\
&\leq \mathbb{P}\left(\hat{S}_{c_l^{(0)},\mu}(c_l) \geq \sqrt{N_l(c_l^{(0)})}\cdot\frac{d_l}{\alpha_l(\mu)} - C\cdot\sqrt{\frac{N_l(\mu)}{n}}\right). \\
&\leq \mathbb{P}\left(\hat{S}_{c_l^{(0)},\mu}(c_l) \geq \sqrt{N_l(c_l^{(0)})}\cdot\frac{d_l}{\alpha_l(\mu)} - C\right). \\
&\approx \mathbb{P}\left(\hat{S}_{c_l^{(0)},\mu}(c_l^{(0)}) \geq \sqrt{N_l(c_l^{(0)})}\cdot\frac{d_l}{\alpha_l(\mu)} - C\right).
\end{aligned}$$

where the last approximation holds because c_l is the smallest one searching candidate with $c_l > c_l^{(0)}$, so it is very close to $c_l^{(0)}$. ■

B.9 Proof of Theorem 23

Proof Let $\hat{\alpha}_l(c_l)$, $\hat{\alpha}_r(c_r)$, $\hat{\alpha}_{mid}(c_l,c_r)$, $\tilde{g}_l(c_l)$, $\tilde{g}_r(c_r)$ be the output of Algorithm 5 with the input c_l, c_r and d_l, d_r. The corresponding estimator of f is termed as $\tilde{f}_{n,m}$. Let

$H(c_l, c_r, \tilde{f}_{n,m})$ be the log likelihood function associated with the optimization problem (3.5.3), and $N(x) = \#\{\hat{s}_i : \hat{s}_i = x\}$. It follows that

$$\begin{aligned}\frac{1}{n}\mathbb{E}_f H(c_l, c_r, \tilde{f}_{n,m}) &= \frac{1}{n}\mathbb{E}_f \sum_{\hat{s}_i} \log \tilde{f}_{n,m}(\hat{s}_i) \\ &= \mathbb{E}_f \log \tilde{f}_{n,m}(\hat{s}_1) \\ &= -KL(f||\tilde{f}_{n,m}) + C,\end{aligned}$$

where $C = \mathbb{E}_f \log f(\hat{s}_1)$. From the relations between total variation, Kullback-Leibler divergence and the χ^2 distance,

$$TV(P, Q) \leq \sqrt{KL(P||Q)} \leq \sqrt{\chi^2(P||Q)},$$

we have

$$\begin{aligned}\frac{1}{n}\mathbb{E}_f H(c_l, c_r, \tilde{f}_{n,m}) = -\mathbb{E}_f KL(f||\tilde{f}_{n,m}) + C &\leq -\mathbb{E}_f(\int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx)^2 + C \\ &\leq -(\mathbb{E}_f \int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx)^2 + C,\end{aligned}$$

where the last inequality uses the Jensen's inequality. On the other hand, it follows that

$$\begin{aligned}\frac{1}{n}\mathbb{E}_f H(c_l, c_r, \tilde{f}_{n,m}) &= -\mathbb{E}_f KL(f||\tilde{f}_{n,m}) + C \\ &\geq -\mathbb{E}_f \int_0^1 (f(x) - \tilde{f}_{n,m}(x))^2/f(x)dx + C \\ &\geq -\mathbb{E}_f \int_0^1 (f(x) - \tilde{f}_{n,m}(x))^2/f_{min}dx + C \\ &\geq -\frac{1}{f_{min}} \cdot (\mathbb{E}_f \int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx)^2 + C,\end{aligned}$$

Then the problem is reduced to bound $\mathbb{E}_f \int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx$. From Theorem 21, we know that when $m \geq C_1 \cdot \max(N_l(c_l^{(0)}), N_r(c_r^{(0)}))$, there exist c_l and c_r in the neighborhoods of $c_l^{(0)}$ and $c_r^{(0)}$ respectively, such that the resulting estimator $\tilde{f}_{n,m}$ satisfies $\mathbb{E}_f \int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx \leq K_1 \cdot (N_l(c_l^{(0)})^{-1/3} + N_r(c_r^{(0)})^{-1/3})$ for some positive constants C_1 and K_1. Along with Lemma 30, we conclude the desired result.

■

Lemma 30 *Let $\tilde{f}_{n,m}$ be the solution by Algorithm 5 with input c_l and c_r and the corresponding $flag$ = True. Assume $f_{\max} < \infty$ and $f_{\min} > 0$. If $m \geq C_1 \cdot \max(N_l(c_l^{(0)}), N_r(c_r^{(0)}))$, then $|\Delta_l| \leq C_2 \cdot N_l(c_l^{(0)})^{-1/3}, |\Delta_r| \leq C_3 \cdot N_r(c_r^{(0)})^{-1/3}$ is a necessary condition for*

$$\mathbb{E}_f \int_0^1 |f(x) - \tilde{f}_{n,m}(x)|dx \leq C_4 \cdot (N_l(c_l^{(0)})^{-1/3} + N_r(c_r^{(0)})^{-1/3}),$$

where C_1, C_2, C_3, C_4 are four constants depending on d_l, d_r, $f_{\max}$ and $f_{\min}$; $\Delta_l = c_l - c_l^{(0)}$, $\Delta_r = c_r - c_r^{(0)}$.

Proof For simplicity, we consider $c_r^{(0)} = 1$, i.e., the case where there is only the decreasing part and the flat part. For a general case where $c_r^{(0)} < 1$, we just need to focus on $[0, \mu]$. If $c_l < c_l^{(0)}$, the L_1 distance between $\tilde{f}_{n,m}$ and f is

$$\begin{aligned}
& \mathbb{E}_f \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \\
= \; & \mathbb{E}_f \int_0^{c_l^{(0)}} |\tilde{f}_{n,m}(x) - f(x)|dx + \mathbb{E}_f \int_{c_l^{(0)}}^1 |\tilde{f}_{n,m}(x) - f(x)|dx \\
\overset{(a)}{\geq} \; & -K_1 \cdot N_l(c_l)^{-1/3} + \mathbb{E}_f \int_{c_l^{(0)}}^1 |\tilde{f}_{n,m}(x) - f(x)|dx \\
\overset{(b)}{=} \; & -K_1 \cdot N_l(c_l)^{-1/3} + \mathbb{E}_f |\frac{1 - \int_0^{c_l} \tilde{f}_{n,m}(x)dx}{1 - c_l} - \frac{1 - \int_0^{c_l^{(0)}} f(x)dx}{1 - c_l^{(0)}}|(1 - c_l^{(0)}) \\
\overset{(c)}{\geq} \; & -K_2 \cdot N_l(c_l^{(0)})^{-1/3} + |\frac{1 - \int_0^{c_l} f(x)dx}{1 - c_l} - \frac{1 - \int_0^{c_l^{(0)}} f(x)dx}{1 - c_l^{(0)}}|(1 - c_l^{(0)}) \\
= \; & -K_2 \cdot N_l(c_l^{(0)})^{-1/3} + |\frac{-\Delta_l + (1 - c_l)\int_{c_l}^{c_l^{(0)}} f(x)dx + \Delta_l \int_0^{c_l^{(0)}} f(x)dx}{1 - c_l}| \\
= \; & -K_2 \cdot N_l(c_l^{(0)})^{-1/3} + |\frac{-\Delta_l - (1 - c_l)\Delta_l\gamma + \Delta_l \int_0^{c_l^{(0)}} f(x)dx}{1 - c_l}| \\
= \; & -K_2 \cdot N_l(c_l^{(0)})^{-1/3} + \kappa|\Delta_l|,
\end{aligned}$$

where K_1 and K_2 are two positive constants that only depend on f, $\min_{x \in [c_l, c_l^{(0)}]} f(x) \leq \gamma \leq \max_{x \in [c_l, c_l^{(0)}]} f(x)$, $\kappa = |\frac{1 + (1 - c_l)\gamma - \int_0^{c_l^{(0)}} f(x)dx}{1 - c_l}| < \infty$ since $f_{\max} < \infty$. The inequality (a) and the equation (c) are obtained using Theorem 19. The equation (b) makes use of

assumption that the right hand side is a flat region. Then, if there exists $C_4 > 0$ such that $\mathbb{E}_f \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \leq C_4 \cdot N_l(c_l^{(0)})^{-1/3}$, then $|\Delta_l| \leq C_2 \cdot N_l(c_l^{(0)})^{-1/3}$, for some positive constant C_2.

Next, we investigate the case when $c_l > c_l^{(0)}$. Denote $a := \lim_{x\to(c_l)_-} \tilde{f}_{n,m}(x)$, $b := \lim_{x\to(c_l)_+} \tilde{f}_{n,m}(x)$, $c := \int_{c_l^{(0)}}^{c_l} \tilde{f}_{n,m}(x)dx$, and $\alpha \equiv f(x)$(when $x > c_l^{(0)}$). It is easy to check that

$$|c - \alpha| \cdot |\Delta_l| \leq \int_{c_l^{(0)}}^{c_l} |\tilde{f}_{n,m}(x) - \alpha|dx.$$

Using the L_1 convergence of the Grenander estimator, it follows that

$$|\Delta_l||c - \alpha| + (1 - c_l)|b - \alpha| \leq \int_{c_l^{(0)}}^{c_l} |\tilde{f}_{n,m}(x) - \alpha|dx + (1 - c_l)|b - \alpha| \leq \int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx.$$

If there exists $C_4 > 0$ such that $\int_0^1 |\tilde{f}_{n,m}(x) - f(x)|dx \leq C_4 \cdot N_l(c_l^{(0)})^{-1/3}$, we have

$$\begin{aligned}
\frac{|\Delta_l|}{1 - c_l^{(0)}}|a - b| &\leq \frac{|\Delta_l|}{1 - c_l^{(0)}}|c - b| \\
&\leq \frac{|\Delta_l|}{1 - c_l^{(0)}}|c - \alpha| + \frac{|\Delta_l|}{1 - c_l^{(0)}}|\alpha - b| \\
&\leq \frac{|\Delta_l|}{1 - c_l}|c - \alpha| + |\alpha - b| \leq \frac{C_4}{1 - c_l} \cdot N_l(c_l^{(0)})^{-1/3}.
\end{aligned}$$

If the output $flag$ of Algorithm 5 is *True*, $a - b \geq \frac{\alpha_{mid}(c_l,\mu)}{\alpha_l(\mu)(1-c_l)} + \frac{d_l}{\alpha_l(\mu)} + K_3 \cdot n^{-1/2}$ for some positive constant K_3. Then it must follow that $|\Delta_l| \leq C_2 \cdot N_l(c_l^{(0)})^{-1/3}$, where $C_2 > 0$. So far, we have proven the lemma for the left hand side. For the right hand side, it can be proven similarly. ■