SUPERVISED DIMENSION REDUCTION TECHNIQUES FOR HIGH-DIMENSIONAL DATA

By

Dylan Molho

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computational Mathematics, Science, and Engineering – Doctor of Philosophy

2022

ABSTRACT

SUPERVISED DIMENSION REDUCTION TECHNIQUES FOR HIGH-DIMENSIONAL DATA

By

Dylan Molho

The data sets arising in modern science and engineering are often extremely large, befitting the era of big data. But these data sets are not only large in the number of samples they have, they may also have a large number of features, placing each data point in a high-dimensional space. However, unique problems arise when the dimension of the data has the same or even greater order than the sample size. This scenario in statistics is known as the High Dimension, Low Sample Size problem (HDLSS). In this paradigm, many standard statistical estimators are shown to perform sub-optimally and in some cases can not be computed at all.

This dissertation develops two novel algorithms that successfully operate in the paradigm of HDLSS. We first propose the Generalized Eigenvalue (GEV) estimator, a unified sparse projection regression framework for estimating generalized eigenvector problems. Unlike existing work, we reformulate a sequence of computationally intractable non-convex generalized Rayleigh quotient optimization problems into a computationally efficient simultaneous linear regression problem, padded with a sparse penalty to deal with high-dimensional predictors. We showcase the applications of our method by considering three iconic problems in statistics: the sliced inverse regression (SIR), linear discriminant analysis (LDA), and canonical correlation analysis (CCA). We show the reformulated linear regression problem is able to recover the same projection space obtained by the original generalized eigenvalue problem. Statistically, we establish the nonasymptotic error bounds for the proposed estimator in the applications of SIR and LDA, and prove these rates are minimax optimal. We present how the GEV is applied to the CCA problem, and adapt the method for a robust Huber-loss based formulation for noisy data. We test our framework on both synthetic and real datasets and demonstrate its superior performance compared with other state-of-the-art methods in high dimensional statistics.

The second algorithm is the scJEGNN, a graphical neural network (GNN) tailored to the task of data integration for HDLSS single-cell sequencing data. We show that with its unique model, the GNN is able to leverage structural information of the biological data relations in order to perform a joint embedding of multiple modalities of single-cell gene expression data. The model is applied to data from the NeurIPS 2021 competition for Open Problems in Single-Cell Analysis, and we demonstrate that our model is able to outperform top teams from the joint embedding task.

ACKNOWLEDGEMENTS

I would like to thank my advisor and committee chair, Dr. Yuying Xie. Since the start of my time with him, he has shown commitment and passion to research while also being endlessly patient and adaptive. His knowledge of a diverse range of applied and theoretical techniques has been invaluable. He has been the perfect combination of professional and personable. I also want to thank my co-advisor, Dr. Qiang Sun. His depth of expertise in statistics and mathematics has been inspiring and served as constant motivation for my own work. His insight and clarity in our work has helped guide both my research and development as a scholar. I regret Covid taking away an opportunity to have worked with him more in person.

I would also like to thank my other committee members, Dr. Ming Yan and Dr. Rongrong Wang for their knowledgeable feedback and suggestions into further research directions.

I also want to thank the CMSE community, Lisa Roy, Heather Williams, etc. They are always there to offer help.

Lastly, thank you to friends and family for providing so much support during this time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

In the past twenty years, we have witnessed an explosion of data from different domain areas, including medical imaging, finance, and genomics. The data sets arising in modern science and engineering are often extremely large, befitting the era of big data. But these data sets are not only large in the number of samples they have, they may also have a large number of features, placing each data point in a high-dimensional space. Data like this is common in fields like biology, where for instance measurements of the gene expression of cells can have tens of thousands or even hundreds of thousands features. This enrichment of data offers promises of solutions to many challenging goals, including detecting genes underlying complex diseases and designing novel drug treatments. However, unique problems arise when the dimension of the data has the same order as or even greater than the sample size. This scenario in statistics is known as a High Dimension, Low Sample Size problem (HDLSS) [HMN05, SSZM16]. In such a regime, many classical statistical methods no longer have guarantees of success, and standard asymptotic theory often fails to provide useful predictions. As well, in high dimensions, our intuitions on basic concepts such as the distance between points begins to break down. As a result of the "curse of dimensionality", higher dimensional versions of the cube no longer have the majority of their mass near the center of the cube, but instead the vast majority of volume is found near the corners. Similarly, multivariate normal distributions more and more act like uniform distributions on hyperspheres, and estimators like k-nearest neighbors become unreliable due to distances being very similar between points in the high-dimensional data set.

To better illustrate this behavior, we give a simple example of Linear Discriminant Analysis (LDA), a classical machine learning technique that trains a classifier for the data. The classifier is determined based on finding a linear projection of the data to a lower dimensional space that best separates the data based on its class. In Figure 1.1 we see unlabeled data in 2-dimensions that have a high spread along the x-axis, with very little variation along the y-axis. If we sought

to simplify the data to a one-dimensional representation, we could simply project the data to the x-axis, preserving most of the variation of the data.

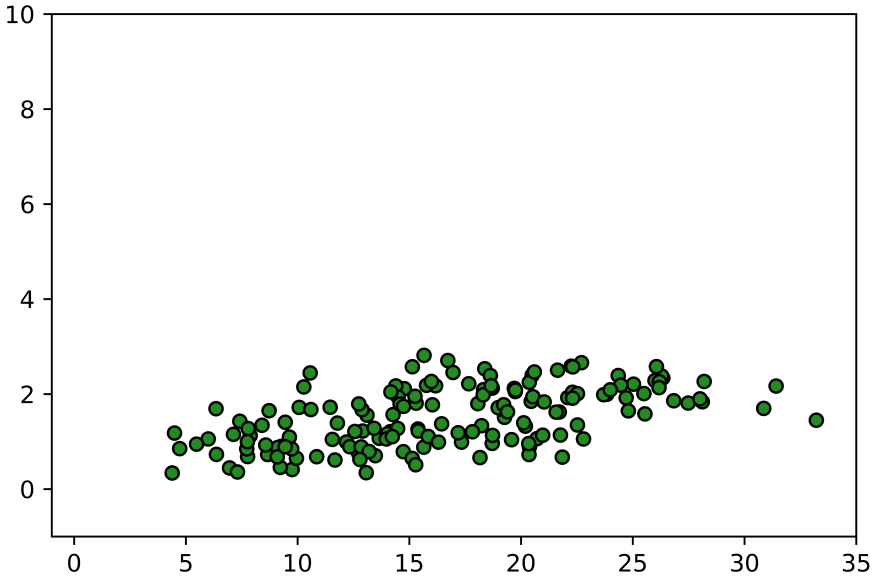

Figure 1.1: Unlabeled 2-dimensional data.

Once the data is given additional class information, we see that most of the pertinent information about how the classes are separated would be lost upon projection to the x-axis. We see in Figure 1.3 that the projection of the data on the x-axis mixes much of the class data, making it nearly impossible to find a good point to separate the classes. On the other hand, projection on y-axis does a good job of separating the class information, despite being more compactly spaced after projection. It is not too surprising that this is case: the data is generated from two multivariate normal distributions with mean $\mu_1 = (13,1)$ for class 1 and mean $\mu_2 = (21,2)$, with both classes sharing a diagonal covariance of

$$\Sigma = \begin{bmatrix} 25 & 0 \\ 0 & .1 \end{bmatrix}.$$

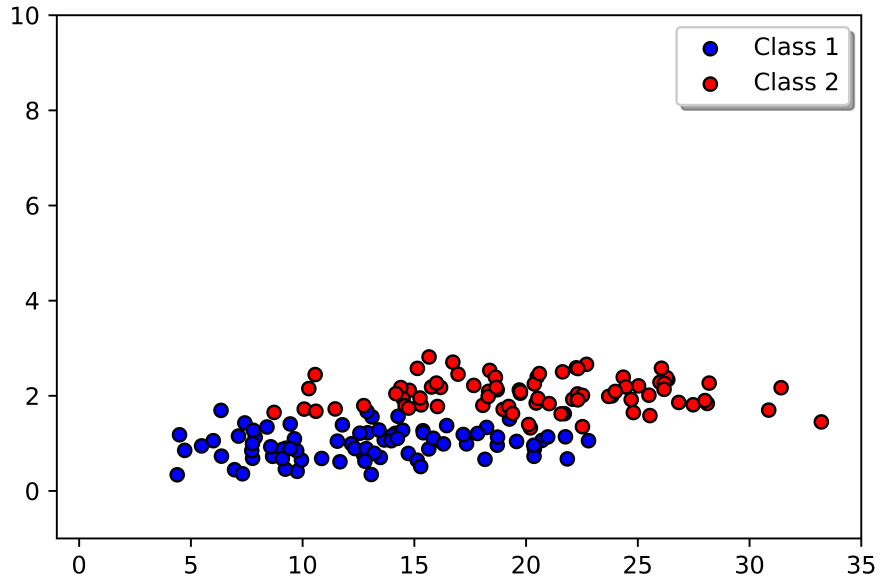

Figure 1.2: Labeled 2-dimensional data.

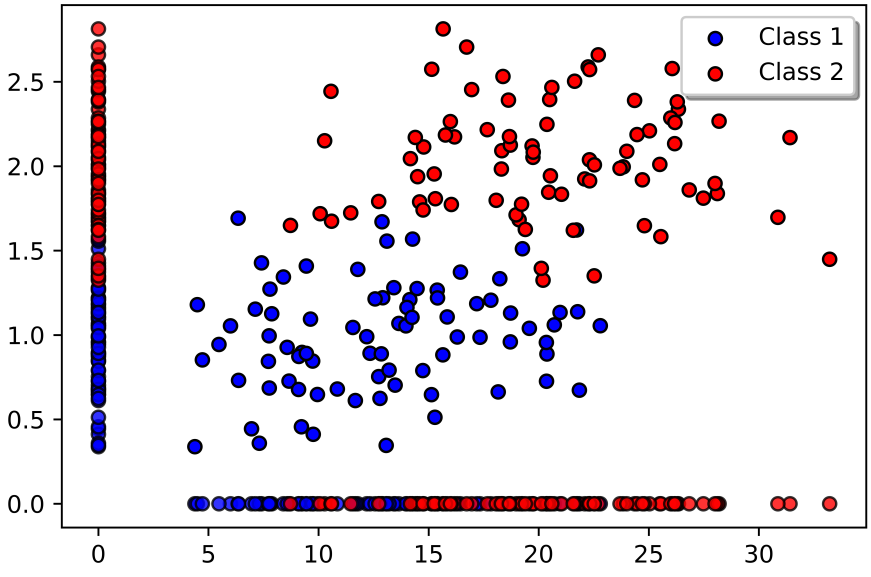

Figure 1.3: Data projection on the x and y axis.

While the projection of the data on the y-axis fared well in distinguishing the class information, we can do even better. The solution provided by LDA takes into account the location of the sample means, and corrects for the covariance of the data to make a more optimal separation of the data. Let $\hat{\mu}_1$ and $\hat{\mu}_2$ be our estimated class means from the data, marked by the white x's in Figure 1.4, and let $\widehat{\Sigma}$ be our estimated covariance. Then the linear discriminant function applied to a new data point $\boldsymbol{x} \in \mathbb{R}^2$ is given by the inner product

$$\widehat{\Psi}(\boldsymbol{x}) = \langle \hat{\mu}_1 - \hat{\mu}_2, \widehat{\Sigma}^{-1} \left(\boldsymbol{x} - \frac{\hat{\mu}_1 + \hat{\mu}_2}{2} \right) \rangle.$$

If this value is less than 0, label the point $\boldsymbol{x}$ as belonging to class 1, and if it is greater than 0, then it is labeled belonging to class 2. This provides a linear decision boundary in the plane, given by the yellow dotted line in Figure 1.4, where the values of $\widehat{\Psi}(\boldsymbol{x}) = 0$. The decision boundary shows how the LDA classifier labels new points, and if we wanted to project the data to best separate the samples by classes, we would project the points to a line that is orthogonal to this decision boundary.

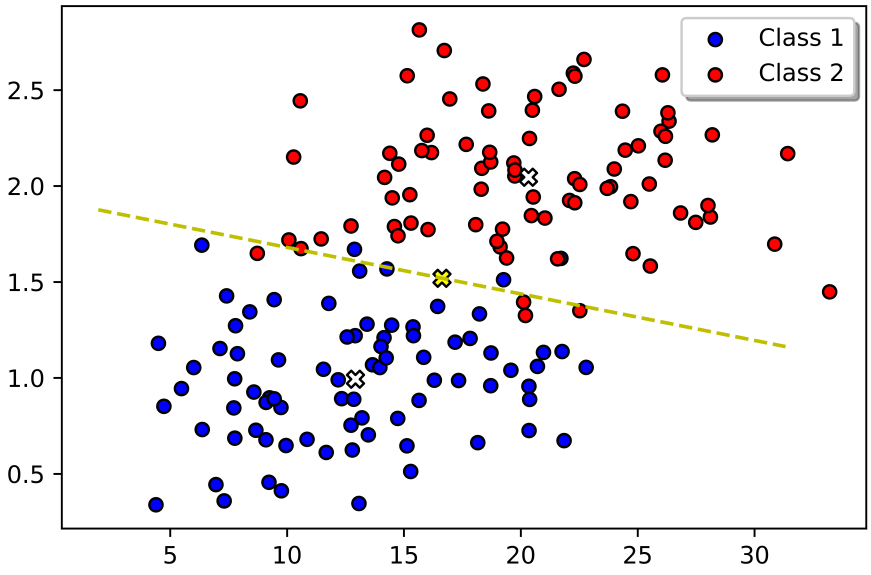

Figure 1.4: The LDA decision boundary between samples from Class 1 and Class2. White x's mark the sample means, and the yellow x gives the midpoint between the two.

If we assume that both classes are equally likely, then we can calculate the error probability using the LDA decision boundary as

$$\mathrm{Err}(\widehat{\Psi}) = \frac{1}{2}\mathbb{P}_1[\widehat{\Psi}(\boldsymbol{x}') \leq 0] + \frac{1}{2}\mathbb{P}_2[\widehat{\Psi}(\boldsymbol{x}'') > 0],$$

where $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are samples drawn independently from probability distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ of class 1 and 2 respectively. We can analyze the behavior of the error asymptotically, i.e. look at the limiting behavior as our sample size n increases to infinity. Define the value $\gamma = \|\mu_1 - \mu_2\|_2$, the distance between the two class means, and let us simplify the problem further by assuming the covariance $\boldsymbol{\Sigma}$ is the identity matrix. Then the random error $\mathrm{Err}(\widehat{\Psi})$ converges in probability to the fixed number

$$\mathrm{Err}(\widehat{\Psi}_{\mathrm{Id}}) \xrightarrow{\text{prob.}} \Phi\left(-\frac{\gamma}{2}\right),$$

where $\Phi(t) = \mathbb{P}[Z \leq t]$ is the cumulative distribution function of the standard normal variable. Thus the error behaves purely as a function of the distance of the class means as n increases: as the means grow in distance, the amount of error goes down. But this is assuming that the dimension of the data remains fixed. If instead we had the dimension of the data d at a fixed ratio with the sample size n, we would see a different picture emerge. In many cases this fixed ratio is more realistic for applications where the dimension is high and collecting much larger samples is infeasible. Assume the ratio d/n converges to some non-negative fraction $\alpha > 0$. Under this high-dimensional scaling, our error converges to a different sub-optimal value

$$\mathrm{Err}(\widehat{\Psi}_{\mathrm{Id}}) \xrightarrow{\text{prob.}} \Phi\left(-\frac{\gamma^2}{2\sqrt{\gamma^2 + 2\alpha}}\right).$$

In this case the classical prediction $\Phi(-\gamma/2)$ drastically underestimates the error rate. Our example application of LDA was applied to data that is only 2-dimensional with a sample size of 80. We would see a quick deterioration in performance if the dimension was much higher, e.g. $d = 40$. At the point $d > n$, we are unable to even calculate our estimator, since estimates for the covariance $\widehat{\boldsymbol{\Sigma}}$ are no longer full rank, which makes computing its inverse impossible.

These HDLSS phenomena necessitates the development of new theory as well as new methods in order to manage these difficult problems, and achieve the desired outcomes for downstream

data science tasks. To overcome the barriers found in HDLSS scenarios, one must make additional assumptions on the data, either with explicit formulations or with implicit beliefs about the behavior of the data. The first type of research leads to structural assumptions placed on the probability model that generates the data, which allow for alterations to classical methods to yield theoretically optimal estimators for the chosen well-defined tasks. The second type of research, in contrast, makes general assumptions usually based on the the causal nature of chosen real-world data application, where the data is assumed to have dependencies between the various parameters.

While there are no theoretical guarantees for such methods, the strength of the estimator is instead demonstrated empirically on simulated or real data sets. We explore ideas from both of these fields, and develop two novel algorithms that successfully operate in the paradigm of HDLSS.

In the first case, we develop an estimator for high-dimensional data with response variables with the assumption that the data has an underlying low-rank structure, and that the lower-dimensional representation is obtained with sparse projection directions. Sparsity and low rank representations are natural assumptions to make on high-dimensional data due to the likelihood of having very few of the potential thousands of variables contribute meaningfully to a response variable. Data exhibiting this structure can be found in a diverse collection of applications, ranging from genomics to economics. Our estimator produces a supervised linear dimension reduction of the data that attempts to maximally preserve the relationship between the covariate data and the response variables. We achieve this through finding vectors that relate the covariance of the covariates with the covariance between the data and response variables, solving a type of generalized eigenvalue optimization problem. We name the method the Generalized Eigenvalue (GEV) Estimator, and show that this method is able to solve three separate classical statistical problems in the HDLSS paradigm: Linear Discriminant Analysis (LDA), Sliced Inverse Regression (SIR), and Canonical Correlation Analysis (CCA). We give theoretical guarantees of convergence that are shown to be minimax optimal for both LDA and SIR, and give empirical results of GEV outperforming competitor methods in all three applications on simulated data, as well as applications of GEV to real-world data analysis tasks on gene expression data.

For the second case, we design a method for multi-modal data integration using deep learning. We tailor a graphical neural network (GNN) for use on single-cell sequencing data, a rich new data source in biology that has revolutionized the field. Single-cell sequencing data produces matrices where each row represents a cell, and each column gives a value corresponding to the expression of some gene. This data often falls in the HDLSS regime, since sequencing cells is an expensive process, but can produce for each cell potentially hundreds of thousands of features given by different genes. We show that by using a bipartite graphical model of the data that represents both cell and genes as nodes, we are able to leverage the causal structure of the gene expression data to create a low-dimensional representation that preserves important biological information. The GNN is combined with an autoencoder model in order to train a low-dimensional representation through latent feature regularization. We apply our model named scJEGNN to the task for single-cell multi-modality data integration in the NeurIPS 2021 special competition for Open Problems in Single-Cell Analysis, and we demonstrate our model is able to outperform the best performing competitor models.

The structure of this dissertation is as follows: in Chapter 2 we go over some preliminary mathematical background leading up to the GEV estimator, introduce the basics of neural networks and the specific models we use from deep learning, and give some description of single-cell data, which features prominently in our applications. In Chapter 3 we develop the formal theory of the GEV estimator, including the nonasymptotic convergence theorems and minimax bounds. In Chapter 4 we give the computational algorithm for the GEV estimator, and show the application of the estimator to simulated and real data problems. In Chapter 5 we detail the scJEGNN architecture and show its performance on the single-cell sequencing data integration task.

CHAPTER 2

BACKGROUND

In this chapter we develop the necessary mathematical background to understand the work. We assume some familiarity with linear algebra, multivariate calculus, and probability theory, but we first review some relevant concepts from these below. Then we review the basics of neural networks, including feedforward networks, autoencoder, and graphical neural networks. Lastly we introduce single-cell data and its structure and behavior.

2.1 Mathematical Preliminaries

Probability. Let X be a continuous real-valued random variable with probability distribution $\mathbb{P}$. If one wishes to understand how spread out X is from its mean $\mathbb{E}[X] = \mu$, then we can look at the tail probabilities $\mathbb{P}(|X-\mu| \geq t)$ for $t > 0$. This value gives us a concentration inequality, which tells us how likely it is for X to be a distance of t from its mean μ. Often it is beneficial to have random variables that concentrate around its mean. A common example is when we have a population parameter we are trying to estimate with a function of the data, which is a random variable, and the function has mean equal to the population parameter. If $X \sim N(\mu, \sigma^2)$ is normally distributed, then we can show that X has quick decay of probability in its tails:

$$\mathbb{P}[|X-\mu| \geq t] \leq 2e^{-\frac{t^2}{2\sigma^2}} \quad \text{for all } t \in \mathbb{R}. \tag{2.1}$$

This quick rate of decay as a function of t is a general property that we wish to generalize. To that end we call a random variable X sub-Gaussian if there is a positive number σ such that equation (2.1) holds for X. A similar but weaker property that we define is as follows: X is sub-exponential with parameters (ν, α) if for all $t \in \mathbb{R}$

$$\mathbb{P}[|X-\mu| \geq t] \leq \begin{cases} 2e^{-\frac{t^2}{2\nu^2}} & \text{if } 0 \leq t \leq \frac{\nu^2}{\alpha} \\ 2e^{-\frac{t}{2\alpha}} & \text{if } t > \frac{\nu^2}{\alpha}. \end{cases} \tag{2.2}$$

For sub-exponential variables, when t is small enough, the concentration inequality is sub-Gaussian in nature (i.e. with the exponent quadratic in t), but for larger t, the exponential component of the bound scales linearly in t. The location of this shift in behavior is then controlled by the parameter α, and in the limit $\alpha \to 0$, we get back sub-Gaussian inequalities.

If we have two continuous random variables X and X'' with probability distributions $\mathbb{P}$ and $\mathbb{Q}$ respectively, then the Kullback-Leibler divergence (KL divergence) between the distributions is defined to be

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

where p and q denote the probability densities functions of $\mathbb{P}$ and $\mathbb{Q}$. For simplicity denote $[n] = \{1,\ldots,n\}$ as the discrete set from 1 to n. Lastly if we have a collection of random variables $X_1,\ldots,X_n$, we define the first order statistic $X_{(1)}$ to be the minimum value of the collection $\{X_i\}_{i\in[n]}$, the second order statistic $X_{(2)}$ to be the second smallest value of the collection, and so on to the n^{th} order statistic $X_{(n)}$, which is the largest value of the collection.

Linear Algebra. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a real m by n matrix, and assume that $m > n$ and that A has full rank. The singular value decomposition of $\mathbf{A}$ is defined to be the identity

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\Sigma = \text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive real numbers on the diagonal and 0's elsewhere, $\mathbf{U} = [\mathbf{u}_1|\mathbf{u}_2|\cdots|\mathbf{u}_n] \in \mathbb{R}^{m\times n}$ is a orthonormal matrix with columns $\mathbf{u}_i \in \mathbb{R}^m$, and $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_n] \in \mathbb{R}^{n\times n}$ is a orthonormal matrix with columns $\mathbf{v}_i \in \mathbb{R}^n$. If $\mathbf{A}$ is a square symmetric matrix in $\mathbb{R}^{n\times n}$, then the eigendecomposition of $\mathbf{A}$ is the identity

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$$

where $\Lambda = \text{diag}(\lambda_1,\lambda_2,\ldots,\lambda_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix and $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_n] \in \mathbb{R}^{n\times n}$ is a orthogonal matrix where each $(\mathbf{v}_i,\lambda_i)$ is an eigenvector/eigenvalue-pair of $\mathbf{A}$ satisfying $\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i$. For two matrices $\mathbf{A},\mathbf{B} \in \mathbb{R}^{d\times d}$, we define the vector/scalar pair $(\mathbf{v},\rho)$ with $\mathbf{v} \in \mathbb{R}^d$ and $\rho \in \mathbb{R}$ to be

a generalized eigenpair if

$$\mathbf{A}\mathbf{v} = \rho\mathbf{B}\mathbf{v}.$$

In the case that $\mathbf{B}$ is nonsingular, the pairs $(\mathbf{v}, \rho)$ correspond to the eigenpairs of the matrix $\mathbf{B}^{-1}\mathbf{A}$. Alternatively, finding these eigenpairs is equivalent to finding the vectors that are critical points of Rayleigh quotient

$$\frac{\mathbf{v}^\top \mathbf{A}\mathbf{v}}{\mathbf{v}^\top \mathbf{B}\mathbf{v}}$$

with the corresponding generalized eigenvalue ρ equal to the value of the quotient. This framework occurs commonly in statistics; when seeking a linear projection of the data many classic methods solve a form of the above generalized eigenvalue problem with $\mathbf{A}$ and $\mathbf{B}$ acting as covariance matrices of the data.

Further notation we use is as follows. For two matrices $\mathbf{A}$ and $\mathbf{B}$, let $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}\big(\mathbf{A}^\top \mathbf{B}\big)$ denotes the trace inner product. For a vector $\mathbf{u} \in \mathbb{R}^d$ and $q \in [0, \infty]$, $\|\mathbf{u}\|_q = \left(\sum_{j=1}^d |u_j|^q\right)^{1/q}$ is the ℓ_q norm if $0 \leq q \leq \infty$; when $q = 0$, $\|\mathbf{u}\|_0$ is the number of nonzero entries of $\mathbf{u}$; when $q = \infty$, $\|\mathbf{u}\|_\infty = \max_{1 \leq j \leq d} |u_j|$. For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we use $\mathrm{tr}(\mathbf{A})$ to denote its trace and A_{ij} for the (i, j)-th element, and for $q \in (0, \infty)$, $\|\mathbf{A}\|_q$ is ℓ_q operator norm, while $\|\mathbf{A}\|_{\mathrm{F}}$ and $\|\mathbf{A}\|_{\max}$ are used to denote the Frobenius norm and the entry-wise maximum norm, respectively. For $q_1, q_2 \in [0, \infty]$, the matrix (q_1, q_2)-pseudonorm $\|\mathbf{A}\|_{q_1, q_2}$ of $\mathbf{A}$ is defined as $\Big\| \big(\|\mathbf{A}_{*1}\|_{q_2}, \|\mathbf{A}_{*2}\|_{q_2}, \ldots, \|\mathbf{A}_{*m}\|_{q_2} \big) \Big\|_{q_1}$, where $\mathbf{A}_{*i}$ denotes the i^{th} column of $\mathbf{A}$. If $J \in [m]$ is a subset of indices of size j, $A_J \in \mathbb{R}^{j \times n}$ is the submatrix given by the j rows with indices in J. We use $\rho_{\max}(\mathbf{A})$ and $\rho_{\min}(\mathbf{A})$ to denote the maximum eigenvalue and minimum eigenvalue, respectively. C, C_1, and C_2 are constants that may vary in different instances of usage.

Graphs. A graph, denoted $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, is a set of nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$, and a set of edges $\mathcal{E} = \{e_1, \ldots, e_m\}$. Nodes commonly represent entities in a data science problem, while the edges give the relations between them, e.g. social media users as nodes and edges representing friendships, or chemical atoms as nodes, and chemical bonds as edges. An edge $e \in \mathcal{E}$ connects two nodes v_e^1, v_e^2, thus e can be represented as (v_e^1, v_e^2), an element of $\mathcal{V} \times \mathcal{V}$. A node v_i is adjacent to another node v_j

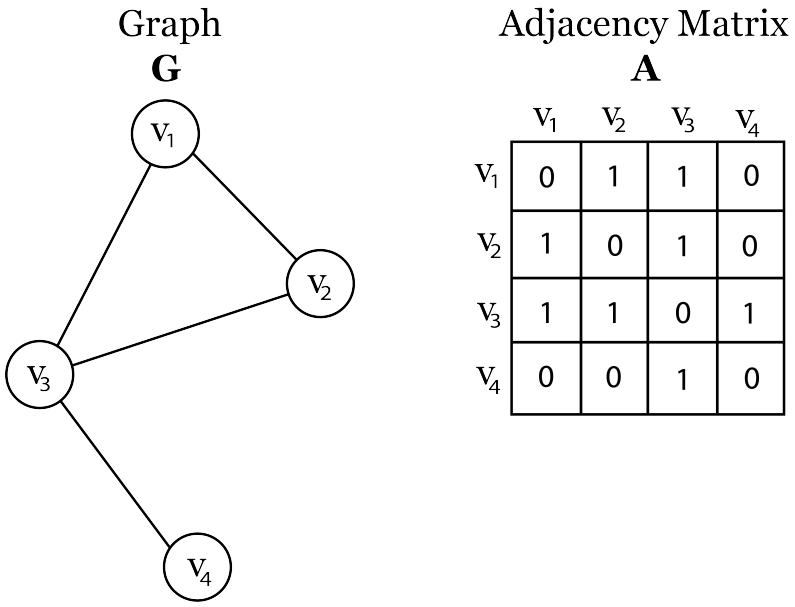

Figure 2.1: A simple graph with adjacency matrix.

if and only if there exists an edge between them. We define the (first-order) neighbors of a node v_i, denoted $\mathcal{N}(v_i)$, as the set of nodes that are adjacent to v_i. A graph $\mathcal{G}$ can be equivalently represented as an adjacency matrix which describes the connectivity between the nodes. Let $\mathbf{A} \in \{0,1\}^{n \times n}$ be a matrix where $\mathbf{A}_{i,j} = 1$ if v_i is adjacent to v_j, and equal to 0 otherwise. The degree of a node v_i is the number of nodes adjacent to v_i,

$$d(v_i) = \sum_{v_j \in \mathcal{V}} \mathbf{1}_{v_j \in \mathcal{N}(v_i)}$$

where $\mathbf{1}_{v_j \in \mathcal{N}(v_i)} = 1$ in the event $v_j \in \mathcal{N}(v_i)$ and 0 otherwise. The degree matrix $\mathbf{D}$ is a diagonal matrix defined as $\mathbf{D}_{i,i} = d(v_i)$, $\mathbf{D}_{i,j} = 0$ if $i \neq j$.

2.2 Deep Learning

Deep learning is a class of machine learning algorithms that are built from artificial neural networks. Originating to a linear model in [MP43], it was further developed into the perceptron in [Ros58], which can learn parameters for the function given training samples. Neural networks (NNs) had a renaissance of interest and research in the early 2000s with the advent of "big data" sources and more powerful computational machines to train the models. Since then deep learning models have consistently proven to outperform state-of-the-art traditional methods in a large

number of applications. The power and flexibility of different deep learning models has firmly established the popularity of the models in machine learning tasks. We introduce the basics of common NN models that we use in our work. A fuller treatment of the subject of deep learning can be found in [GBC16], and [MT21] provides an excellent reference for graphical neural networks.

Feedforward Networks. A feedfoward network is simply a special type of function that is made by composing a collection of simpler functions. As in all machine learning tasks, the feedfoward network is an approximation of a sought after function $f^*()b$, for instance if the task is classification, one wishes to find a mapping $f(\mathbf{x}|\Theta)$ that best approximates the ideal classifier $f^*(\mathbf{x}) = y$. The values of the parameters Θ that determine the feedforward network $f(\boldsymbol{x}|\Theta)$ are learned during training.

In feedfoward networks, the function $f : \mathbb{R}^d \to \mathbb{R}^k$ given by the network is a composition of simpler functions that are referred to as the layers of the network. The output dimension k is chosen to suit the chosen application of the network. A single layer generally has an affine transformation followed by a nonlinear "activation function" applied pointwise. This means for the input vector $\mathbf{x}$, the first layer would produce

$$\mathbf{h}^1 = \sigma\left(\mathbf{b}^1 + \mathbf{W}^1\mathbf{x}\right),$$

where $\mathbf{W}^1 \in \mathbb{R}^{d\times d_1}, \mathbf{b}^1 \in \mathbb{R}^{d_1}$, and $\sigma : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}$ is the activation function. The output $\mathbf{h}^1$ is the first "hidden layer" of the network, and the depth of the network is determined by the number of layers. In general if a network has depth m, then the network function would be

$$f(\mathbf{x}) = \mathbf{b}^m + \mathbf{W}^m\sigma(\mathbf{b}^{m-1} + \mathbf{W}^m\sigma(\cdots\sigma(\mathbf{b}^1 + \mathbf{W}^m\mathbf{x})))),$$

where $\mathbf{W}^i \in \mathbb{R}^{d_i\times d_{i-1}}$ and $\mathbf{b} \in \mathbb{R}^{d_i}$, and our output dimension k would be equal to d_m. The values of the weights of $\mathbf{W}^i$ and $\mathbf{b}^i$, $i \in [m]$ give the collection of parameters Θ that determine the function, and are learned during training. These transformations between layers are depicted in Figure 2.2 as bipartite graphs, where the coordinates before and after are given as left and right nodes, and the edges between them represent the weights of the matrix $\mathbf{W}^i$.

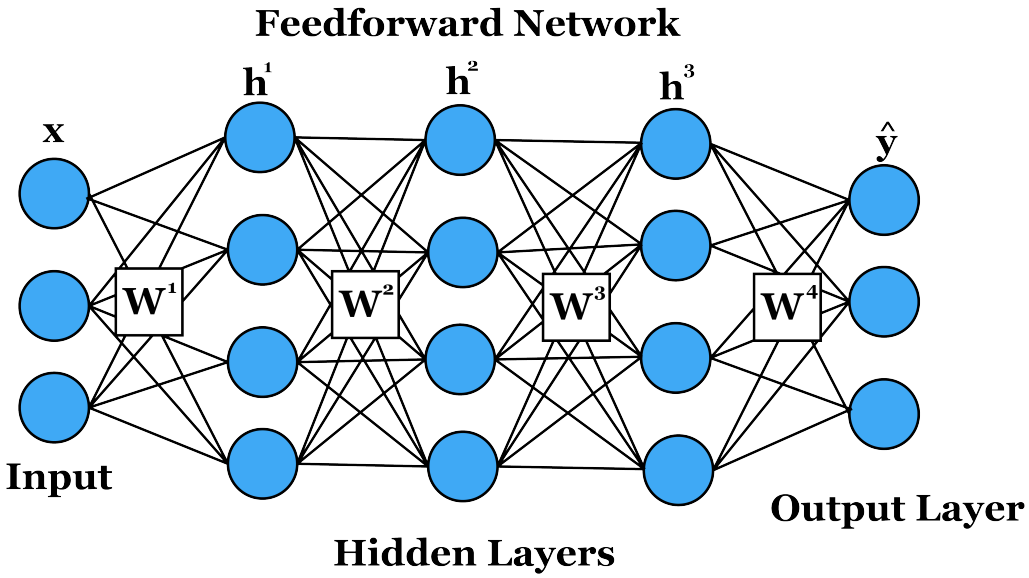

Figure 2.2: The architecture of a feedforward network. The nodes represent the coordinates of each layer, and the edges represent the weights of the linear transformations.

Activation functions were originally designed to recreate the behaviors of biological neurons, which receive a signal and either kill it or propagate it to further neurons. These functions introduce non-linearity into the neural network which leads to strong theorems guaranteeing the function's approximation capabilities under certain conditions. There are a collection of commonly used functions, and one of the most used is the Rectified Linear Unit (ReLU). The ReLU function is linear (identity) for all positive inputs and 0 for all negative values;

$$\text{ReLU}(z) = \max(0, z).$$

Many variants of the function exist, many of which attempt to address the lack of gradient the function has for negative inputs, like the LeakyReLU, ELU, and GELU. Prior to the use of ReLU, using sigmoid functions were the norm, like the logisitic sigmoid

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

due to the belief that activation functions had to be continuous to properly train the model.

The output of the function is run through a chosen loss function that allows for optimization methods like gradient descent to learn the parameters. If the task is regression, so that each training

sample has the pair $(\mathbf{x}, \mathbf{y}), \mathbf{y} \in \mathbb{R}^k$, the output $f(\mathbf{x}) = \hat{\mathbf{y}} \in \mathbb{R}^k$ could be run through the simple squared loss function to measure the difference between the predicted $\hat{\mathbf{y}}$ and ground truth $\mathbf{y}$:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{k} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2.$$

If the task is classification, the neural network needs to output the class label of the input. Instead of a discrete output from a finite set of labels, probabilities are given for each class of C possible classes, so that the output $\hat{\mathbf{y}}$ would be a vector in $\mathbb{R}^C$. The softmax function is used to output values between 0 and 1 so that the total adds up to 1:

$$\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{z})_i = \frac{\exp(\mathbf{z}_i)}{\sum_j \exp(\mathbf{z}_j)}, i \in [C],$$

where $\mathbf{z}_i$ is the i^{th} element of the vector $\mathbf{z}$. Then with the predicted $\hat{\mathbf{y}}$ the loss function of cross-entropy is used to measure the difference from the truth

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} \mathbf{y}_i \log(\hat{\mathbf{y}}_i).$$

Here $\mathbf{y}_i = 1$ if the class of $\mathbf{x}$ is i, and 0 otherwise. During inference, an unlabeled input is given label i if $\hat{\mathbf{y}}_i$ is the largest value among all the coordinates of $\hat{\mathbf{y}}$.

Autoencoders. An autoencoder is a special type of neural network that tries to reproduce its input as its output. The autoencoder consists of two components: an encoder $\mathbf{h} = f(\mathbf{x})$, which encodes $\mathbf{x}$ into a hidden representation (called a code) $\mathbf{h}$, and a decoder g which attempts to reconstruct $\mathbf{x}$ from $\mathbf{h}$, represented $g(\mathbf{h}) = \hat{\mathbf{x}}$. The network is trained to minimize the reconstruction error

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))),$$

where $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ measures the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$. The utility of an autoencoder comes from its limitations; perfectly recreating the input is made to be impossible, so good approximations $\hat{\mathbf{x}}$ are trained by encoding only important information in the code $\mathbf{h}$. This limitation is achieved through the creation of a bottleneck that the input $\mathbf{x}$ is forced through. As seen in Figure 2.3, this bottleneck occurs at the code layer that constrains its representation in some manner.

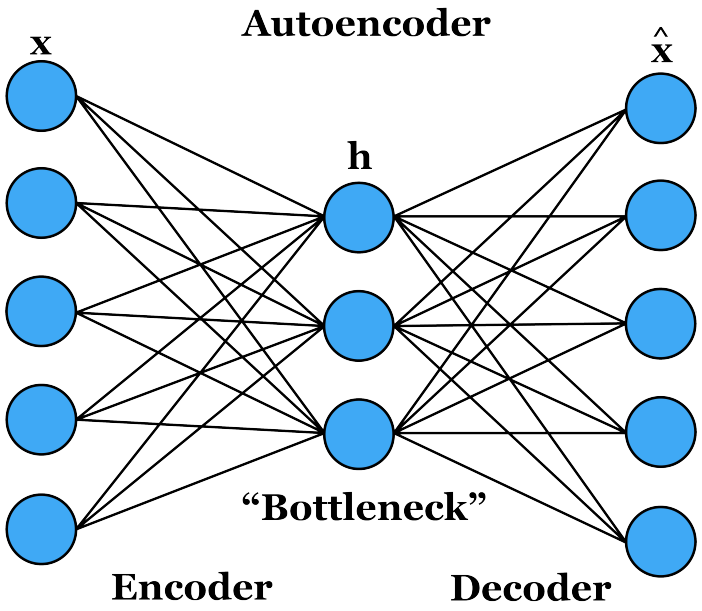

Figure 2.3: The general framework of an autoencoder. The hidden layer gives the code of the encoder, and the lower dimension of the code constrains the representation.

There are two main ways this constraint in the autoencoder is implemented, by making the dimension of the code smaller than the input, or by placing certainly penalties on the latent representation that discourages memorization between input and output. The first leads to undercomplete autoencoders, which forces the lower dimension code $\mathbf{h}$ to preserve the most important features of the input. The second leads to regularized autoencoders, which have additional terms added to the loss function

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))) + \eta \cdot \Omega(\mathbf{h}),$$

where $\Omega(\mathbf{h})$ is the regularization term applied to the code $\mathbf{h}$ and η is a hyperparameter controlling the amount of penalty. One regularization is the ℓ_1-regularization

$$\Omega(\mathbf{h}) = \|\mathbf{h}\|_1$$

which promotes sparsity in the code $\mathbf{h}$. Other regularizations may explicitly promote certain features that are data specific to be preserved in the hidden representation, such as the cell type of of single-cell gene expression data.

Graphical Neural Networks. Graph neural networks (GNNs) are a collection of deep learning architectures that are designed to deal with graph-structured data. Other architectures like feedforward networks or convolutional neural networks (not covered) are more amenable to data that is structured as a regular grid, like vectors or matrices. GNNs expand the functionality of neural networks to to this more multifarious data, and allow for both node-based and graph-based learning tasks. These models are quite recent innovations, as the first GNN model was published as recently as 2005 in [SYG+05].

Like all neural networks, GNNs act as a type of representation learning of its input data, and it is through learning a good representation of its input data that it is able to perform well in designated tasks. Since the input data for GNNs are graphs, there are two ways to go about this representation. For node-based tasks, the GNN learns good features for each node, using the graph structure to facilitate the calculation of this representation. For graph-based tasks, the GNN aims to learn features to represent the entire graph, and learning node features occurs only as an intermediate ancillary step.

To learn updated node features on the graph, the GNN takes in both the input node features and the graph structure given by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, where n gives the number of nodes. If the nodes features are given by vectors in $\mathbb{R}^d$, the collection of features can be given by a matrix $\mathbf{F} \in \mathbb{R}^{n\times d}$, and the update takes the form

$$\mathbf{F}^1 = h(\mathbf{A}, \mathbf{F})$$

for some function h called the "graph filter". For node-based tasks, the graph filtering operation is usually sufficient, and the GNN consists of multiple graph filters stacked consecutively to generate final node features. Other operations are necessary for graph-based tasks to generate the features for the entire graph from the node features. Tailored functions that take into account the graph structure like graph pooling operations are used to generate global features. We forego further discussion of graph-based task models, since our applications are node-based only.

Like feedforward networks, the general framework for node-based GNNs is a composition of multiple steps of graph filtering followed by a non-linear activation. If the network has depth m,

then the collection of operations would be denoted

$$\mathbf{F}^m = h_m(\mathbf{A}, \sigma(h_{m-1}(\mathbf{A}, \cdots \sigma(h_1(\mathbf{A}_1, \mathbf{F}))))).$$

The final output $\mathbf{F}^m$ is then used for a downstream task related, e.g. classification on the nodes. The non-linear activation functions come from the same collection of activation functions used for other neural networks, but they can be combined with the graph filterings in novel ways. The spectral filtering process, for instance, has the nodes transformed via a Graph Fourier Transform, applies the activation function to the transformed coefficients, and then reconstructs the nodes from the spectral representation. While there are a large collection of graph filters, including a whole class of spectral-based filters, we focus here on the simplest spatial-based graph filter.

Let our filter h_i, followed by the activation function σ, be defined as

$$\sigma(h_i(\mathbf{A}, \mathbf{F}_{i-1})) = \sigma\left(\mathbf{A}\mathbf{F}_{i-1}\mathbf{W}^i\right),$$

for $\mathbf{W}^i \in \mathbb{R}^{d_{i-1} \times d_i}$. This transformation is the same as the feedforward network with the exception of the multiplication by $\mathbf{A}$. This product $\mathbf{A}\mathbf{F}_{i-1}$ means that for every node, we sum up all the feature vectors of all the neighboring nodes but not the node itself (unless there are self-loops in the graph.) This allows the topology of the graph to be taken into account during these transformations, but one generally wishes for a node to propagate its own features into its next updated representation. To correct for this, the adjacency matrix is replaced by $\widehat{\mathbf{A}} = \mathbf{A} + I$ for I the identity matrix in $\mathbb{R}^n$, so that $\widehat{\mathbf{A}}$ supplies self-loops to the graph. Furthermore, since $\mathbf{A}$ is typically not normalized, the product $\mathbf{A}\mathbf{F}_{i-1}$ can change the scale of the feature nodes based on the number of neighbors a node has. One can normalize $\mathbf{A}$ via multiplication with the inverse degree matrix $\mathbf{D}^{-1}$, but more often a symmetric normalization is used giving $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. Combining these two we get

$$\sigma(h_i(\mathbf{A}, \mathbf{F}_{i-1})) = \sigma\left(\widehat{\mathbf{D}}^{-1/2}\widehat{\mathbf{A}}\widehat{\mathbf{D}}^{-1/2}\mathbf{F}_{i-1}\mathbf{W}^i\right),$$

for $\widehat{\mathbf{D}}$ the degree matrix of $\widehat{\mathbf{A}}$.

Training. The training of deep learning models occurs through the minimization of a loss function $\mathcal{L}$ with respect to the parameters of the models – the weights of the affine transformations. We

denote the loss function as $\mathcal{L}(\Theta)$ where Θ denotes all the parameters to be optimized. Gradient descent, a first-order iterative optimization algorithm, is often used to minimize the loss function. At each iteration the parameters Θ are updated by shifting them in the direction of the negative gradient (the direction the loss function decreases the most at that location):

$$\Theta' = \Theta - \eta \cdot \nabla_\Theta \mathcal{L}(\Theta),$$

where $\nabla_\Theta \mathcal{L}(\Theta)$ denotes the gradient of $\mathcal{L}$ at Θ, and $\eta > 0$ is the learning rate, which is usually fixed at a small constant. The gradient is usually averaged over a collection of training samples in a batch, which provides greater statistical consistency and computational efficiency. The process is iterated until convergence or some condition is met.

2.3 Single-Cell Data

In the biological sciences, the advent of single-cell technologies has revolutionized the investigation of cellular behavior in the context of its microenvironment. Single-cell sequencing is able to measure multiple molecular features in multiple modalities in a cell, such as gene expressions, protein abundance and chromatin accessibility. Measurements at the single-cell level allow for unprecedented resolution for studying cell-to-cell heterogeneity. Such data sheds new insights across biological disciplines including oncology [LYS18], neurology [RWM$^+$18], and immunology [SDB$^+$18]. Technologies like single-cell transcriptome sequencing (scRNA-seq) and single-cell assay for transposase-accessible chromatin with sequencing (scATAC-seq) provide data on the RNA and DNA gene expression of individual cells respectively. scRNA-seq data makes it possible to measure transcriptome-wide gene expression, and enables researchers to distinguish different cell types based on their gene expression, organize cell populations, and identify cells transitioning between states [AHB$^+$16, HBR$^+$17, Con18]. Similarly, scATAC-seq studies can reveal somatic clonal structures such as those found in cancer [ZNC$^+$19], helping monitor cell lineage development.

The technologies that produce these data sources are very interesting in their own right, but for our purposes it suffices to understand the format this data is presented to us after the sampling

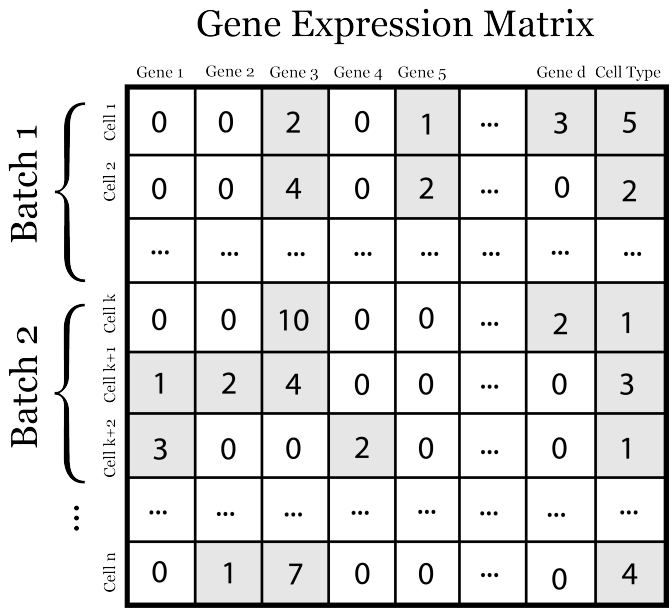

Figure 2.4: An example matrix for single-cell data.

takes place. For both scRNA-seq and scATAC-seq, the data can be simply represented as a $n \times d$ matrix where n gives the number of cells sampled and d gives the "sequencing depth", i.e. the number of genes tested for in the study. Then each row of the matrix gives a unique cell, and each column give a unique gene. The values of the matrix consist of non-negative integers giving the count data of how many times the gene expressed for a particular cell. These matrices can have further information supplementing the data, such as cell type annotations assigning a class to each cell, or cell-cycle scores quantifying the developmental stage of the cell. The cell-cycle scoring is based on the expression of G2/M and S phase markers, where S is the synthesis phase for the replication of the chromosomes (also part of interphase), G2 is the gap 2 phase representing the end of interphase, prior to entering the mitotic phase, and the M phase is the nuclear division of the cell (consisting of prophase, metaphase, anaphase and telophase) [NHS+16]. This additional information would be included as a collection of additional columns at the end of the matrix, indicating the appropriate labels or scores. An example matrix giving the gene expression data of a single-cell experiment is given in Figure 2.4.

There are two important aspects of single-cell data that are necessary to understand before using it for analysis: batch effects and dropout. In single-cell sequencing methods, data is organized

into separate batches, where large groups of cells are potentially sampled in multiple laboratories using different cell dissociation and handling protocols, library preparation technologies and/or sequencing platforms. These different factors result in batch effects [TBH+17] that can change the expression of genes systematically from one batch to another. Such differences can mask underlying biology or introduce spurious structure in the data, and must be corrected prior to further analysis to avoid misleading conclusions [AHMM18]. Since becoming a standard data preprocessing step for singe-cell analysis, there have been many advanced techniques developed to address batch effect removal, including techniques based on CCA [ZWT19] and a number of deep learning models [LWL+20, SSZ+17].

Dropout is the name given to the technical error that occurs when performing single-cell sampling, which leads to artificial counts of zero in the gene expression read outs. The error occurs during library preparation – a technical step that duplicates a gene many times in order to be counted during sampling – and occurs more frequently for genes that express at low levels in their respective cell types [Qui20]. This leads to sequencing data that is notoriously sparse, where the vast majority of features may be zero in a typical dataset due to dropout [SNL+17], which can be even more pronounced in multi-modal data [LHH20]. Dropout events lead to increased technical variability and noise in the single-cell data, and makes it difficult to differentiate "true" zero counts from "false" ones. Here, true zero counts indicate that a gene is not expressed in a particular cell type, which could act as important information to differentiate cell types. Addressing dropout requires specialized data processing methods such as imputation. Imputation takes in data and attempts to replace artificial zero counts with realistic count values, hopefully while preserving true zero counts. A diverse collection of methods exist for imputation, most of which rely on gaining information about cell behaviors from similar cells in the dataset, or by transferring knowledge of cell behaviors from other datasets. Methods like Phenograph [LSB+18], MAGIC [vDSN+18], and Seurat [BHS+18] use K-nearest neighbor (KNN) graphs to model relationship between cells, but the high sparsity of the data may cause these neighborhood estimates to be unreliable, and may over-simplify the complex cell and gene relationships of cell population. Deep learning methods

have improved on these [WAH+19, APYG19], with the top performing methods using GNNs, such as GraphSCI [RZL+21] and scGNN [WMC+21].

CHAPTER 3

THEORETICAL PROPERTIES OF THE GEV ESTIMATOR

Let $\boldsymbol{x}$ be a d-dimensional random vector of covariates with covariance matrix $\boldsymbol{\Sigma}$. Let y be a one-dimensional continuous response and let $\boldsymbol{x} \mid y$ be a d-dimensional random vector of covariates and $\boldsymbol{\Omega} = \text{cov}(\mathbb{E}[\boldsymbol{x}|y])$. It is advantageous to find a linear projection of $\boldsymbol{x}$ to a subspace of dimension $K \ll d$ such that the population centroids, $\mathbb{E}[\boldsymbol{x}|y]$, separate the most in the projected space. This amounts to finding the vectors that maximize $\mathbf{v}^\top \boldsymbol{\Omega} \mathbf{v}$, but minimize overlap of the data after projection; i.e. minimize $\mathbf{v}^\top \text{cov}(\mathbf{x}|y)\mathbf{v}$. Assuming $\text{cov}(\boldsymbol{x}|y)$ is the same for all y, we can consider the following optimization procedure by maximizing a sequence of Rayleigh quotients:

$$\mathbf{v}_k^* = \underset{\mathbf{v}_k}{\text{argmax}} \frac{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_k}, \text{ s.t. } \mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_j = 0, \text{ for all } 1 \leq j < k \leq K. \tag{3.1}$$

where, in classification problems with y taking discrete values (such as LDA), $\boldsymbol{\Omega}$ and $\boldsymbol{\Sigma}$ are frequently referred to as the "between-class" and "within-class" covariance matrices, respectively.[1] It is easily verified that the quotient $\frac{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_k}$ has critical points for each generalized eigenvector of $(\boldsymbol{\Omega}, \boldsymbol{\Sigma})$, so that $\mathbf{v}_k^*$ is a critical point if $\boldsymbol{\Omega} \mathbf{v}_k^* = \rho \boldsymbol{\Sigma} \mathbf{v}_k^*$ for some $\rho \in \mathbb{R}$. If $\boldsymbol{\Sigma}$ is invertible, the problem reduces to finding eigenvectors of $\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}$, but since estimates of $\boldsymbol{\Sigma}$ are singular in the high-dimensional regime, other means of solving the problem have to be used.

Thus the first projection vector is sought to maximize the between-class covariance relative to the within-class covariance. Then it seeks the second projection vector that maximizes the between-class covariance subject to the constraint that it is orthogonal to the first projection direction with respect to $\boldsymbol{\Sigma}$. This procedure is then continued up to K times, where K is chosen to fully capture the signal. In this work we focus on the case $K = \text{rank}(\boldsymbol{\Omega})$, where in the ap-

[1] Note that the actual within-class covariance would be $\text{cov}(\boldsymbol{x}|y) = \Sigma_{\boldsymbol{x}|y}$. However, with the assumption of homoscedasticity, that $\Sigma_{\boldsymbol{x}|y} = \Sigma_{\boldsymbol{x}|y'}$ for all y, y', we have the pooled covariance Σ_P equal to any within-class covariance, and with the law of total covariance $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_P + \boldsymbol{\Omega}$ we have the equivalence of

$$\underset{\mathbf{v}_k}{\text{argmax}} \frac{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_k} = \underset{\mathbf{v}_k}{\text{argmax}} \frac{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Sigma}_P \mathbf{v}_k + \mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k} = \underset{\mathbf{v}_k}{\text{argmax}} \frac{1}{\frac{\mathbf{v}_k^\top \boldsymbol{\Sigma}_P \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k} + 1} = \underset{\mathbf{v}_k}{\text{argmax}} \frac{\mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k}{\mathbf{v}_k^\top \boldsymbol{\Sigma}_{\boldsymbol{x}|y} \mathbf{v}_k}.$$

plications we consider $\text{rank}(\boldsymbol{\Omega}) \ll d$. Then the K projection vectors are concatenated to obtain $\mathbf{V}_K = \{\mathbf{v}_1^*, \ldots, \mathbf{v}_K^*\}$, and the projection space, $\mathcal{V}_K$, is obtained by setting $\mathcal{V}_K = \text{span}\{\mathbf{V}_K\}$, the space spanned by the linear combinations of the K projection vectors.

However as it stands, the above approach is undesirable from both a computational standpoint as well as from an estimation perspective. Each subsequent projection vector relies on all estimates of previous projection directions. Thus, propagation of the estimation error is possible. In addition, the corresponding optimization problems are nonconvex, hence, the convergence of any optimization algorithms to the global optima is not assured. This computational intractability poses additional theoretical challenges and thus, most methods that are based on (3.1) do not have theoretical guarantees. We reformulate (3.1) such that the projection space $\mathcal{V}_K$ can be recovered in a simultaneous manner via a convex optimization problem we call the Generalized EigenValue (GEV) projection. Using a sparsity assumption, we formulate the proposed GEV procedure into an optimization analogous to a type of matrix lasso regression problem.

3.1 General Error Bound

Without loss of generality, we assume that $\bar{\boldsymbol{\mu}} \equiv \mathbb{E}(\boldsymbol{x}) = \mathbf{0}$. If $\bar{\boldsymbol{\mu}} \neq \mathbf{0}$, we can center the data via $\boldsymbol{x}' \equiv \boldsymbol{x} - \mathbb{E}(\boldsymbol{x})$, taking $\boldsymbol{x}'$ as our centralized data with mean $\mathbf{0}$. Let $\boldsymbol{\Omega}$ have an eigendecomposition

$$\boldsymbol{\Omega} = \text{var}\{\mathbb{E}[\boldsymbol{x}|y]\} = \sum_{k=1}^{K} \rho_k \boldsymbol{u}_k \boldsymbol{u}_k^\top = \mathbf{U}\mathbf{U}^\top$$

where $\mathbf{U} = (\sqrt{\rho_1}\boldsymbol{u}_1, \ldots, \sqrt{\rho_K}\boldsymbol{u}_K) \in \mathbb{R}^{d\times K}$. Let $\mathbf{W}^*$ be the solution to the following optimization problem

$$\mathbf{W}^* = \underset{\mathbf{W}\in\mathbb{R}^{d\times K}}{\text{argmin}} \left\{ \frac{1}{2} \left\| \boldsymbol{\Sigma}^{1/2}\mathbf{W} - \boldsymbol{\Sigma}^{-1/2}\mathbf{U} \right\|_{\text{F}}^2 \right\}. \tag{3.2}$$

Theorem 1. *Let $\mathcal{W} = \text{span}\{\mathbf{W}^*\}$. Then we have $\mathcal{V}_K = \mathcal{W}$.*

Proof. Let $Q(\mathbf{W}) = \frac{1}{2}\left\| \boldsymbol{\Sigma}^{1/2}\mathbf{W} - \boldsymbol{\Sigma}^{-1/2}\mathbf{U} \right\|_{\text{F}}^2$. Then $\mathbf{W}^*$ is the minimizer of $Q(\mathbf{W})$ and $\mathbf{W}^*$ satisfies

$$\nabla Q(\mathbf{W}^*) = \boldsymbol{\Sigma}\mathbf{W}^* - \mathbf{U} = 0,$$

which yields $\mathbf{W}^* = \boldsymbol{\Sigma}^{-1}\mathbf{U}$.

To proceed, we need the following two lemmas. Our first lemma concerns the relation between the eigenpairs of $\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}$ and those of $\mathbf{\Sigma}^{-1}\mathbf{A}$.

Lemma 2. *Let* $\mathbf{A}$ *be a symmetric matrix,* $\mathbf{\Sigma}$ *symmetric positive definite. If* $(\rho, \mathbf{v})$ *is an eigenpair of* $\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}$*, then* $(\rho, \mathbf{\Sigma}^{-1/2}\mathbf{v})$ *is an eigenpair of* $\mathbf{\Sigma}^{-1}\mathbf{A}$*; and vice versa.*

Our next lemma connects the sequential optimization problem (3.1) to the eigenpairs of $\mathbf{\Sigma}^{-1}\mathbf{\Omega}$.

Lemma 3. *The eigenvectors of* $\mathbf{\Sigma}^{-1}\mathbf{\Omega}$ *corresponding to the nonzero eigenvalues solves* (3.1)*.*

Using Lemma 3, we only need to show that $\mathbf{W}^*$ is equal to the eigenvectors of $\mathbf{\Sigma}^{-1}\mathbf{\Omega}$ up to an orthogonal transformation. Suppose we have the following eigendecomposition of $\mathbf{U}^\top\mathbf{\Sigma}^{-1}\mathbf{U}$:

$$\mathbf{U}^\top\mathbf{\Sigma}^{-1}\mathbf{U} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top.$$

Then left-multiplying both sides by $\mathbf{\Sigma}^{-1}\mathbf{U}$ and right-multiplying $\mathbf{P}$, we have

$$\mathbf{\Sigma}^{-1}\mathbf{U}\mathbf{U}^\top\mathbf{\Sigma}^{-1}\mathbf{U}\mathbf{P} = \mathbf{\Sigma}^{-1}\mathbf{U}\mathbf{P}\mathbf{\Lambda},$$

or equivalently

$$\mathbf{\Sigma}^{-1}\mathbf{\Omega}\mathbf{W}^*\mathbf{P} = \mathbf{W}^*\mathbf{P}\mathbf{\Lambda}.$$

Using the fact that $\mathcal{V} = \text{span}\{\mathbf{W}^*\mathbf{P}\} = \text{span}\{\mathbf{W}^*\} = \mathcal{W}$ finishes the proof. □

3.1.1 Proof of Lemma 2

Proof. Suppose $(\rho, \mathbf{v})$ is an eigenpair of $\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}$, then

$$\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}\mathbf{v} = \rho\mathbf{v}. \tag{3.3}$$

Multiplying both sides in (3.3) by $\mathbf{\Sigma}^{-1/2}$, we obtain that $(\rho, \mathbf{\Sigma}^{-1/2}\mathbf{v})$ is an eigenpair of $\mathbf{\Sigma}^{-1}\mathbf{A}$; and vice versa. Since $\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}$ and $\mathbf{\Sigma}^{-1}\mathbf{A}$ have the same rank (since $\mathbf{\Sigma}$ is full rank), we further conclude that the eigenpairs of $\mathbf{\Sigma}^{-1/2}\mathbf{A}\mathbf{\Sigma}^{-1/2}$ and those of $\mathbf{\Sigma}^{-1}\mathbf{A}$ have a one-to-one correspondence with the same eigenvalues. □

3.1.2 Proof of Lemma 3

Proof. We rewrite problem (3.1) as:

$$\begin{aligned}
\mathbf{u}_1^* &= \text{argmax}\, \frac{\mathbf{u}_1^\top \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Omega}\boldsymbol{\Sigma}^{-1/2}\mathbf{u}_1}{\mathbf{u}_1^\top \mathbf{u}_1}, \\
\mathbf{u}_k^* &= \text{argmax}\, \frac{\mathbf{u}_k^\top \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Omega}\boldsymbol{\Sigma}^{-1/2}\mathbf{u}_k}{\mathbf{u}_k^\top \mathbf{u}_k} \text{ s.t. } \mathbf{u}_k \perp \mathbf{u}_j = 0, \forall 1 \leq j < k, \forall 1 \leq k \leq K, \\
\mathbf{v}_k^* &= \boldsymbol{\Sigma}^{-1/2}\mathbf{u}_k^*,
\end{aligned}$$

Applying Lemma 2 finishes the proof of Lemma 3. □

Formulation (3.2) resembles the least square loss in linear models, and the loss function in (3.2) can be regarded as the matrix version of least square loss. Despite its simpleness, it recovers the same projection space as produced by (3.1). Note that any estimator function $Q(\mathbf{W})$ that satisfies $\nabla Q(\mathbf{W}) = \boldsymbol{\Sigma}\mathbf{W} - U$ will recover $\mathcal{V}$. We will exploit this later for an alternative algorithm that makes use of Huber norm regularlization for noisy data. As it stands, however, the estimator $Q(\mathbf{W})$ is not able to statistically recover $\mathbf{W}^*$ in the paradigm of high-dimensional data. To handle high-dimensional features, we impose an assumption of sparsity on the structure on $\mathbf{W}$ and propose to solve the following penalized regression problem:

$$\underset{\mathbf{W}}{\text{argmin}} \Big\{ \frac{1}{2} \text{tr}\left(\mathbf{W}^\top \boldsymbol{\Sigma}\mathbf{W}\right) - \text{tr}\left(\mathbf{W}^\top \mathbf{U}\right) + \lambda \|\mathbf{W}\|_{1,1} \Big\}. \tag{3.4}$$

The first two terms above are just an expansion of $\frac{1}{2}\left\|\boldsymbol{\Sigma}^{1/2}\mathbf{W} - \boldsymbol{\Sigma}^{-1/2}\mathbf{U}\right\|_{\text{F}}^2$.

Let $\widehat{\boldsymbol{\Sigma}}$ and $\widehat{\mathbf{U}}$ be the estimates of $\boldsymbol{\Sigma}$ and $\mathbf{U}$, respectively. Plugging these estimates into (3.4) above, we obtain the sample version

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\text{argmin}} \Big\{ \frac{1}{2} \text{tr}\left(\mathbf{W}^\top \widehat{\boldsymbol{\Sigma}}\mathbf{W}\right) - \text{tr}\left(\mathbf{W}^\top \widehat{\mathbf{U}}\right) + \lambda \|\mathbf{W}\|_{1,1} \Big\}. \tag{3.5}$$

Let $\mathcal{S} = \{(i,j) : w_{i,j}^* \neq 0\}$ be the total support set of $\mathbf{W}^*$ and assume $\mathbf{W}^*$ is s-sparse, that is $|\mathcal{S}| = s$. To give the main theoretical result for the estimation error, we need the following definition of the generalized restricted eigenvalue (GRE) for matrices and the corresponding GRE condition.

Definition 4 (Generalized Restricted Eigenvalue for Matrices)**.** *Let $K, m \in \mathbb{N}$, and $\gamma \in \mathbb{R}$. The generalized restricted eigenvalue (GRE) for matrices is defined as*

$$\begin{aligned} \kappa_+(K,m,\gamma) &= \sup_{\mathbf{V}} \left\{ \operatorname{tr}\left(\mathbf{V}^\top \widehat{\mathbf{\Sigma}} \mathbf{V}\right) / \|\mathbf{V}\|_{\mathrm{F}}^2 : \mathbf{V} \in \mathcal{C}(K,m,\gamma) \right\}, \\ \kappa_-(K,m,\gamma) &= \inf_{\mathbf{V}} \left\{ \operatorname{tr}\left(\mathbf{V}^\top \widehat{\mathbf{\Sigma}} \mathbf{V}\right) / \|\mathbf{V}\|_{\mathrm{F}}^2 : \mathbf{V} \in \mathcal{C}(K,m,\gamma) \right\}, \end{aligned} \tag{3.6}$$

where $\mathcal{C}(K,m,\gamma) = \left\{ \mathbf{V} \in \mathbb{R}^{d \times K} : \mathcal{S} \subseteq J, |J| \leq m, \|\mathbf{V}_{J^c}\|_{1,1} \leq \gamma \|\mathbf{V}_J\|_{1,1} \right\}$ and $J \subset [d]$.

Assumption 5. *There exists $K, m \in \mathbb{N}$ and $\gamma \in \mathbb{R}$ and constants $\kappa_*, \kappa^* \in \mathbb{R}$ such that*

$$0 < \kappa_* \leq \kappa_-(K,m,\gamma) \leq \kappa_+(K,m,\gamma) \leq \kappa^* < \infty.$$

The assumption above is necessary for our theoretical development and was first proposed by [BRT09] for the vector case and various versions are standard in high-dimensional estimation literature. Our definition is a direct extension of theirs to the matrix case.

Define $\mathcal{L}(\mathbf{W}) \equiv \frac{1}{2} \operatorname{tr}\left(\mathbf{W}^\top \widehat{\mathbf{\Sigma}} \mathbf{W}\right) - \operatorname{tr}\left(\mathbf{W}^\top \widehat{\mathbf{U}}\right)$ as our cost function without the regularization term. As well, assume that there exists $M, \rho > 0$ such that $1/M \leq \rho_{\min}(\mathbf{\Sigma}) \leq \rho_{\max}(\mathbf{\Sigma}) \leq M$ and $\rho \leq \rho_K(\mathbf{\Omega})$. We are ready to state our first result on the estimation error, which concerns the performance of $\widehat{\mathbf{W}}$, under the event that $\left\{ \|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq \lambda/2 \right\}$.

Theorem 6. *Assume that Assumption 5 holds with $k = K$, $m = s$, $\gamma = 3$. Suppose that $\left\|\nabla \mathcal{L}(\mathbf{W}^*)\right\|_{\infty,\infty} \leq \lambda/2$. Then we have*

$$\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} \leq 3M\kappa_*^{-1}\rho^{-1}\lambda\sqrt{s}. \tag{3.7}$$

Remark. *The theorems above follow a general method of giving error bounds on M-estimators. M-estimators are a family of estimators that combine a cost function with a regularizer, which require two properties for consistency in high dimensions: the decomposibility of the regularizer and restricted strong convexity of their respective cost function. See [Wai19] chapter 9 for an explanation of these methods. Note that by conditioning on the random event $\mathbb{G}(\lambda) = \left\{ \|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq \lambda/2 \right\}$ these theorems give deterministic bounds; it is by further specifying the application-dependent structure of $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ that leads to probabilistic bounds. In all applications this will yield a statement of the form $\mathbb{G}(\lambda)$ holds with high probability which identifies $\lambda = f(n,d)$ with a function of the sample size and dimension of the data.*

3.1.3 Proof of Theorem 6

Proof. Let $\mathcal{S} = \bigcup_{j=1}^{d} \mathcal{S}_j$ be the union of supports for each projection direction. We first need the following lemma.

Lemma 7. *Suppose that* $\|\nabla \mathcal{L}(\mathbf{W}^*)\|_\infty \leq \lambda/2$. *For a* $\mathcal{E}$ *such that* $\mathcal{S} \subseteq \mathcal{E}$ *and* $\|\mathcal{E}\|_0 \leq 2s$, *we have*

$$\big\|\big(\widehat{\mathbf{W}} - \mathbf{W}^*\big)_{\mathcal{E}^c}\big\|_{1,1} \leq 3\big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\big\|_{1,1}.$$

3.1.3.1 Proof of Lemma 7

Proof. By the mean value theorem, there exists a $\widetilde{\mathbf{W}}$, some convex combination of $\widehat{\mathbf{W}}$ and $\mathbf{W}^*$, such that $\nabla \mathcal{L}\big(\widehat{\mathbf{W}}\big) - \nabla \mathcal{L}\big(\mathbf{W}^*\big) = \mathbf{H}(\widehat{\mathbf{W}} - \mathbf{W}^*)$, where $\mathbf{H} = \nabla^2 \mathcal{L}(\widetilde{\mathbf{W}}) \in \mathbb{R}^{p\times K\times p\times K}$ is a forth-order tensor

$$\mathbf{H} = \begin{pmatrix} \nabla \frac{\partial \mathcal{L}(\widetilde{\mathbf{W}})}{\partial w_{11}} & \cdots & \nabla \frac{\partial \mathcal{L}(\widetilde{\mathbf{W}})}{\partial w_{1K}} \\ \vdots & & \vdots \\ \nabla \frac{\partial \mathcal{L}(\widetilde{\mathbf{W}})}{\partial w_{d1}} & \cdots & \nabla \frac{\partial \mathcal{L}(\widetilde{\mathbf{W}})}{\partial w_{dK}} \end{pmatrix}.$$

The tensor-matrix product is defined as $\mathbf{HW} = (a_{ij})_{p\times K} \in \mathbb{R}^{d\times K}$, with $a_{ij} = \langle \nabla \partial \mathcal{L}(\widetilde{\mathbf{W}})/\partial w_{ij}, \mathbf{W}\rangle$. Let $\widehat{\mathbf{\Gamma}}$ be some sub-gradient matrix of $\|\mathbf{W}\|_{1,1}$ evaluated at $\widehat{\mathbf{W}}$. The Karush-Tuhn-Tucker (KKT) conditions of $\widehat{\mathbf{W}}$ imply

$$\begin{aligned} 0 &= \big\langle \nabla \mathcal{L}\big(\widehat{\mathbf{W}}\big) + \lambda \widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle \\ &= \big\langle \nabla \mathcal{L}\big(\widehat{\mathbf{W}}\big) - \nabla \mathcal{L}\big(\mathbf{W}^*\big) + \nabla \mathcal{L}\big(\mathbf{W}^*\big) + \lambda \widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle \\ &= \big\langle \mathbf{H}\big(\widehat{\mathbf{W}} - \mathbf{W}^*\big) + \nabla \mathcal{L}\big(\mathbf{W}^*\big) + \lambda \widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle \end{aligned}$$

Using the positive semi-definiteness of the quadratic form $\big\langle \mathbf{H}\big(\widehat{\mathbf{W}} - \mathbf{W}\big), \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle$, we further have

$$0 \leq -\underbrace{\big\langle \nabla \mathcal{L}(\mathbf{W}^*), \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle}_{\mathrm{I}_1} - \underbrace{\big\langle \lambda \widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle}_{\mathrm{I}_2}. \tag{3.8}$$

Next, we bound I_1 and I_2 respectively. Applying Hölder inequality to I_1 obtains us that

$$\big\langle \nabla \mathcal{L}(\mathbf{W}^*), \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle \geq -\big\|\nabla \mathcal{L}(\mathbf{W}^*)\big\|_{\infty,\infty} \big\|\widehat{\mathbf{W}} - \mathbf{W}^*\big\|_{1,1}.$$

For I_2, separating the support of $\widehat{\mathbf{\Gamma}}$ and $\widehat{\mathbf{W}} - \mathbf{W}^*$ into $\mathcal{E}$ and $\mathcal{E}^c$, using the assumption $\mathcal{E}^c \cap \mathcal{S} = \emptyset$, we have $\mathbf{W}^*_{\mathcal{E}^c} = \mathbf{0}$ and thus

$$\begin{aligned}\big\langle \widehat{\mathbf{\Gamma}}_{\mathcal{E}^c}, (\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c} \big\rangle &= \big\langle \mathbf{1}_{\mathcal{E}^c}, \big|\widehat{\mathbf{W}}_{\mathcal{E}^c}\big| \big\rangle \\ &= \big\langle \mathbf{1}_{\mathcal{E}^c}, \big|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c}\big| \big\rangle \\ &= \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c}\|_{1,1}\end{aligned}$$

since $\widehat{\mathbf{\Gamma}}_{ij} = \operatorname{sign}(\widehat{\mathbf{W}}_{ij})$ when $\widehat{\mathbf{W}}_{ij} \neq 0$. On the other hand, we have

$$\begin{aligned}\langle \widehat{\mathbf{\Gamma}}_{\mathcal{E}}, (\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}} \rangle &\geq -\|\widehat{\mathbf{\Gamma}}_{\mathcal{E}}\|_{\infty,\infty} \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\|_{1,1} \\ &= \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\|_{1,1}\end{aligned}$$

by the Hölder inequality and the identity $\|\widehat{\mathbf{\Gamma}}_{\mathcal{E}}\|_{\infty,\infty} = 1$. Plugging the derived inequalities above, we obtain

$$\begin{aligned}\big\langle \lambda \widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \big\rangle &= \big\langle \lambda \widehat{\mathbf{\Gamma}}_{\mathcal{E}^c}, (\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c} \big\rangle + \big\langle \lambda \widehat{\mathbf{\Gamma}}_{\mathcal{E}}, (\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}} \big\rangle \\ &\geq \lambda \big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c}\big\|_{1,1} - \lambda \big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\big\|_{1,1}.\end{aligned}$$

Plugging the bounds for I_1 and I_2 back into (3.8) yields that

$$\begin{aligned}0 &\leq \big\|\nabla \mathcal{L}(\mathbf{W}^*)\big\|_{\infty,\infty} \big\|\widehat{\mathbf{W}} - \mathbf{W}^*\big\|_{1,1} - \lambda \big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c}\big\|_{1,1} + \lambda \big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\big\|_{1,1} \\ &\leq -\big(\lambda - \big\|\nabla \mathcal{L}(\mathbf{W}^*)\big\|_{\infty,\infty}\big) \big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}^c}\|_{1,1} \\ &\quad + \big(\lambda + \big\|\nabla \mathcal{L}(\mathbf{W}^*)\big\|_{\infty,\infty}\big) \big\|\big(\widehat{\mathbf{W}} - \mathbf{W}^*\big)_{\mathcal{E}}\big\|_{1,1},\end{aligned}$$

which further yields that

$$\big\|\big(\widehat{\mathbf{W}} - \mathbf{W}^*\big)_{\mathcal{E}^c}\big\|_{1,1} \leq \frac{\lambda + \big\|\nabla \mathcal{L}\big(\mathbf{W}^*\big)\big\|_{\infty,\infty}}{\lambda - \big\|\nabla \mathcal{L}\big(\mathbf{W}^*\big)\big\|_{\infty,\infty}} \big\|\big(\widehat{\mathbf{W}} - \mathbf{W}^*\big)_{\mathcal{E}}\big\|_{1,1} \leq 3\big\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{E}}\big\|_{1,1},$$

which completes the proof. □

Taking $\mathcal{E} = \mathcal{S}$ in Lemma 7 and using the restrictive eigenvalue condition with lower bound $\kappa_-(K, s, 3) \geq \kappa_*$, we obtain

$$\kappa_* \big\|\widehat{\mathbf{W}} - \mathbf{W}^*\big\|_{2,2}^2 \leq \big\langle \nabla \mathcal{L}\big(\widehat{\mathbf{W}}_t\big) - \nabla \mathcal{L}\big(\mathbf{W}^*\big), \widehat{\mathbf{W}}_t - \mathbf{W}^* \big\rangle.$$

since $\nabla\mathcal{L}(W) = \mathbf{\Sigma}\mathbf{W} - U$. We note that, for any matrix $\mathbf{A}$, we have $\|\mathbf{A}\|_{\mathrm{F}} = \|\mathbf{A}\|_{2,2}$, that is the Frobenius norm and the $\ell_{2,2}$-norm coincides.

Let $\widehat{\mathbf{\Gamma}}$ be defined as above as a sub-gradient matrix of $\|\mathbf{W}\|_{1,1}$ evaluated at $\widehat{\mathbf{W}}$. To bound the right hand side of the inequality above, we add $\langle \lambda\widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle$ to both sides and obtain

$$\kappa_* \|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{2,2}^2 + \langle \nabla\mathcal{L}(\mathbf{W}^*) + \lambda\widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle \leq \langle \nabla\mathcal{L}(\widehat{\mathbf{W}}) + \lambda\widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle. \tag{3.9}$$

Since we have

$$\langle \nabla\mathcal{L}(\widehat{\mathbf{W}}) + \lambda\widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle = 0,$$

plugging the above equality back into (3.9), we obtain

$$\kappa_* \|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{1,2}^2 \leq \underbrace{-\langle \nabla\mathcal{L}(\mathbf{W}^*), \widehat{\mathbf{W}} - \mathbf{W}^* \rangle}_{\mathrm{II}_1} + \underbrace{-\langle \lambda\widehat{\mathbf{\Gamma}}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle}_{\mathrm{II}_2}. \tag{3.10}$$

In a similar argument to the proof of Lemma 7 above, applying Hölder inequality to both II_1 and II_2, we obtain

$$\begin{aligned}
-\langle \nabla\mathcal{L}(\mathbf{W}^*), \widehat{\mathbf{W}} - \mathbf{W}^* \rangle &\leq \|\nabla\mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{1,1} \\
&= \|\nabla\mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \left(\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}} + \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}^c} \right) \\
&\leq 4\|\nabla\mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1},
\end{aligned}$$

and

$$-\langle \lambda\mathbf{\Gamma}, \widehat{\mathbf{W}} - \mathbf{W}^* \rangle \leq -\lambda \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}^c}\|_{1,1} + \lambda \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1},$$

where we use Lemma 7 in the first displayed inequality. Plugging the bounds above for II_1 and II_2 back into (3.10) and then applying the Cauchy-Schwartz inequality to the term $\|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1} = \langle \mathbf{1}_{\mathcal{S}}, |(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}| \rangle$, we further obtain

$$\begin{aligned}
\kappa_* \|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{\mathrm{F}}^2 &\leq \left(4\|\nabla\mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} + \lambda\right) \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1} - \lambda \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}^c}\|_{1,1} \\
&\leq \left(4\lambda/2 + \lambda\right) \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1} \\
&\leq 3\lambda \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{1,1} \\
&\leq 3\lambda\sqrt{s} \|(\widehat{\mathbf{W}} - \mathbf{W}^*)_{\mathcal{S}}\|_{\mathrm{F}} \\
&\leq 3\lambda\sqrt{s} \|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{\mathrm{F}}
\end{aligned}$$

Canceling the term $\left\|\widehat{\mathbf{W}}-\mathbf{W}^*\right\|_{\mathrm{F}}$ on both sides, we obtain

$$\left\|\widehat{\mathbf{W}}-\mathbf{W}^*\right\|_{\mathrm{F}} \leq 3\kappa_*^{-1}\lambda\sqrt{s}. \tag{3.11}$$

We need the following lemma.

Lemma 8. *Let $\rho_{K,\mathbf{W}^*}$ be the K-th largest singular value of $\mathbf{W}^*$. Then we must have*

$$\left\|\mathbf{P}_{\widehat{\mathbf{W}}}-\mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} \leq \sqrt{2}\rho_{K,\mathbf{W}^*}^{-1}\left\|\widehat{\mathbf{W}}-\mathbf{W}^*\right\|_{\mathrm{F}}.$$

3.1.3.2 Proof of Lemma 8

Proof. Let $\mathbf{W}^*$ have the singular value decomposition that $\mathbf{W}^* = \mathbf{E}\mathbf{D}\mathbf{F}^\top$, $\mathbf{F} \in \mathbb{R}^{K\times K}$, $\mathbf{E} \in \mathbb{R}^{p\times K}$ are orthogonal, and $\mathbf{D} \in \mathbb{R}^{K\times K}$ is a diagonal matrix. Then we have $\mathbf{P}_{\mathbf{W}^*} = \mathbf{P}_{\mathbf{E}}$. Looking at the regression problem $\inf_{\mathbf{Q}}\left\|\mathbf{E}-\widehat{\mathbf{W}}\mathbf{Q}\right\|_{\mathrm{F}}^2$, the least squares solution is
$\mathbf{Q} = (\widehat{\mathbf{W}}^\top\widehat{\mathbf{W}})^{-1}\widehat{\mathbf{W}}^\top\mathbf{E}$, giving

$$\begin{aligned}
\inf_{\mathbf{Q}}\left\|\mathbf{E}-\widehat{\mathbf{W}}\mathbf{Q}\right\|_{\mathrm{F}}^2 &= \left\|\mathbf{E}-\mathbf{P}_{\widehat{\mathbf{W}}}\mathbf{E}\right\|_{\mathrm{F}}^2 \\
&= \mathrm{tr}\left(\mathbf{E}\mathbf{E}^\top - \mathbf{P}_{\widehat{\mathbf{W}}}\mathbf{E}\mathbf{E}^\top - \mathbf{E}\mathbf{E}^\top\mathbf{P}_{\widehat{\mathbf{W}}} + \mathbf{P}_{\widehat{\mathbf{W}}}\mathbf{E}\mathbf{E}^\top\mathbf{P}_{\widehat{\mathbf{W}}}\right) \\
&= \mathrm{tr}\left[\left(\mathbf{I}-\mathbf{P}_{\widehat{\mathbf{W}}}\right)\mathbf{P}_{\mathbf{E}}\right]
\end{aligned}$$

using identities $\mathbf{P}_{\widehat{\mathbf{W}}} = \widehat{\mathbf{W}}(\widehat{\mathbf{W}}^\top\widehat{\mathbf{W}})^{-1}\widehat{\mathbf{W}}^\top$, $\mathbf{P}_{\widehat{\mathbf{W}}}^2 = \mathbf{P}_{\widehat{\mathbf{W}}}$, $\mathbf{E}\mathbf{E}^\top = \mathbf{P}_{\mathbf{E}}$, and the cylic property of trace.

$$\begin{aligned}
\mathrm{tr}\left[\left(\mathbf{I}-\mathbf{P}_{\widehat{\mathbf{W}}}\right)\mathbf{P}_{\mathbf{E}}\right] &= K - \mathrm{tr}\left(\mathbf{P}_{\widehat{\mathbf{W}}}\mathbf{P}_{\mathbf{E}}\right) \\
&\geq \frac{1}{2}\mathrm{tr}\left(\mathbf{P}_{\mathbf{E}}\right) - \mathrm{tr}\left(\mathbf{P}_{\widehat{\mathbf{W}}}\mathbf{P}_{\mathbf{E}}\right) + \frac{1}{2}\mathrm{tr}\left(\mathbf{P}_{\widehat{\mathbf{W}}}\right) \\
&= \frac{1}{2}\left\|\mathbf{P}_{\mathbf{E}}-\mathbf{P}_{\widehat{\mathbf{W}}}\right\|_{\mathrm{F}}^2 \\
&= \frac{1}{2}\left\|\mathbf{P}_{\mathbf{W}^*}-\mathbf{P}_{\widehat{\mathbf{W}}}\right\|_{\mathrm{F}}^2,
\end{aligned}$$

where the first equality and first inequality uses the fact that $\mathbf{P}_{\mathbf{E}}$ is rank K and $\mathbf{P}_{\widehat{\mathbf{W}}}$ is at most rank K. Therefore, taking $\mathbf{Q} = \mathbf{F}\mathbf{D}^{-1}$, we can bound $\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}}^2$ as

$$\begin{aligned}
\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} &\leq \sqrt{2}\inf_{\mathbf{Q}}\left\|\widehat{\mathbf{W}}\mathbf{Q} - \mathbf{E}\right\|_{\mathrm{F}} \\
&\leq \sqrt{2}\left\|\widehat{\mathbf{W}}\mathbf{F}\mathbf{D}^{-1} - \mathbf{E}\mathbf{D}\mathbf{F}^{\top}\mathbf{F}\mathbf{D}^{-1}\right\|_{\mathrm{F}} \\
&\leq \sqrt{2}\left\|\widehat{\mathbf{W}} - \mathbf{W}^*\right\|_{\mathrm{F}}\|\mathbf{D}^{-1}\|_2 \\
&\leq \sqrt{2}\rho_{K,\mathbf{W}^*}^{-1}\left\|\widehat{\mathbf{W}} - \mathbf{W}^*\right\|_{\mathrm{F}}^2.
\end{aligned}$$

□

Now using Lemma 8, we obtain that

$$\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} \leq 3\sqrt{2}\rho_{K,\mathbf{W}^*}^{-1}\kappa_*^{-1}\lambda\sqrt{s}.$$

Then it remains to lower bound $\rho_K(\mathbf{W}^*)$. We start by writing $\mathbf{W}^*(\mathbf{W}^*)^{\top} = \mathbf{\Sigma}^{-1}\mathbf{\Omega}\mathbf{\Sigma}^{-1}$. We need the following lemma, which can be proved by the min-max theorem and thus the proof is omitted.

Lemma 9. *Let $\mathbf{A} \in \mathbb{R}^{d\times d}$ be a symmetric positive definite matrix and $\mathbf{B} \in \mathbb{R}^{d\times d}$ a symmetric positive semidefinite matrix. Then for any $1 \leq k \leq d$, we have*

$$\rho_{\min}(\mathbf{A})\rho_k(\mathbf{B}) \leq \rho_k(\mathbf{A}\mathbf{B}) \leq \rho_{\max}(\mathbf{A})\rho_k(\mathbf{B}).$$

Applying Lemma 9, we obtain that

$$\rho_{K,\mathbf{W}^*}^2 \geq \rho_K(\mathbf{\Sigma}^{-1/2}\mathbf{\Omega}\mathbf{\Sigma}^{-1/2})\rho_{\max}^{-1}(\mathbf{\Sigma}) \geq \rho_K(\mathbf{\Omega})\rho_{\max}^{-2}(\mathbf{\Sigma}).$$

Therefore, we complete the proof of desired statement by plugging the bound above. □

3.2 Sliced Inverse Regression

Supervised dimension reduction that preserves the conditional dependence of the data has a history in Sufficient Dimension Reduction (SDR), [Coo98]. As a method for performing SDR, the SIR method first developed in [Li91]. For a random vector $\boldsymbol{x}$ with elliptic distribution and univariate

response variable $y = f(x)$, the goal of finding a low-dimensional representation of $\boldsymbol{x}$ should take into account the relationship of the data to y, ideally without losing any information which is essential in predicting y. The objective of SDR methods is to find, without knowing f, a subspace $\mathcal{V} \subseteq \mathbb{R}^d$ such that $y \perp \boldsymbol{x} | \mathbf{P}_{\mathcal{V}} \boldsymbol{x}$. A subspace that satisfies this property is called an effective dimension reduction (EDR) space. Under some minor conditions, the intersection of all EDR spaces is also an EDR space with minimum dimension, called the central space. The minimal model for SDR methods is to assume the multiple index model where the link function takes the form

$$y = f(\mathbf{v}_1^\top \boldsymbol{x}, \ldots, \mathbf{v}_K^\top \boldsymbol{x}, \boldsymbol{\varepsilon})$$

for $\mathbf{v}_i \in \mathbb{R}^d$ for $i = 1, \ldots, K$ and the error $\boldsymbol{\varepsilon}$ is independent of $\boldsymbol{x}$ and $\mathbb{E}[\boldsymbol{\varepsilon}] = 0$. Thus, under this model it suffices to find $\text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$ to determine the central space.

The SIR estimator was one of many techniques developed for SDR, but was favored due to its simplicity and computational efficiency. The name Sliced Inverse Regression comes from both the use of the "inverse regression curve" $\mathbb{E}[\boldsymbol{x}|y]$ as well as the sliced estimator of $\boldsymbol{\Omega} = \text{cov}(\mathbb{E}[\boldsymbol{x}|y])$ to determine the central space. As proved in [Li91], since the column space $\text{col}(\boldsymbol{\Omega}) = \boldsymbol{\Sigma}\text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$, the central space can be estimated via $\hat{\boldsymbol{\Sigma}}^{-1}\text{col}(\hat{\boldsymbol{\Omega}})$. The additional linearity condition that $\forall b, \mathbf{v}_i \in \mathbb{R}^d, \mathbb{E}[b^\top \boldsymbol{x} | \mathbf{v}_1^\top \boldsymbol{x}, \ldots, \mathbf{v}_K^\top \boldsymbol{x}] = c_0 + \sum_{i=1}^K c_i \mathbf{v}_k^\top \boldsymbol{x}$, where $c_0, \ldots, c_K \in \mathbb{R}$, is required, but this is automatically satisfied assuming $\boldsymbol{x}$ is elliptically distributed. [**?**] proves the consistency of SIR holds if and only if $\lim d/n = 0$, motivating high dimension methods.

For the application of SDR, numerous competing procedures have been developed in the past couple decades, including [CL98, LN06, Li07]. While many approaches built flexible semiparametric models such as projection pursuit regression [FS81], and MAVE [XTLZ02], none of these function in HDLSS scenarios. A major breakthrough was achieved in regularlized SDR methods proposed by [LZL19] using the Lasso SIR method for the HDLSS under the sparsity assumption. The GEV method is closely related to the Lasso SIR method, but has important differences from their own that leads to performance improvements in scenarios that have significant eigengaps, and has consistent performance that is as good or better elsewhere. A closely related method to the GEV estimator can be found in [WCZZ18] which stems from similar motivations

but attempts only to compute the optimal projected coordinates of the data instead of determining the projection explicitly. Their regularization uses a group lasso approach in contrast to the GEV estimator, and the non asymptotic rates of error they demonstrated are suboptimal.

3.2.1 Consistency for SIR

The space $\Sigma^{-1}\text{col}(\Omega)$ is given by the span of the generalized eigenvectors of (Ω, Σ), justifying the GEV estimator. As in the original SIR estimation technique, we use the sliced estimator of Ω, defined as follows. Given the samples $(\boldsymbol{x}_i, y_i)$, $i \in [n]$, for a chosen constant $H \in \mathbb{N}, K < H < n$, divide the data into H groups determined by the order statistics of $y_{(i)}$ that give H "slices" of the domain of y in the form of intervals $(y_{(h_i)}, y_{(h_{i+1})}]$, $i \in \{2, \ldots, H-1\}$, with $(-\infty, y_{(h_1)}]$ and $(y_{(h_{H-1})}, \infty)$ at the tails. In general these may lead to different sized groups, but without loss of generality we may assume $n = cH$ so that each slice is chosen to consist of c points. We may relabel the data as $(\boldsymbol{x}_{h,j}, y_{h,j})$ for the j-th sample in slice number h, i.e., $y_{h,j} = y_{(c(h-1)+j)}$ and $\boldsymbol{x}_{h,j} = \boldsymbol{x}_{(c(h-1)+j)}$. Then the sample mean in h-th slice is $\bar{\boldsymbol{x}}_h \equiv \frac{1}{c}\sum_{j=1}^{c} \boldsymbol{x}_{h,j}$, allowing us to estimate $\widehat{\Omega}_H = \frac{1}{H}\sum_{h=1}^{H} \bar{\boldsymbol{x}}_h \bar{\boldsymbol{x}}_h^\top = \frac{1}{H}\mathbf{X}_H \mathbf{X}_H^\top$, where $\mathbf{X}_H = (\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_H)$ is the $d \times H$ matrix with the slice means as columns.

Here, the intuition for the estimate $\Omega = \text{cov}(\mathbb{E}[\boldsymbol{x}|y])$ is that each sample slice mean $\bar{\boldsymbol{x}}_h$ serves as a local estimate of $\mathbb{E}[\boldsymbol{x}|y \in I_h]$ for $I_h = (y_{h-1,c}, y_{h,c}]$. First, it is worth noting that it is not immediate that the samples $\bigcup_{j\in[c]} \boldsymbol{x}_{h,j}$ can be treated as coming from the distribution $\boldsymbol{x}|\,(y \in I_h)$; this is proven to be the case in [LZL18]. Furthermore, given the estimate $\frac{1}{H}\sum_H \mathbb{E}[\boldsymbol{x}|y \in I_h]\mathbb{E}[\boldsymbol{x}|y \in I_h]^\top$ of Ω, it is not guaranteed to be consistent. We can compare this estimate with the classic consistent estimator $\frac{1}{n}\sum_i \mathbb{E}[\boldsymbol{x}|y_i]\mathbb{E}[\boldsymbol{x}|y_i]^\top$ of Ω. A necessary condition for the sliced based estimate to be consistent is the average loss of variance in each slice decreases to zero as H increases. This holds automatically if $\mathbb{E}[\boldsymbol{x}|y]$ is smooth and y is compactly supported. In general though, one needs a basic assumption on the smoothness and tail distribution of $\mathbb{E}[\boldsymbol{x}|y]$. We give that assumption below, ϑ-stability ([LZL19]), which is minimal and holds for many distributions. Note that while an estimate based on samples of $\mathbb{E}[\boldsymbol{x}|y_i]$ is possible, the estimate would fair very poorly, since each mean $\mathbb{E}[\boldsymbol{x}|y_i]$

would have at best a single point estimate given by $\boldsymbol{x}_i$ from the pair $(\boldsymbol{x}_i, y_i)$.

Given this structure of $\widehat{\boldsymbol{\Omega}}$ and the usual estimator for $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum \boldsymbol{x}_i \boldsymbol{x}_i^\top$, we show consistency of the GEV estimator given the assumption of the Stability condition. To do so, given a decomposition of $\|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty}$ into terms involving estimation error of $\widehat{\boldsymbol{\Sigma}}$ and $\widehat{\mathbf{U}}$, we will use standard techniques to bound the former, but the latter will require extra work. We will show that while $\|\widehat{\mathbf{U}} - \mathbf{U}\|_{\infty,\infty}$ has inherent difficulties in controlling an upper bound, if one instead uses $\|\widehat{\mathbf{U}} - \tilde{\mathbf{U}}\|_{\infty,\infty}$ there is a direct way to control the upper bound with high probability, where $\tilde{\mathbf{U}} = \mathbf{P}_{\mathbf{U}}(\widehat{\mathbf{U}})$ is the projection of the estimated eigenvectors on the span of $\mathbf{U}$. As we will see, this avoids any cumbersome eigengap assumptions, but requires $\text{rank}(\tilde{\mathbf{U}}) = \text{rank}(\mathbf{U})$. To make this substitution one needs to guarantee that an estimator that uses $\tilde{\mathbf{U}}$ instead of $\mathbf{U}$ will recover the desired subspace. This is proven showing lower bounds on the norms of projected eigenvectors hold with high probability given the assumption of the stability condition. It is important to note that our parameter $\mathbf{U} = \left(\sqrt{\rho_1}\boldsymbol{u}_1, \ldots, \sqrt{\rho_K}\boldsymbol{u}_d\right)$ with the additional coefficients of the square roots of estimated eigenvalues makes this easier to bound than if there was no coefficients as found in the [LZL19] model.

Assumption 10 (Stability)**.** *For positive constants $\alpha_1 < 1 < \alpha_2$ let $A_H(\alpha_1, \alpha_2)$ be the collection of all partitions $-\infty = a_0 < a_1 < \cdots < a_H = \infty$ of $\mathbb{R}$ satisfying*

$$\frac{\alpha_1}{H} \leq \mathbb{P}(a_i \leq y < a_{i+1}) \leq \frac{\alpha_2}{H}.$$

The inverse regression curve $\mathbb{E}[\boldsymbol{x}|y]$ is ϑ-sliced stable with respect to y for some $\vartheta > 0$ if $\exists \alpha_1, \alpha_2, \alpha_3$ such that for any $\mathbf{v} \in \mathbb{R}^d$ and partition $A_H(\alpha_1, \alpha_2)$

$$\frac{1}{H}\left|\sum_{h=0}^{H-1} \text{var}(\mathbf{v}^\top \mathbb{E}[\boldsymbol{x}|y] | a_h \leq y \leq a_{h+1})\right| \leq \frac{\alpha_3}{H^\vartheta}\text{var}(\mathbf{v}^\top \mathbb{E}[\boldsymbol{x}|y])$$

for sufficiently large H. The curve is stable if it is ϑ-sliced stable for some positive constant ϑ.

Theorem 11. *Assume that assumption (10) holds for $\mathbb{E}[\boldsymbol{x}|y]$, $\boldsymbol{x}$ is sub-Gaussian with variance proxy σ, $n = \rho_K d^\alpha$ for some $\alpha > 1/2$, and $\lambda = C\sqrt{\frac{\log(d)}{n}}$ for some constant C. Then, there exists constants C_1, C_2 such that with probability at least $1 - C_1 \exp(-C_2 \log(d))$*

$$\|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq C\sqrt{\frac{\log(d)}{n}}.$$

Corollary 12. *There exist constants C, C_1, C_2 such that*

$$\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} \leq CM\kappa_*^{-1}\rho^{-1}\sqrt{\frac{s\log(d)}{n}}$$

holds with probability at least $1 - C_1\exp(-C_2\log(d))$.

3.2.2 Proof of Theorem 11

Proof. We have

$$\left\|\nabla\mathcal{L}(\mathbf{W}^*)\right\|_{\infty,\infty} \leq \underbrace{\left\|\widehat{\mathbf{\Sigma}}\mathbf{W}^* - \mathbf{\Sigma}\mathbf{W}^*\right\|_{\infty,\infty}}_{\mathrm{I}_1} + \underbrace{\left\|\mathbf{\Sigma}\mathbf{W}^* - \widehat{\mathbf{U}}\right\|_{\infty,\infty}}_{\mathrm{I}_2}.$$

We first bound I_1. Since we have $\|\mathbf{AB}\|_{\infty,\infty} \leq \|\mathbf{A}\|_{\infty,\infty}\|\mathbf{B}\|_1$ where the latter is the operator norm, since

$$\begin{aligned}
\|\mathbf{AB}\|_{\infty,\infty} &= \max_{i,j}\left(\sum_k \mathbf{A}_{i,k}\mathbf{B}_{k,j}\right) \\
&\leq \max_{i,j}|\mathbf{A}_{i,j}|\max_{\ell}\sum_k|\mathbf{B}_{k,\ell}| \\
&= \|\mathbf{A}\|_{\infty,\infty}\|\mathbf{B}\|_1
\end{aligned}$$

Then $\mathrm{I}_1 \leq \|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{\infty,\infty}\|\mathbf{W}^*\|_1$, where the second factor may be treated as a constant. It suffices to bound the estimation error of $\mathbf{\Sigma}$.

Lemma 13. *Let $\widetilde{M} = \max_i\sqrt{\mathbf{\Sigma}_{ii}}$. Suppose that $n > 6\log d$. Then, for some universal constant C, with probability at least* $1 - \exp(-\log(d/4))$*, we must have*

$$\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{\infty,\infty} \leq C\sqrt{\frac{\widetilde{M}\log d}{n}}.$$

Proof of Lemma 13. Using a similar argument to the proof of Lemma 1 in [RWRY11], which is omitted for simplicity, we obtain, for all $t \in \left(0, C_1\sqrt{\widetilde{M}}\right)$,

$$\mathbb{P}\left(\left|\widehat{\mathbf{\Sigma}}_{ij} - \mathbf{\Sigma}_{ij}\right| \geq t\right) \leq 4\exp\left\{-\frac{nt^2}{C_2\widetilde{M}}\right\},$$

where $C_1 = 3200$ and $C_2 = 40$ are two universal constants. Therefore, taking $t = \sqrt{3C_2\widetilde{M}n^{-1}\log d}$ such that $t < C_1\sqrt{\widetilde{M}}$ and applying union bound over the d^2 entries gives

$$\mathbb{P}\left(\left\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\right\|_{\infty,\infty} \geq \sqrt{\frac{3C_2\widetilde{M}\log d}{n}}\right) \leq 4d^{-1}$$

Observing that $t = \sqrt{3C_2\widetilde{M}n^{-1}\log d} < C_1\sqrt{\widetilde{M}}$ implies $n > 6\log d$ and this finishes the proof. □

What remains is bounding I_2. Using identity $\mathbf{\Sigma}\mathbf{W}^* = \mathbf{U}$, we have $\mathrm{I}_2 = \|\mathbf{U} - \widehat{\mathbf{U}}\|_{\infty,\infty}$. Let $\hat{\mathbf{V}} \equiv [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_K]$ and $\hat{\mathbf{\Lambda}} = \mathrm{diag}(\hat{\rho}_1, \ldots, \hat{\rho}_K)$. Then let $\mathbf{A} = \frac{1}{\sqrt{H}}\mathbf{X}_H^\top\hat{\mathbf{V}}$ be a $H \times K$ matrix. Let $\mathbf{x} = \mathbf{z} + \mathbf{w}$ be the orthogonal decomposition with respect to $\mathrm{col}(\mathbf{\Omega})$ and its complement. Then we have the decomposition $\mathbf{X}_H = \mathbf{Z}_H + \mathbf{W}_H$. This leads to the identity

$$\begin{aligned}\widehat{\mathbf{U}} &= \left(\frac{1}{H}\mathbf{X}_H\mathbf{X}_H^\top\hat{\mathbf{V}}\right)\hat{\mathbf{\Lambda}}^{-1/2} \\ &= \left(\frac{1}{\sqrt{H}}\mathbf{Z}_H\mathbf{A} + \frac{1}{\sqrt{H}}\mathbf{W}_H\mathbf{A}\right)\hat{\mathbf{\Lambda}}^{-1/2}\end{aligned}$$

and

$$\widetilde{\mathbf{U}} = \mathbf{P}_{\mathbf{U}}(\widehat{\mathbf{U}}) = \frac{1}{\sqrt{H}}\mathbf{Z}_H\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}.$$

Then $\widehat{\mathbf{U}} - \widetilde{\mathbf{U}} = \frac{1}{\sqrt{H}}\mathbf{W}_H\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}$ and

$$\|\frac{1}{\sqrt{H}}\mathbf{W}_H\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \|\frac{1}{\sqrt{H}}\mathbf{W}_H\|_{\infty,\infty}\|\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}\|_{1,\infty}$$

Note that $\|\hat{\mathbf{\Lambda}}^{-1/2}\mathbf{A}\|_{1,\infty} \leq \sqrt{H}\|\hat{\mathbf{\Lambda}}^{-1/2}\mathbf{A}\|_{2,\infty}$ by the basic inequality $\|\mathbf{v}\|_1 \leq \sqrt{H}\|\mathbf{v}\|_2$ for any $\mathbf{v} \in \mathbb{R}^H$. Then $\forall i \in [d]$, we have $\|\mathbf{A}_{*,i}\|_2 = \sqrt{\frac{1}{H}\hat{\mathbf{V}}_{*,i}^\top\mathbf{X}_H\mathbf{X}_H^\top\hat{\mathbf{V}}_{*,i}} = \sqrt{\hat{\rho}_i}$, thus

$$\|\frac{1}{\sqrt{H}}\mathbf{W}_H\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \|\mathbf{W}_H\|_{\infty,\infty}. \tag{3.12}$$

The benefit of having $\mathbf{U}$ in our model appears – if we had taken $\hat{\mathbf{V}}$ as our parameter and not $\widehat{\mathbf{U}}$, we would have had the above bound times $\max_i\sqrt{\hat{\rho}_i}$. Instead we merely need to bound the behavior of $\mathbf{W}_H$ to give an upper bound on I_2 given the legitimacy of the substitution of $\widetilde{\mathbf{U}}$ for $\mathbf{U}$, which we will see is very manageable.

To show the substitution of $\tilde{\mathbf{U}}$ is legitimate, define $\mathbf{W}^{**}$ as

$$\mathbf{W}^{**} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{\Sigma}^{1/2}\mathbf{W} - \boldsymbol{\Sigma}^{-1/2}\widetilde{\mathbf{U}}\|_{\mathrm{F}}^2 \right\} \tag{3.13}$$

We show with high probability that this model recovers the desired reduction space. It is trivial to show that assuming $\widetilde{\mathbf{U}}$ is of rank K, $\operatorname{span}(\mathbf{W}^*) = \operatorname{span}(\mathbf{W}^{**})$. To show that the rank is K, it suffices to show that none of the projected vectors of $\tilde{\boldsymbol{u}}_i = \mathbf{P}_{\mathbf{U}}(\hat{\boldsymbol{u}}_i)$ are 0, and that the projection of the K vectors are injective. Thus we give a positive lower bound on the norms $\|\tilde{\boldsymbol{u}}\|_2$, and lower bounds on the angles between $\angle(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_j)\ \forall i, j \in [K], i \neq j$.

These important results come directly from [LZL19].

Theorem 14. *If $n\rho_K = d^\alpha$ for some $\alpha > 1/2$, there exists positive constants C_1, C_2, C_3, C_4, such that*

1. *for $j = 1, \ldots, K$*
$$\|\tilde{\boldsymbol{u}}_j\|_2 \geq C_1 \sqrt{\frac{\rho_K}{\hat{\rho}_j}}$$
2. *for $j = K+1, \ldots, H$*
$$\|\tilde{\boldsymbol{u}}_j\|_2 \leq C_2 \frac{\sqrt{d\log(d)}}{n\rho_K} \sqrt{\frac{\rho_K}{\hat{\rho}_j}}$$

hold with probability at least $1 - C_3 \exp(-C_4 \log(d))$.

Its the first inequality that matters for our purposes; it not only gives us the lower bound on projected norms, it is also used in the next theorem:

Theorem 15. *The angles between any two vectors in $\{\tilde{\boldsymbol{u}}_1, \ldots, \tilde{\boldsymbol{u}}_d\}$ are nearly $\pi/2$ with high probability. In particular there exist constants C_1, C_2, C_3 such that*

$$|\cos(\angle(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_j))| \leq C_1 \frac{\sqrt{d\log(d)}}{n\rho_K}$$

holds with probability at least $1 - C_2 \exp(-C_3 \log(d))$. for any $i \neq j$.

Both proofs above are done in the case that $\boldsymbol{x}$ is Gaussian. The proofs rely on two core lemmas to show their claims hold with high probability. The first was proven in [**?**] and is already done in

the case that $\boldsymbol{x}$ is sub-Gaussian. The other lemma gives basic tail bounds for χ^2 variables and can be extended to the case of sub-exponential random variables.

Lemma 16. *Let $c_1,\dots,c_p$ be positive constants. We have the following statements:*

1. *For d sub-Gaussian random variables $x_1,\dots,x_d$, there exist constants C_1 and C_2 such that for any sufficiently small a we have*

$$\mathbb{P}\left(\left|\frac{1}{d}\sum_i c_i(x_i^2-\mathbb{E}[x_i^2])\right|>a\right)\leq C_1\exp\left(-\frac{d^2a^2}{C_2\sum_j c_j^2}\right). \tag{3.14}$$

2. *For $2d$ sub-Gaussian random variables $x_1,\dots,x_p,y_1,\dots,y_p$ with $\mathbb{E}[x_i]=\mathbb{E}[y_i]=0$ for all $i\in[d]$, there exist constants C_1 and C_2 such that for any sufficiently small a, we have*

$$\mathbb{P}\left(\left|\frac{1}{d}\sum_i c_i x_i y_i\right|>a\right)\leq C_1\exp\left(-\frac{d^2a^2}{C_2\sum_j c_j^2}\right). \tag{3.15}$$

Proof. If x_i is sub-Gaussian with parameter σ, then trivially x_i^2 is sub-exponential since $\forall t>0$ we have $\mathbb{P}\left(x_i^2\geq t\right)=\mathbb{P}\left(|x_i|\geq\sqrt{t}\right)\leq 2\exp(\frac{-t}{2\sigma^2})$ where the inequality comes from the sub-Gaussian property. It is straightforward to show that x_i^2 has sub-exponential parameters (ν_i,σ) for any $\nu_i^2>\mathbb{E}[x_i^4]$. As well for any $c_i\in\mathbb{R}$, $c_ix_i^2$ is sub-exponential with parameters $(c_i\nu_i,c_i\sigma)$ since we have the tail bound

$$\mathbb{P}(c_ix_i^2\geq t)=\mathbb{P}(x_i^2\geq t/c_i)\leq\exp\left(\frac{-t^2}{2c_i^2\nu_i^2}\right)$$

for all $0\leq t/c_i\leq\nu_i^2/\sigma\Leftrightarrow 0\leq t\leq\frac{c_i^2\nu_i^2}{c_i\sigma}$. Then $\frac{1}{d}\sum_i c_ix_i^2$ is sub-exponential with parameters $(\frac{\sqrt{\sum_i c_i^2\nu_i^2}}{d},\frac{\sigma(\max_i c_i)}{d})$. Then for $a\leq\frac{\sum_i c_i^2\nu_i^2}{d\sigma\max_i c_i}$ we have

$$\mathbb{P}\left(\left|\frac{1}{d}\sum_i c_i(x_i^2-\mathbb{E}[x_i^2])\right|>a\right)\leq 2\exp(-\frac{c^2a^2}{\sum_j c_j^2\nu_i^2}). \tag{3.16}$$

The second statement follows likewise since $x_iy_j=\frac{(x_i+y_j)^2-x_i^2-y_j^2}{2}$ is sub-exponential. □

Corollary 17. *Let $\Sigma_1\equiv\mathrm{cov}(\mathbf{w})$ and $\mathbf{I}_H$ the identity on $\mathbb{R}^H$. Then if $\frac{\sqrt{d\log(d)}}{n}$ is sufficiently small, the event*

$$\Omega=\left\{\omega\left|\|\frac{1}{H}\mathbf{W}_H^\top\mathbf{W}_H-\frac{\mathrm{tr}(\Sigma_1)}{n}\mathbf{I}_H\|_F\right.\right\}\leq a\frac{\sqrt{d\log(d)}}{n}$$

happens with probability at least

$$1 - C_1 \exp\left(-C_2 \log(d)\right).$$

Proof. Since $\boldsymbol{x}$ is sub-Gaussian, $\mathbf{X}_H$ has columns that are averages of c independent sub-Gaussians, and under linear projection each entry of $\mathbf{W}_H$ is likewise sub-Gaussian. The term inside the Frobenius norm is a matrix with entries that are a sum of sub-exponential random variables, with the subtraction of the means for the squared terms. In particular if W_{ij} is the ij-th entry of $\mathbf{W}_H$ and δ_{ij} is the usual Kronecker delta,

$$\frac{1}{H}\mathbf{W}_H^\top \mathbf{W}_H - \frac{\operatorname{tr}(\boldsymbol{\Sigma}_1)}{n}\mathbf{I}_H = \left(\sum_k^d \frac{W_{ki}W_{kj}}{H} - \delta_{ij}\frac{\operatorname{tr}(\boldsymbol{\Sigma}_1)}{n}\right)_{ij}.$$

Since every $W_{ki} = \frac{1}{c}\sum_{e=1}^c w_e^{ki}$ for $c = n/H$,

$$\sum_k^d \frac{W_{ki}W_{kj}}{H} = \frac{1}{c^2 n}\sum_k^d \left[\sum_{e=1}^c w_e^{ki}\right]\left[\sum_{e=1}^c w_e^{kf}\right].$$

When $i = j$ we have $\mathbb{E}\left[\sum_k^p \frac{W_{ki}^2}{H}\right] = \frac{\operatorname{tr}(\boldsymbol{\Sigma}_1)}{n}$. Then each $W_{ki}W_{kj}$ is sub-exponential and

$$\begin{aligned}
&\mathbb{P}\left(\|\frac{1}{H}\mathbf{W}_H^\top\mathbf{W}_H - \frac{\operatorname{tr}(\boldsymbol{\Sigma}_1)}{n}\mathbf{I}_H\|_F \le a\frac{\sqrt{d\log(d)}}{n}\right)\\
=&\mathbb{P}\left(\|\frac{1}{c^2}\mathbf{W}_H^\top\mathbf{W}_H - \operatorname{tr}(\boldsymbol{\Sigma}_1)\mathbf{I}_H\|_F \le a\sqrt{d\log(d)}\right)\\
=&\mathbb{P}\left(\frac{1}{c^2}\sum_{ij}^{i=H,j=H}\left[\sum_k^d W_{ki}W_{kj} - \delta_{ij}\operatorname{tr}(\boldsymbol{\Sigma}_1)\right]^2 \le a^2 d\log(d)\right)\\
\le& H^2\mathbb{P}\left(\frac{1}{c^2}\sum_k^d W_{ki}W_{kj} - \delta_{ij}\operatorname{tr}(\boldsymbol{\Sigma}_1) \le \frac{a\sqrt{d\log(d)}}{H}\right)\\
\le& C_1\exp\left(-\frac{\log(d)a^2}{C_2}\right)
\end{aligned}$$

where the last inequality comes from the application of (16) to $\frac{1}{c^2}\sum_k^d W_{ki}W_{kj} - \delta_{ij}\operatorname{tr}(\boldsymbol{\Sigma}_1)$ with first sub-exponential parameter being $O(d)$. □

Given the above, the proof of Theorem 14 goes through exactly as in [LZL19]. The proof of Theorem 15 uses a transformation of $\mathbf{X}_H$ via an orthogonal matrix T such that $\frac{1}{\sqrt{H}}T\mathbf{Z}_H = (\mathbf{A}^\top, 0)^\top$

and $\frac{1}{\sqrt{H}}T\mathbf{W}_H = (0, \mathbf{B}^\top)^\top$, where $\mathbf{A} \in \mathbb{R}^{K\times H}$ and $\mathbf{B} \in \mathbb{R}^{(d-K)\times H}$. Then via this transformation the proof depends on the following events happening with high probability: (i) $\rho_{\min}(\mathbf{A}\mathbf{A}^\top) \geq \lambda$, (ii) $\|\mathbf{P}_{T\mathbf{Z}_H}(T\hat{\boldsymbol{u}}_j)\|_2 \geq C\sqrt{\frac{\rho_K}{\hat{\rho}_j}}$ for all $j \in [K]$, and (iii) $\|\mathbf{B}^\top\mathbf{B} - \lambda\mathbf{I}_H\|_F \leq C\frac{\sqrt{d\log(d)}}{n}$ for some scalar $\lambda > 0$. (i) follows from the Sine-Theta Theorem, (ii) follows from Theorem 14, and (iii) follows from Lemma 16.

Lemma 18 (Bounding $\mathbf{W}_H$)**.** *Assume $\lambda = C\sqrt{\frac{\log(d)}{n}}$ for some constant C. Then there exist constants C_1, C_2 such that*

$$\|\frac{1}{\sqrt{H}}\mathbf{W}_H\mathbf{A}\hat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \lambda/2$$

with probability at least $1 - C_1\exp(-C_2\log(d))$.

Proof. Using the bound in (3.12), it suffices to bound $\|\mathbf{W}_H\|_{\infty,\infty}$. As a linear function of a sub-Gaussian variable, each entry $W_{ij} = \frac{1}{c}\sum_{k=1}^{c} w_{ij}^k$ of $\mathbf{W}_H$ is sub-Gaussian with parameter $\sigma_w/\sqrt{c}$, for some $\sigma_w \in \mathbb{R}$, $i \in [d]$ and $j \in [H]$. Then using a union bound we have

$$\begin{aligned}\mathbb{P}\left(\max_{i,j}|W_{ij}| \geq \sqrt{\frac{CH\log(dH)}{4n}}\right) &\leq dH\mathbb{P}\left(|\sqrt{c}W_{ij}| \geq \sqrt{C\log(dH)}\right) \\ &\leq 2e^{-(C-1)\log(dH)}\end{aligned}$$

□

Combining the bounds for I_1 and I_2 we may seek bounds

$$\mathbb{P}\left(\|\nabla\mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq \frac{\lambda}{2}\right) \leq \mathbb{P}\left(\left\|\widehat{\mathbf{\Sigma}}\mathbf{W}^* - \mathbf{\Sigma}\mathbf{W}^*\right\|_{\infty,\infty} \leq \frac{\lambda}{4}\right) + \mathbb{P}\left(\left\|\mathbf{\Sigma}\mathbf{W}^* - \widehat{\mathbf{U}}\right\|_{\infty,\infty} \leq \frac{\lambda}{4}\right)$$

Setting $\lambda = 2\|\mathbf{W}^*\|_1 C\sqrt{\frac{\log(d)}{n}}$ yields the desired probabilities, with a change of constants. This completes the proof of the bound. □

3.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a classification technique, that assumes $\boldsymbol{x} \in \mathbb{R}^d$ is a random vector and $y \in \{1, \ldots, K+1\}$ is a discrete random variable over $K+1$ classes, such that $\boldsymbol{x}|y = k \sim$

$N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ for $k \in \{1, \ldots, K+1\}$. Given the normal model, the Bayes rule for estimating y can be explicitely derived as

$$\hat{y} = \underset{k}{\operatorname{argmax}}\, \mathbf{v}_k^\top (\boldsymbol{x} - \frac{\boldsymbol{\mu}_k}{2}) + \log(\pi_k),$$

where $\pi_k = \mathbb{P}(y = k)$ and $\mathbf{v}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$ for $k \in [K+1]$. Alternatively, LDA can be viewed from the perspective of dimensionality reduction, given by Fisher's discriminant problem where one seeks a low dimensional projection of the data such that the between-class variance is large relative to the within class variance. This problem is formulated as

$$\max_{\mathbf{v}_k} \mathbf{v}_k^\top \boldsymbol{\Omega} \mathbf{v}_k \text{ subject to } \mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_k \leq 1,\ \mathbf{v}_k^\top \boldsymbol{\Sigma} \mathbf{v}_i = 0 \quad \forall i < k.$$

where $\boldsymbol{\Omega} \equiv \operatorname{cov}(\mathbb{E}[\boldsymbol{x}|y])$ as before.

It is simple to show that this gives us exactly the same problem that motivates the GEV estimator for the case of a discrete response variable y. Note here that $\boldsymbol{\Sigma}$ is the homoskedastic within-class covariance, and not $\operatorname{cov}(\boldsymbol{x})$. However, it is straightforward to show using the law of total covariance that the generalized eigenvectors of $(\boldsymbol{\Omega}, \boldsymbol{\Sigma})$ are the same as those of $(\boldsymbol{\Omega}, \operatorname{cov}(\boldsymbol{x}))$ (with different eigenvalues). Thus we can treat $\boldsymbol{\Sigma}$ above as $\operatorname{cov}(\boldsymbol{x})$ without affecting the problem. The vectors $\mathbf{v}_k$ determined by the optimization are called the discriminant vectors, and for a $d \times n$ data matrix $\mathbf{X}$ a classification rule is obtained by computing the projection $\mathbf{V}^\top \mathbf{X}$ for $\mathbf{V} \equiv [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, and then assigning each observation to the nearest $\mathbf{V}^\top \boldsymbol{\mu}_k$.

In recent years, several references have extended LDA to the high dimensional setting using sparsity, most of which use a lasso penalty [THNC03, Len08, CHWE11]. The derivation of LDA using Fisher's Discriminant is leveraged in the well-known method found in [WT11], that uses a group lasso regularization with a diagonal estimate for the covariance of the data. A powerful technique called MSDS for multiclass sparse discriminant analysis is given in [ZMY18], which has both theoretical and empirical justification. This estimator attempts to find sparse discriminant vectors from the Bayes rule of the normal model for LDA, whereas our own method uses the Fisher's Discriminant derivation. The authors also show the estimators of ROAD [FFT12], and DSDA [MZY12], occur as special cases of MSDS up to particular constants. While error bounds

are given on the estimator, they are not in the form of a function of the data dimension and sample size. We show our estimator performs better empirically and has strong error bounds in terms of the data dimension and sample size.

3.3.1 Consistency for LDA

Fisher's Discriminant problem in many ways can be seen as a special case of SIR for a discrete output of y. With $\mathbf{\Omega}$ as defined above and with the assumption that $\mathbb{E}[\boldsymbol{x}] = 0$, its estimator simplifies to $\hat{\mathbf{\Omega}} = \frac{1}{K+1}\sum_{k=1}^{K+1} \bar{\boldsymbol{x}}_k \bar{\boldsymbol{x}}_k^\top$ where $\bar{\boldsymbol{x}}_k = \frac{1}{n_k}\sum_{i=1}^{n} x_i \mathbf{1}(y_i = k)$. Here we use $\mathbf{1}(y_i = k)$ as the indicator variable that x_i is of class k, and $n_k = \sum_{i=1}^{n} \mathbf{1}(y_i = k)$ is the number of samples that are of class k. Let $\hat{\mathbf{U}} = (\sqrt{\hat{\rho}_1}\hat{\boldsymbol{u}}_1, \ldots, \sqrt{\hat{\rho}_K}\hat{\boldsymbol{u}}_K)$ be columns using the eigenpairs $(\hat{\rho}_k, \hat{\boldsymbol{u}}_k)$ of $\hat{\mathbf{\Omega}}$. Then the GEV solution $\hat{\mathbf{W}}$ of (3.5) using $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{U}}$ gives us K estimated discriminant vectors which can be used for LDA classification.

Due to the similarity of LDA to SIR, much of the proof of consistency of the discriminant vectors can proceed analogously to that of SIR. Here, the Stability assumption (10) takes the form of a "balanced clusters" assumption. Since the Stability assumption holds for any H sufficiently large, we may take $H = K+1$ so that for $\gamma_1 = K+1 \min_{k\in[K+1]} \pi_k$ and $\gamma_2 = K \max_{k\in[K]} \pi_k$, any partition $A_H(\gamma_1, \gamma_2)$ will separate the support values of y into intervals $[a_{k-1}, a_k]$, $k-1 < a_{k-1} < k < a_k < k+1$. Thus for $k \in [K+1]$, the random vector

$$\big(\mathbb{E}[\boldsymbol{x}|y]\big|a_{k-1} \le y \le a_k\big) = \mathbb{E}[x|y=k]$$

is a constant. Then $\mathrm{var}(\mathbf{v}^\top \mathbb{E}[\boldsymbol{x}|y]\big|a_{k-1} \le y \le a_k) = 0$ for all $\mathbf{v} \in \mathbb{R}^d$, $k \in [K+1]$. The existence of positive constants $\gamma_1 < 1 < \gamma_2$ that serve as bounds to the probabilities of the classes of $\boldsymbol{x}$ guarantee that there are no asymptotic behaviors of the form $\lim_{d\to\infty} \pi_k = 0$ as a function of dimensionality of $\boldsymbol{x}$. For the sake of simplicity, we assume all probabilities π_k are equal, with the extension to the general case being straightforward.

Theorem 19. *Let $\boldsymbol{x}$ be multivariate Gaussian, $n = \rho_K d^\alpha$ for some $\alpha > 1/2$, and $y = f(\boldsymbol{x})$ has finite support along with assumption (10) holding for $\mathbb{E}[\boldsymbol{x}|y]$. Then, with probability at least* $1-$

$C_1 \exp(-C_2 \log(d))$, *we have*

$$\|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq C\sqrt{\frac{\log(d)}{n}}.$$

Corollary 20. *Assume that* $1/M \leq \rho_{\min}(\mathbf{\Sigma}) \leq \rho_{\max}(\mathbf{\Sigma}) \leq M$ *and* $\rho \leq \rho_K(\mathbf{\Omega})$. *Then there exist constants* C, C_1, C_2 *such that*

$$\left\|\mathbf{P}_{\widehat{\mathbf{W}}} - \mathbf{P}_{\mathbf{W}^*}\right\|_{\mathrm{F}} \leq CM\kappa_*^{-1}\rho^{-1}\sqrt{\frac{s\log(d)}{n}}$$

holds with probability at least $1 - C_1\exp(-C_2\log(d))$.

3.3.1.1 Proof of Theorem 19

Proof. For the sake of simplicity, we assume all probabilities π_k are equal, with the extension to the general case being straightforward. As earlier, we have

$$\left\|\nabla \mathcal{L}(\mathbf{W}^*)\right\|_{\infty,\infty} \leq \underbrace{\left\|\widehat{\mathbf{\Sigma}}\mathbf{W}^* - \mathbf{\Sigma}\mathbf{W}^*\right\|_{\infty,\infty}}_{\mathrm{I}_1} + \underbrace{\left\|\mathbf{U} - \widehat{\mathbf{U}}\right\|_{\infty,\infty}}_{\mathrm{I}_2}.$$

The bound on I_1 follows from the theorem used earlier in [RWRY11]. What remains is bounding I_2.

Let $\widehat{\mathbf{V}} = [\widehat{\boldsymbol{u}}_1, \ldots, \widehat{\boldsymbol{u}}_K]$, $\widehat{\mathbf{\Lambda}} = \mathrm{diag}(\hat{\rho}_1, \ldots, \hat{\rho}_K)$, and $\mathbf{X}_{K+1} = (\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_{K+1})$. Then let $\mathbf{A} = \frac{1}{\sqrt{K+1}}\mathbf{X}_{K+1}^\top \widehat{\mathbf{V}}$ be a $(K+1)\times K$ matrix. Using the same decomposition $\mathbf{x} = \mathbf{z} + \mathbf{w}$ with respect to $\mathrm{col}(\mathbf{\Omega})$ and its complement we get $\mathbf{X}_{K+1} = \mathbf{Z}_{K+1} + \mathbf{W}_{K+1}$. Since $\frac{1}{K+1}\mathbf{X}_{K+1}\mathbf{X}_{K+1}^\top = \widehat{\mathbf{\Omega}}$, we get the identity

$$\widehat{\mathbf{U}} = \left(\frac{1}{\sqrt{K+1}}\mathbf{Z}_{K+1}\mathbf{A} + \frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\mathbf{A}\right)\widehat{\mathbf{\Lambda}}^{-1/2}$$

and

$$\widetilde{\mathbf{U}} = \mathbf{P}_{\mathbf{\Omega}}(\widehat{\mathbf{U}}) = \frac{1}{\sqrt{H}}\mathbf{Z}_{K+1}\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}.$$

Then $\widehat{\mathbf{U}} - \widetilde{\mathbf{U}} = \frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}$ and

$$\|\frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \|\frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\|_{\infty,\infty}\|\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}\|_{1,\infty}.$$

Since $\|\widehat{\mathbf{\Lambda}}^{-1/2}\mathbf{A}\|_{1,\infty} \leq \sqrt{K+1}\|\widehat{\mathbf{\Lambda}}^{-1/2}\mathbf{A}\|_{2,\infty}$, $\forall i \in [d]$, and

$$\|\mathbf{A}_{*,i}\|_2 = \sqrt{\frac{1}{K+1}\widehat{\mathbf{V}}_{*,i}^\top \mathbf{X}_{K+1}\mathbf{X}_{K+1}^\top \widehat{\mathbf{V}}_{*,i}} = \sqrt{\hat{\rho}_i},$$

we have

$$\|\frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \|\mathbf{W}_{K+1}\|_{\infty,\infty}.$$

As in the SIR application, we are able to use $\widetilde{\mathbf{U}}$ in a substitute model (3.13) that recovers the desired reduction space with high probability. All that needs to be shown is that with high probability $\widetilde{\mathbf{U}}$ is of rank K. We may directly use the results of (14) and (15), since all assumptions of SIR hold in the LDA application, with the only requirement being a similar simplifying assumption that $\exists c > 0$ such that $c(K+1) = n$. Thus we get that $\widetilde{\mathbf{U}}$ is of rank K with probability at least $1 - C_1\log(-C_2 d)$ for constants C_1, C_2.

Lemma 21 (Bounding $\mathbf{W}_{K+1}$)**.** *Assume $\lambda = C\sqrt{\frac{\log(d)}{n}}$ for some constant C. Then there exist constants C_1, C_2 such that*

$$\|\frac{1}{\sqrt{K+1}}\mathbf{W}_{K+1}\mathbf{A}\widehat{\mathbf{\Lambda}}^{-1/2}\|_{\infty,\infty} \leq \lambda/2$$

with probability at least $1 - C_1\exp(-C_2\log(d))$.

Proof. Without loss of generality, each entry $W_{ij} = \frac{1}{c}\sum_{k=1}^{c} w_{ij}^k$ of $\mathbf{W}_H$ is Gaussian with variance $\text{var}(\mathbf{w})/c^2$, assuming each w_{ij}^k is Gaussian with variance $\text{var}(\mathbf{w})$ for $i \in [d]$ and $j \in [K+1]$. Then using a union bound we have

$$\begin{aligned}
&\mathbb{P}\left(\max_{i,j}|W_{ij}| \geq \sqrt{\frac{C(K+1)\log(d(K+1))}{4n}}\right) \\
&\leq d(K+1)\mathbb{P}\left(|c^2 W_{ij}| \geq \sqrt{C\log(d(K+1))}\right) \\
&\leq 2e^{-(C-1)\log(d(K+1))}
\end{aligned}$$

□

This completes the proof. □

Given the consistency of $\widehat{\mathbf{W}}$, it is straightforward to show the consistency of the classification rate of the estimated classifier. The classifier $\widehat{Y}_{\widehat{\mathbf{W}}}$ gives the label k to a new point $\boldsymbol{x}'$ if $\boldsymbol{\mu}_k$ is the nearest class centroid to $\boldsymbol{x}'$ after projection by $\widehat{\mathbf{W}}$:

$$\widehat{Y}_{\widehat{\mathbf{W}}}(\mathbf{X}) \equiv \underset{k}{\operatorname{argmin}} \|\widehat{\mathbf{W}}^\top \boldsymbol{x}' - \widehat{\mathbf{W}}^\top \hat{\boldsymbol{\mu}}_k\|_2^2.$$

Let the misclassification error rate be given by $R_n \equiv \mathbb{P}(\widehat{Y} \neq Y | \text{observed data})$ where Y is the true label. Likewise define R as the misclassification rate of the population classifier using the parameters $\mathbf{W}^*$ and $\boldsymbol{\mu}_k$. Then we have the following.

Theorem 22. *There exist constants C, C_1, C_2 such that*

$$|R_n - R| \leq C\kappa_*^{-1}\rho^{-1}\sqrt{\frac{s\log(d)}{n}}$$

with probability at least $1 - C_1\exp(-C_2\log(d))$.

3.3.1.2 Proof of Theorem 22

Proof. Let Y be the classifier using the population parameters of $\mathbf{W}^*$ and $\boldsymbol{\mu}_k$. Define $l_k = \|\mathbf{W}^{*\top}\mathbf{X} - \mathbf{W}^{*\top}\boldsymbol{\mu}_k\|_2^2$ and $\hat{l}_k = \|\widehat{\mathbf{W}}^\top\mathbf{X} - \widehat{\mathbf{W}}^\top\hat{\boldsymbol{\mu}}_k\|_2^2$. For any $\varepsilon > 0$, we have

$$\begin{aligned} R_n - R &\leq \mathbb{P}(\widehat{Y} \neq Y) \\ &\leq 1 - \mathbb{P}(|\hat{l}_k - l_k| < \varepsilon/2,\ |l_k - l_{k'}| > \varepsilon, \text{ for any } k, k') \\ &\leq \mathbb{P}(|\hat{l}_k - l_k| \geq \varepsilon/2 \text{ for some } k) + \mathbb{P}(|l_k - l_{k'}| \leq \varepsilon \text{ for some } k, k'). \end{aligned}$$

For the first term

$$|\hat{l}_k - l_k| \leq |(\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top\hat{\boldsymbol{\mu}}_k - \mathbf{W}^*\mathbf{W}^{*\top}\mu_k)^\top\mathbf{X}| + |\hat{\boldsymbol{\mu}}_k^\top\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top\hat{\boldsymbol{\mu}}_k - \mu_k^\top\mathbf{W}^*\mathbf{W}^{*\top}\mu_k|.$$

It is straightforward to show given (3.11) that bounds $\|\widehat{\mathbf{W}} - \mathbf{W}^*\|_{\mathrm{F}} \leq 3\kappa_*^{-1}\lambda\sqrt{s}$, we have $\|\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top - \mathbf{W}^*\mathbf{W}^{*\top}\|_2 \leq C\kappa_*^{-1}\lambda\sqrt{s}$ and given a standard error bound on sample mean estimation we get $|\hat{l}_k - l_k| \leq C\kappa_*^{-1}\sqrt{\frac{s\log(d)}{n}}$ with probability $1 - C_1\exp(-C_2\log(d))$. The expression $|l_k - l_{k'}|$ is normally distributed with mean $(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^\top\mathbf{W}^*\mathbf{W}^{*\top}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_{k'})$ and variance $(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^\top\mathbf{W}^*\mathbf{W}^{*\top}\boldsymbol{\Sigma}\mathbf{W}^*\mathbf{W}^{*\top}(\boldsymbol{\mu}_k -$

$\boldsymbol{\mu}_{k'}$), and since we have the assumption (5) which implies $(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})$ is bounded away from 0, with high probability $|l_k - l_{k'}| > C\kappa_*^{-1}\sqrt{\frac{s\log(d)}{n}}$. Applying a union bound over k, k' for both terms yields the result. □

3.4 Minimax Rate

We show that both the error rates for SIR and LDA fall under a model which achieve minimax rates. The setup closely follows [TSY20]. Define the GEV parameter space as follows: let $\{J_1, \ldots, J_H\}$ for $H > K$ be a measurable partition of the sample space of y, and define $\tilde{y} = \sum_{c=1}^{H} c \cdot \mathbf{1}(y \in J_c)$ as the discretized version of y. Then we may define the corresponding conditional covariance $\widetilde{\boldsymbol{\Omega}} \equiv \mathrm{cov}[\mathbb{E}(\boldsymbol{x}|\tilde{y})]$. Let $\mathcal{F}(s, d, K, \tilde{\gamma}; \kappa, M)$ be the set of all pairs of matrices $(\boldsymbol{\Sigma}, \widetilde{\boldsymbol{\Omega}})$ with generalized eigenpairs $(\tilde{\gamma}_i, \tilde{\boldsymbol{v}}_i)$, $\tilde{\gamma}_i \in \mathbb{R}$, $\tilde{\boldsymbol{v}}_i \in \mathbb{R}^d$ for $i \in [K]$, such that

1. $\sum_{i=1}^{K} \|\tilde{\boldsymbol{v}}_i\|_0 = s$.
2. $1/M \leq \rho_{\min}(\boldsymbol{\Sigma}) \leq \rho_{\max}(\boldsymbol{\Sigma}) \leq M$.
3. $\kappa\tilde{\gamma} \geq \tilde{\gamma}_1 \geq \cdots \geq \tilde{\gamma}_K \geq \tilde{\gamma} > 0$ for a fixed constant $\kappa > 1$.

For simplicity denote $\mathcal{F}(s, d, K, \tilde{\gamma}; \kappa, M)$ by $\mathcal{F}$. Let $\mathcal{L}(\boldsymbol{x})$ denote the distribution of a random variable $\boldsymbol{x}$. Then the probability spaces for the GEV problem are

$$\mathcal{P}(n, H, s, d, K, \tilde{\gamma}; \kappa, M) = \{\mathcal{L}((\boldsymbol{x}_1, \tilde{y}_1), \ldots, (\boldsymbol{x}_n, \tilde{y}_n)) :$$

$$(\boldsymbol{x}_i, \tilde{y}_i)\text{'s are i.i.d. such that } (\mathrm{cov}[\mathbb{E}(\boldsymbol{x}_i|\tilde{y}_i)], \mathrm{cov}(\boldsymbol{x}_i)) \in \mathcal{F}\},$$

where n is the sample size, and parameters s, d, and $\tilde{\gamma}$ may depend on n, while κ, M are fixed constants. Note that this framework gives a fixed slicing scheme where H is treated as a bounded integer, so that K is also bounded above by $H - 1$. Denote $\widetilde{\mathbf{V}} \in \mathbb{R}^{d \times K}$ as the matrix where each column is one of the generalized eigenvectors of $(\boldsymbol{\Sigma}, \widetilde{\boldsymbol{\Omega}})$ with nonzero eigenvalue. Let $\widehat{\mathbf{V}}$ be any estimator for $\widetilde{\mathbf{V}}$. The following provides a lower bound on the minimax rate among all estimators.

Theorem 23. *Assume $n\tilde{\gamma}^2 \geq C \log \frac{ed}{s}$ for some sufficiently large constant C_0. Then there exist positive constants C_1 and C_2 such that*

$$\inf_{\hat{\mathbf{V}}} \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\left\{ \|\mathbf{P}_{\hat{\mathbf{V}}} - \mathbf{P}_{\tilde{\mathbf{V}}}\|_{\mathrm{F}}^2 \geq C_1 \frac{s\log(ed/s)}{n\tilde{\gamma}^2} \wedge C_2 \right\} \geq 0.8,$$

where $\mathcal{P} = \mathcal{P}(n, H, s, d, K, \tilde{\gamma}; \kappa, M)$.

3.4.1 Proof of Theorem 23

The proof follows from [TSY20], for completeness we include the outline of the proof and the relevant papers for intermediate steps. Since any special case of the estimation problem yields a lower bound for the general case, we are able to specify further that the data distribution has an assumption of normality on the conditional distribution $\boldsymbol{x}|\tilde{y}$.

We specify a subset of the parameter space as follows: let $K = 1$, $H = 2$, and for $i = 1, 2$, let

$$\boldsymbol{x}_i|(\tilde{y}_i = 1) \sim N_d((1-\alpha)\mathbf{v}, I_d - \tilde{\boldsymbol{\Omega}}), \quad \mathbb{P}(\tilde{y} = 1) = \alpha,$$

$$\boldsymbol{x}_i|(\tilde{y}_i = 2) \sim N_d(-\alpha\mathbf{v}, I_d - \tilde{\boldsymbol{\Omega}}), \quad \mathbb{P}(\tilde{y} = 2) = 1-\alpha.$$

Here we have $\mathbf{v} \in \mathbb{O}(d, 1)$, (unit length d-vectors), and for $0 < \alpha < 1$ and $\gamma = \alpha$ we have $\mathbb{E}[\boldsymbol{x}] = 0$, $\tilde{\boldsymbol{\Omega}} = \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|\tilde{y}]) = \gamma\mathbf{v}\mathbf{v}^\top$, and $\boldsymbol{\Sigma} = \mathrm{cov}(\boldsymbol{x}) = I_d$ for I_d the identity matrix on $\mathbb{R}^d$.

The minimax results are derived using Fano's Lemma found in [Yu97] (Lemma 3).

Lemma 24 (Fano's Lemma)**.** *Let (Θ, ρ) be a metric space and $\{\mathbb{P}_\theta : \theta \in \Theta\}$ a collection of probability measures. For any totally bounded $T \subset \Theta$, denote by $\mathcal{M}(T, \rho, \varepsilon)$ the ε-packing number of T with respect to ρ, that is, the maximimal number of points in T whose pairwise minimum distance in ρ is at least ε. Define the Kullback-Leibler diameter of T by*

$$d_{\mathrm{KL}}(T) \equiv \sup_{\theta,\theta'\in T} D(\mathbb{P}_\theta \| \mathbb{P}_{\theta'})$$

for $D(\mathbb{P}_\theta || \mathbb{P}_{\theta'})$ the KL divergence between distributions $\mathbb{P}_\theta$ and $\mathbb{P}_{\theta'}$. Then

$$\inf_{\hat{\theta}} \sup_{\theta\in\Theta} \mathbb{E}_\theta[\rho^2(\hat{\theta}(\boldsymbol{x}), \theta)] \geq \sup_{T\subset\Theta} \sup_{\varepsilon>0} \frac{\varepsilon^2}{4}\left(1 - \frac{d_{\mathrm{KL}}(T) + \log 2}{\log \mathcal{M}(T, \rho, \varepsilon)}\right)$$

and equivalently,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} \mathbb{P}_\theta \left(\rho^2(\hat{\theta}(\boldsymbol{x}), \theta) \geq \frac{\varepsilon^2}{4} \right) \geq 1 - \frac{d_{\mathrm{KL}}(T) + \log 2}{\log \mathcal{M}(T, \rho, \varepsilon)}.$$

Then two steps are required, first determining the Kullback-Leibler divergence between the data distributions in T, and second determining the ε-packing of T.

The first task is straightforward. For $i = 1, 2$, let $\boldsymbol{\Sigma} = I_d$ and $\widetilde{\boldsymbol{\Omega}}_i = \gamma \mathbf{v}_i \mathbf{v}_i^\top$, for $\gamma \in (0,1)$, $\mathbf{v}_i \in \mathbb{O}(d,1)$. Then let $\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma})$ denote the distribution of a random i.i.d. sample of size n from the mixture Guassian distribution $\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma}) = \alpha \mathbb{P}_1(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma}) + (1-\alpha)\mathbb{P}_2(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma})$, where $\mathbb{P}_1(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma})$ and $\mathbb{P}_2(\widetilde{\boldsymbol{\Omega}}_i, \boldsymbol{\Sigma})$ denote multivariate normal distributions $N_d((1-\alpha)\mathbf{v}_i, I_d - \widetilde{\boldsymbol{\Omega}}_i)$ and $N_d(-\alpha \mathbf{v}_i, I_d - \widetilde{\boldsymbol{\Omega}}_i)$, respectively. Then using convexity of the K-L divergence, for two mixture-Gaussian distributions $\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_1, \boldsymbol{\Sigma})$ and $\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_2, \boldsymbol{\Sigma})$ we have

$$D(\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_1, \boldsymbol{\Sigma}) || \mathbb{P}(\widetilde{\boldsymbol{\Omega}}_2, \boldsymbol{\Sigma})) \leq \alpha D(\mathbb{P}_1(\widetilde{\boldsymbol{\Omega}}_1 || \boldsymbol{\Sigma}), \mathbb{P}_1(\widetilde{\boldsymbol{\Omega}}_2, \boldsymbol{\Sigma})) + (1-\alpha) D(\mathbb{P}_2(\widetilde{\boldsymbol{\Omega}}_1, \boldsymbol{\Sigma}) || \mathbb{P}_2(\widetilde{\boldsymbol{\Omega}}_2, \boldsymbol{\Sigma})),$$

thus it is sufficient to bound the K-L divergence between two Gaussian distributions.

Using the explicit formula for the K-L divergence between Gaussian distributions and the properties of the chosen parameters, the authors of [TSY20] determine the upper bound

$$D(\mathbb{P}(\widetilde{\boldsymbol{\Omega}}_1, \boldsymbol{\Sigma}) || \mathbb{P}(\widetilde{\boldsymbol{\Omega}}_2, \boldsymbol{\Sigma})) \leq \frac{3\gamma^2}{1-\gamma^2} \cdot n \|\mathbf{v}_1 - \mathbf{v}_2\|_2^2.$$

Once the K-L divergence is determined, the second task proceeds according to a well established packing argument that can be found in [CGRZ13] or [CMW13].

Corollary 25. *Assume that ρ is a lower bound of both ρ_K and $M\gamma_K$, where γ_K is the K-th generalized eigenvalue of the pair $(\boldsymbol{\Omega}, \boldsymbol{\Sigma})$. Then GEV estimator obtains the minimax rate up to constants.*

3.4.2 Proof of Corollary 25

Proof. The assumption that $\sum_{i=1}^K \|\tilde{\boldsymbol{v}}_i\|_0 = s$ is equivalent to the sparsity assumptions on $\mathbf{W}^*$. We have $\exists \gamma > 0$ such that $\gamma < \gamma_K$ for generalized eigenpair $(\mathbf{v}_K, \gamma_K)$ of $(\boldsymbol{\Omega}, \boldsymbol{\Sigma})$. It is simple to show that with the assumption $\rho < M\gamma_K$, then we may substitute γ with ρ/M for a lower bound on γ_K.

One needs to compare γ with $\tilde{\gamma}$. We can do this with the stability theorem and Weyl's theorem. Let $\mathbb{P}(y \in J_h) = \mathbb{P}_h$ and $\mu_h = \mathbb{E}[\boldsymbol{x}|y \in J_h]$. We need the following from [**?**] (Lemma 11 in supplementary materials).

Lemma 26. *Define the event* $E(\varepsilon) = \{\omega \big| |\mathbb{P}_h - \frac{1}{H}| > \varepsilon, \forall H\}$. *There exist a positive constant C such that, for any* $\varepsilon > \frac{4}{Hc-1}$, *we have*

$$\mathbb{P}(E(\varepsilon)) < CH^2\sqrt{Hc+1}\exp(-(Hc+1)\frac{\varepsilon^2}{32})$$

for sufficiently large H and c.

Then for any $\mathbf{v} \in \mathbb{R}^d$ we have

$$\begin{aligned}
|\mathbf{v}^\top \boldsymbol{\Omega} \mathbf{v}^\top - \mathbf{v}^\top \widetilde{\boldsymbol{\Omega}} \mathbf{v}^\top| &= |\mathbf{v}^\top \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|y])\mathbf{v} - \mathbf{v}^\top \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|\tilde{y}])\mathbf{v}| \\
&= \left| \mathbf{v}^\top \sum_h^H \mathbb{P}_h \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|y]|y \in J_h)\mathbf{v} - \mathbf{v}^\top \sum_h^H \mathbb{P}_h \mu_h \mu_h^\top \mathbf{v} \right| \\
&= \sum_h^H \mathbb{P}_h \left| \mathbf{v}^\top \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|y]|y \in J_h)\mathbf{v} - \mathbf{v}^\top \mu_h \mu_h^\top \mathbf{v} \right| \\
&\leq \left(\frac{1}{H} + \varepsilon \right) \sum_h \mathbf{v}^\top \mathrm{cov}(\mathbb{E}[\boldsymbol{x}|y]|y \in J_h)\mathbf{v} \\
&\leq (1 + H\varepsilon)\frac{\alpha_3}{H^\vartheta} \mathbf{v}^\top \boldsymbol{\Omega} \mathbf{v}
\end{aligned}$$

where the last inequality follows from the Stability Assumption (10). Taking maximum over norm 1 vectors yields $\|\boldsymbol{\Omega} - \widetilde{\boldsymbol{\Omega}}\|_2 \leq (1 + H\varepsilon)\frac{\alpha_3}{H^\vartheta}\rho_1$. From Weyl's inequality we have $|\rho_K(\boldsymbol{\Omega}) - \rho_K(\widetilde{\boldsymbol{\Omega}})| \leq \|\boldsymbol{\Omega} - \widetilde{\boldsymbol{\Omega}}\|_2$. Thus $\rho - (1 + H\varepsilon)\frac{\alpha_3}{H^\vartheta}\rho_1 < \rho_K(\widetilde{\boldsymbol{\Omega}})$, and serves as a lower bound. Then we can replace $\tilde{\gamma}$ with $\frac{\rho - (1+H\varepsilon)\frac{\alpha_3}{H^\vartheta}}{M}$. This completes the proof. □

3.5 Canonical Correlation Analysis

The GEV estimator can also be applied to the Canonical Correlation Analysis problem [Hot33], which has also had a number of techniques proposed in the past two decades for performing the task under HDLSS conditions where sparsity is assumed on the canonical directions [WKI08, WTH09, PTB09, HST11]. [CGRZ13] first gives the characterization of the probabilistic CCA

model for sparse canonical directions, and presents the CAPIT method for the problem. Rates of convergence are given that depend on an independent estimate of the precision matrices of the two data sources, which can often be difficult to compute even diagonal approximations of. A modern standard for estimation in this problem is Penalized Multivariate Analysis (PMA) method [WTH09] that estimates a regularized version of the singular value decomposition. Our estimator is shown to perform better empirically on simulations for sparse CCA.

Canonical Correlation Analysis is a classical technique that finds the linear combination of two sets of random variables with maximal correlation. It has been applied to a number of different fields, including pyschology, neurology, genomics and economics. Let $\boldsymbol{x} \in \mathbb{R}^{d_1}$ and $\boldsymbol{y} \in \mathbb{R}^{d_2}$ be zero-mean random vectors with joint covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_x & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_y \end{pmatrix},$$

where $\boldsymbol{\Sigma}_x = (\Sigma_{x,k\ell})$ and $\boldsymbol{\Sigma}_y = (\Sigma_{y,k\ell})$ are the covariance matrices for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $\boldsymbol{\Sigma}_{xy} = (\Sigma_{xy,k\ell}) = \boldsymbol{\Sigma}_{yx}^\top$ is the cross-covariance matrix between $\boldsymbol{x}$ and $\boldsymbol{y}$. Then CCA determines the K canonical direction vectors by solving

$$\max \mathbf{v}_{xk}^\top \boldsymbol{\Sigma}_{xy} \mathbf{v}_{yk}, \quad \text{subject to} \quad \mathbf{v}_{xk}^\top \boldsymbol{\Sigma}_x \mathbf{v}_{xk} = \mathbf{v}_{yk}^\top \boldsymbol{\Sigma}_y \mathbf{v}_{yk} = 1, \mathbf{v}_{xk}^\top \boldsymbol{\Sigma}_x \mathbf{v}_{xj} = \mathbf{v}_{yk}^\top \boldsymbol{\Sigma}_y \mathbf{v}_{yj} = 0 \tag{3.17}$$

for $k \in [K]$ and $j < k$. The optimization problem can be solved by applying singular value decomposition on the matrix $\boldsymbol{\Sigma}_x^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_y^{-1/2}$, and a sample version is given by replacing the covariances with their usual estimators. This leads to consistent estimation of the canonical directions when the dimensions d_1 and d_2 are fixed and the sample size n increases.

In the high-dimensional setting, when the dimensions exceed the sample size, one cannot compute the inverse sample covariances. This leads to the structural assumption of sparsity in the canonical directions, which allows for successful estimation. As shown in [CGRZ13], the set of $(\mathbf{v}_{xk}, \mathbf{v}_{yk})$ are solutions to 3.17 if and only if

$$\boldsymbol{\Sigma}_{xy} = \boldsymbol{\Sigma}_x \left(\sum_{k=1}^{K} \lambda_k \mathbf{v}_{xk} \mathbf{v}_{yk}^\top \right) \boldsymbol{\Sigma}_y$$

for some $\lambda_k > 0$, giving the correlation weights. We show that (3.17) is a special case of the generalized eigenvalue problem (3.1) with

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{0} & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{yx} & \mathbf{0} \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_y \end{pmatrix}, \quad \text{and} \quad \mathbf{v}_k = \begin{pmatrix} \mathbf{v}_{xk} \\ \mathbf{v}_{yk} \end{pmatrix}.$$

Substituting the above into (3.1), we yield

$$\mathbf{v}_k^* = \underset{\mathbf{v}_{xk}, \mathbf{v}_{yk}}{\operatorname{argmax}} \frac{2\mathbf{v}_{xk}^\top \mathbf{\Sigma}_{xy} \mathbf{v}_{yk}}{\mathbf{v}_{xk}^\top \mathbf{\Sigma}_x \mathbf{v}_{xk} + \mathbf{v}_{yk}^\top \mathbf{\Sigma}_y \mathbf{v}_{yk}}$$

It is straightforward to show this is equivalent to (3.17).

Now if we assume that the $\mathbf{\Sigma}_{xy}$ has the following singular value decomposition

$$\mathbf{\Sigma}_{xy} = \sum_{k=1}^{K} \sigma_k \boldsymbol{u}_{x,k} \boldsymbol{\nu}_{y,k}^\top$$

so that $\mathbf{\Sigma}_{xy} = \mathbf{O}_x \mathbf{D} \mathbf{O}_y^\top$. Then we have

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{0} & \mathbf{O}_x \mathbf{D} \mathbf{O}_y^\top \\ \mathbf{O}_x \mathbf{D} \mathbf{O}_y^\top & \mathbf{0} \end{pmatrix}.$$

We need the following lemma which gives the eigendecomposition of $\mathbf{\Omega}$.

Lemma 27. *We have* $\mathbf{\Omega} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top$, *where*

$$\mathbf{Q} = \begin{pmatrix} \mathbf{O}_x & \mathbf{O}_x \\ \mathbf{O}_y & -\mathbf{O}_y \end{pmatrix} \quad \text{and} \quad \mathbf{\Lambda} = \begin{pmatrix} \mathbf{D} & 0 \\ 0 & -\mathbf{D} \end{pmatrix}.$$

Proof. The proof of this lemma follows from direct calculation. □

Using Lemma 27, we observe that $\mathbf{U}$ in (3.4) can be taken as $(\mathbf{O}_x^\top, \mathbf{O}_y^\top)^\top / \sqrt{2}$. An estimator of $\mathbf{U}$ can be obtained by concatenating the top rank-K left and right singular matrix of $\widehat{\mathbf{\Sigma}}_{xy} = n^{-1} \sum_{i=1}^n \mathbf{X}_i \mathbf{Y}_i^\top$, where $\mathbf{X}_i$'s and $\mathbf{Y}_i$'s are independent and identically distributed samples of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Thus we may apply the GEV estimator (3.5) to the CCA problem with the estimates for $\widehat{U}$ and $\widehat{\mathbf{\Sigma}}$ defined above, and recover the spaces given by $\operatorname{span}\{\mathbf{v}_{x1}, \ldots, \mathbf{v}_{xK}\}$ and $\operatorname{span}\{\mathbf{v}_{y1}, \ldots, \mathbf{v}_{yK}\}$.

CHAPTER 4

EMPIRICAL RESULTS OF THE GEV ESTIMATOR

4.1 Implementation

To efficiently solve for **W** from (3.5), we implement from [BT09] the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). FISTA is an alteration of the iterative first order method ISTA used to solve ℓ_1-regularized convex optimization problems. The alteration uses a version of Nesterov acceleration [Nes83] to achieve a convergence rate of $O(1/k^2)$. Define $S_\lambda(\cdot)$, a element-wise soft thresholding operator, the gradient of the (3.4), as follows:

$$\big(S_\lambda(A)\big)_{ij} = \begin{cases} \operatorname{sgn}(a_{ij})|a_{ij} - \lambda|, & \text{if} |a_{ij}| > \lambda \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 4.1: A fast iterative shrinkage-thresholding algorithm for GEV.

Input: U, Σ, and λ. Initialization: take $W_X^{(0)} \in \mathbb{R}^{d\times K}$, $\rho^{-1} = 1/\rho_1(\Sigma)$, $t^{(0)} = 1$, and $k = 0$
Output: W

repeat
  $W_Y^{(k+1)} = S_{\rho^{-1}\lambda}\big(W_X^{(k)} - \rho^{-1}(\Sigma W_X^{(k)} - U)\big)$;
  $t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$;
  $W_X^{(k+1)} = W_Y^{(k+1)} + \frac{t^{(k)}-1}{t^{(k+1)}}(W_Y^{(k+1)} - W_Y^{(k)})$
until converge;

Since the GEV problem estimator acts like a matrix version of Lasso regression, there are many algorithms one could potentially apply to a ℓ_1-regularized optimization problem. We sought a comparison of the following methods to solve the problem for a fixed choice of the hyper-parameter λ giving the regularization weight: Subgradient [Nes04, Chapter 3.1], Proximal method [Roc70], FISTA, ADMM [BPC$^+$11], and Chambolle-Pock [CP11]. The model used to test the convergence is taken from the SIR simulations below, using Model 1 in the continuous responses, with Gaussian features and noise, and $d = 50, n = 200$.

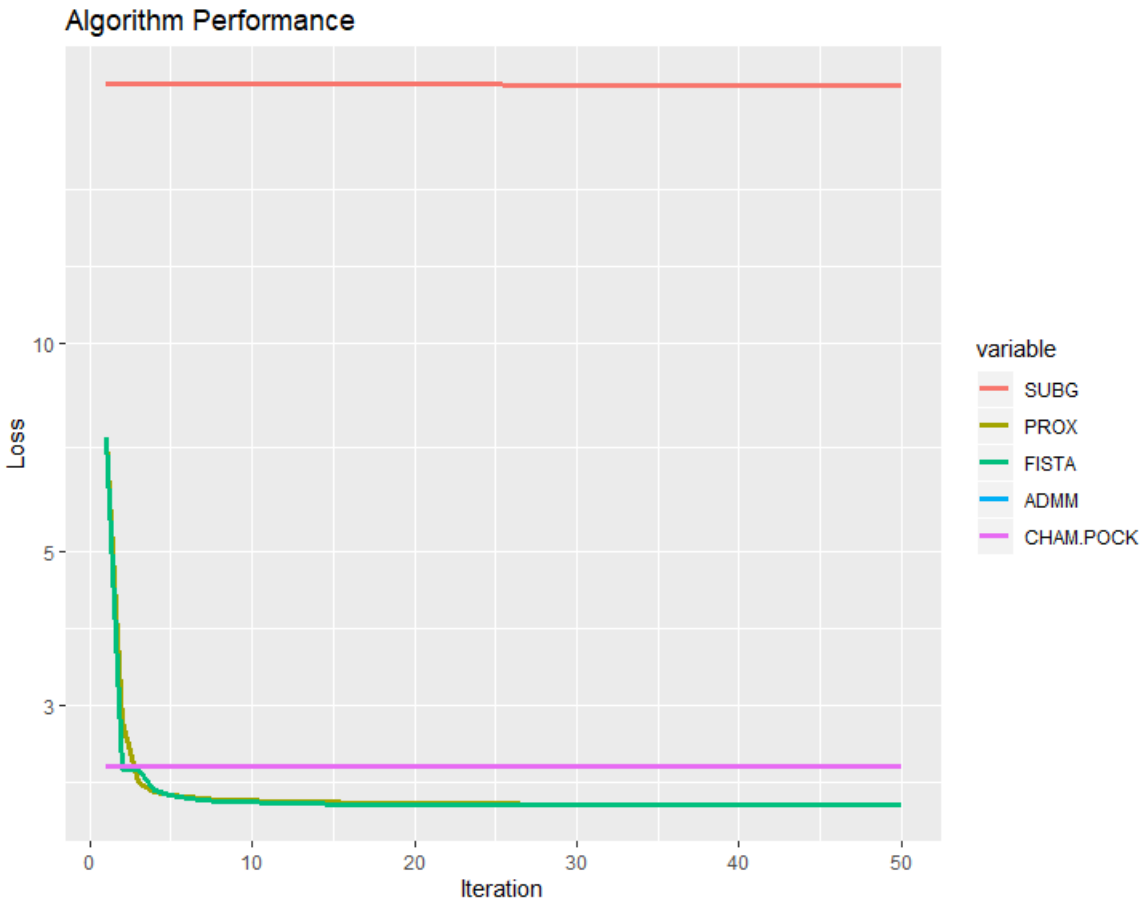

Figure 4.1: Comparison of convergence rates of different algorithms.

As expected, in Figure (4.1) we see the subgradient method was suboptimal compared to all other methods, with a very slow convergence rate. However, due to the incredibly low number of iterations required to reach a stable critical point, empirically all the other methods performed equivalently in terms of iterations, with the exception of Chambolle-Pock, which had problems with convergence after the first iteration. As well, important differences occurred in run-time; ADMM in particular requires the computation of a matrix inverse, which drastically increases its run-time. Due to the simplicity of implementation, the speed of convergence and quick run-time, we maintained the FISTA implementation of the GEV method above.

4.1.1 Robust Modification

This algorithm is sufficient for most applications of the GEV algorithm, but an important case arises for data that comes from heavy noise distributions or data with outliers. In many applications, the assumption of sub-Gaussian tails is unrealistic; applications using functional magnetic resonance imaging (fMRI) [ENK16] or microarray data giving gene expression level [WPL15] have been observed to have heavy tails and large kurtosis, regardless of normalization methods used. The kurtosis of a random variable X is defined as the fourth centralized moment $\mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$, and high values indicate either that the probability mass is concentrated around the mean and the data-generating process produces occasional values far from the mean (i.e. outliers), or that the probability mass is concentrated in the tails of the distribution.

To that end, we robustify the matrix square loss by introducing the following matrix Huber loss as a substitute to the Frobenius loss term in (3.2)

$$\mathcal{L}_\alpha = \sum_{ij} \ell_\alpha \left([\boldsymbol{\Sigma}\mathbf{W}]_{ij} - \mathbf{U}_{ij} \right)$$

where

$$\ell_\alpha(x) = \begin{cases} x^2, & \text{if} |x| \leq \frac{1}{\alpha} \\ 2\alpha^{-1}|x| - \alpha^{-2} & \text{otherwise.} \end{cases}$$

The Huber loss [Hub73], $\ell_\alpha(x)$ is quadratic for small values of x, and becomes linear when x gets larger. The parameter α controls the blending of quadratic and linear loss. The least square loss and least absolute deviation (LAD) loss can be regarded as two extremes of the Huber loss for $\alpha = 0$ and $\alpha = \infty$, respectively. Using this loss in the ℓ_1-regularized scheme, we get the estimate

$$\widehat{\mathbf{W}}_{\alpha,\lambda} = \underset{\mathbf{W}\in\mathbb{R}^{d\times K}}{\operatorname{argmin}} \left\{ \sum_{ij} \ell_\alpha \left([\widehat{\boldsymbol{\Sigma}}\mathbf{W}]_{ij} - \widehat{\mathbf{U}}_{ij} \right) + \lambda \|\mathbf{W}\|_{1,1} \right\} \tag{4.1}$$

Notice however that the term $(\boldsymbol{\Sigma}^{1/2}\mathbf{W} - \boldsymbol{\Sigma}^{-1/2}\mathbf{U})$ has been replaced in the Huber loss for the term $(\boldsymbol{\Sigma}\mathbf{W} - \mathbf{U})$. As noted after the proof of Theorem 1, either expression when used in the Frobenius norm loss yields the same solution for the space of $\mathcal{V}$, but will give different models when combined with the ℓ_1-regularization term and used in (3.4). Computationally, if the expression

$(\mathbf{\Sigma W} - \mathbf{U})$ is used in Algorithm **??**, the gradient of the expression requires computation of $\mathbf{\Sigma}^2$, which leads to greater run-time and worse performance likely due to floating point error in compared to the performance of the expression $(\mathbf{\Sigma}^{1/2}\mathbf{W} - \mathbf{\Sigma}^{-1/2}\mathbf{U})$. However, when using the Huber loss, the gradient computation of this model will lead to the computation of $\mathbf{\Sigma}^{-1/2}$ if we use expression $(\mathbf{\Sigma}^{1/2}\mathbf{W} - \mathbf{\Sigma}^{-1/2}\mathbf{U})$. This is statistically and computationally undesirable, especially in the case of $d > n$ where $\widehat{\mathbf{\Sigma}}$ is singular. Then with D_α as the gradient of the Huber loss, we define the algorithm below.

$$\big(D_\alpha(A)\big)_{ij} = \begin{cases} 2a_{ij} & \text{if} |a_{ij}| \leq \alpha^{-1} \\ 2\alpha^{-1}\text{sgn}(a_{ij}), & \text{otherwise.} \end{cases}$$

Algorithm 4.2: Huber loss algorithm for robust GEV.

Input: U, Σ, λ and α. Initialization: take $W_X^{(0)} \in \mathbb{R}^{d\times K}$, $\rho^{-1} = 1/\rho_1(\Sigma)$, $t^{(0)} = 1$, and $k = 0$

Output: W

repeat

- $W_Y^{(k+1)} = S_{\alpha\lambda}\big(W_X^{(k)} - \rho^{-2}\Sigma D_\alpha(\Sigma W_X^{(k)} - U)\big)$;
- $t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$;
- $W_X^{(k+1)} = W_Y^{(k+1)} + \frac{t^{(k)}-1}{t^{(k+1)}}(W_Y^{(k+1)} - W_Y^{(k)})$

until converge;

4.2 Sliced Inverse Regression

We compare our method of applying GEV to the sliced inverse regression problem against the classical method we label SIR [Li91], and the modern method LassoSIR (LSIR) [LZL19], used for high dimensional problems. To facilitate a fair comparison with SIR and LSIR, all the simulation studies are generated under forward models including both categorical and continuous responses for low ($d = 100$) and high ($d = 1000$) dimensional predictors. Throughout the simulations, we use a K-fold cross-validation (CV) to select the tuning parameters and quantify the estimation accuracy using three different metrics defined as follows: the canonical correlation (CCA) between

$\boldsymbol{x}^\top\hat{\mathbf{W}}$ and $\boldsymbol{x}^\top\mathbf{W}$; the Frobenius norm distance (FD) between $\boldsymbol{P}_{\mathbf{W}}$ and its estimate $\mathbf{P}_{\hat{\mathbf{W}}}$; the trace correlation (TC) defined as $\text{tr}(\mathbf{P}_{\mathbf{W}}\mathbf{P}_{\hat{\mathbf{W}}})/K$ with K being the structural dimensions. Let $\Sigma = (\sigma_{ij})_{d\times d}$, where $\sigma_{ij} = 0.5^{|i-j|}$ and d is taken to be 50 or 500. To demonstrate the robustness of SDR for categorical response, we consider two simulation scenarios for generating the predictor variables $\boldsymbol{x}$: 1) from $\mathcal{N}(0,\Sigma)$ and 2) from $t_5(0,\Sigma)$. Let β_1 and β_2 be the $d-$dimensional vectors with their first six elements being $(1,1,1,1,1,1)/\sqrt{6}$ and $(1,-1,1,-1,1,-1)/\sqrt{6}$ and the rest being zero.

The response Y is generated from the multinomial distribution with

$$\Pr(y=k) = \frac{f_k(\boldsymbol{x})}{1+\sum_{j=1}^{K-1} f_j(\boldsymbol{x})}, \quad k = 1,\ldots,k-1,$$

where K is the number of categories and $f_k(\boldsymbol{x})$ is the component connecting $\boldsymbol{x}$ with y. We consider the following two different models of $f_k(\boldsymbol{x})$:

1. Model 1: $f_k(\boldsymbol{x}) = \sin(\boldsymbol{x}^\top\beta_k/4)+1$;
2. Model 2: $f_k(\boldsymbol{x}) = \exp(\boldsymbol{x}^\top\beta_k)$.

For both models, the $f_k(\boldsymbol{x})$ components are monotone within the domain of $\boldsymbol{x}$, so they are favorable to SIR. Moreover, we can see that model 2 is actually the multinomial logistic regression.

The simulation includes comparing all combination of the two scenarios, two models and the two configurations $(n,d) = (200,50)$ and $(500,1000)$ with 100 replicates. As shown in Table 4.1, GEV-SIR dominates SIR and LSIR for all three metrics.

Table 4.1: Summary of estimation accuracy for categorical response in low and high dimensions. We report the means of three accuracy metrics (CCA, FD and TC) with their standard deviations in parentheses. The results are based on 100 replications.

Sample and Error Type	Model	Method	(d,n)	FD	TC	CCA
Gaussian-X	Model 1	**GEV**	50, 200	**3.38 (.359)**	**.154 (.09)**	**.602 (.188)**
		SIR	50, 200	3.75 (.137)	.062 (.034)	.393 (.098)
		LSIR	50, 200	3.49 (.397)	.127 (.099)	.562 (.232)
	Model 2	**GEV**	50, 200	**2.06 (.321)**	**.484 (.08)**	**.950 (.035)**
		SIR	50, 200	2.90 (.237)	.276 (.059)	.811 (.042)
		LSIR	50, 200	2.26 (.386)	.436 (.096)	.945 (.034)
	Model 1	**GEV**	1000, 500	**3.61 (.319)**	**.097 (.08)**	.209 (.120)
		LSIR	1000, 500	3.71 (.326)	.073 (.082)	**.305 (.206)**
	Model 2	**GEV**	1000, 500	**1.97 (.379)**	**.507 (.095)**	.495 (.146)
		LSIR	1000, 500	2.08 (.432)	.479 (.108)	**.718 (.178)**
Elliptical-X	Model 1	**GEV**	50, 200	**3.31 (.383)**	**.174 (.096)**	**.652 (.192)**
		SIR	50, 200	3.73 (.165)	.068 (.041)	.420 (.100)
		LSIR	50, 200	3.43 (.424)	.143 (.106)	.620 (.218)
	Model 2	**GEV**	50, 200	**1.87 (.385)**	**.534 (.096)**	**.954 (.034)**
		SIR	50, 200	2.83 (.275)	.292 (.069)	.813 (.035)
		LSIR	50, 200	2.10 (.400)	.474 (.101)	.951 (.029)
	Model 1	**GEV**	1000, 500	**3.41 (.436)**	**.149 (.109)**	.353 (.182)
		LSIR	1000, 500	3.48 (.416)	.131 (.104)	**.443 (.208)**
	Model 2	**GEV**	1000, 500	**1.88 (.461)**	**.530 (.115)**	.618 (.169)
		LSIR	1000, 500	1.89 (.486)	.527 (.121)	**.727 (.167)**

For continuous response, we consider the following four scenarios for both low ($d = 50$) and high ($d = 1000$) dimensional data with either

1 : Gaussian predictors and Gaussian noise.

2 : Gaussian predictors and elliptical noise.

3 : Elliptical predictors and Gaussian noise.

We randomly generate $n = 500$ predictors $\boldsymbol{x}$ from either a multivariate normal or elliptical distribution with mean zero and and the same covariance matrix as in categorical cases. For the continuous responses, we then generate the responses variable according to the following three models:

1. Model 1: $y = (\boldsymbol{x}^\top \beta_1)/\{0.5 + (\boldsymbol{x}^\top \beta_2 + 1.5)^2\} + 0.5\varepsilon$;

2. Model 2: $y = \boldsymbol{x}^\top \beta_1 + 2 + \exp(\boldsymbol{x}^\top \beta_2) + 0.5 * \varepsilon |\boldsymbol{x}^\top \beta_1 + 2|$,

3. Model 3: $y = (\boldsymbol{x}^\top \beta_1 + 1)^2 + (\boldsymbol{x}^\top \beta_2 + 1)^2 + 0.5 * \varepsilon$,

The ε's are independently generated from either standard normal or t_5 distribution. Here we set $\beta_1 = (1,\ldots,1,0,\ldots,0)^\top/\sqrt{6}$ and $\beta_2 = (1,-1,1,-1,1,-1,0,\ldots,0)^\top/\sqrt{6}$ with the first 6 elements of both vectors being nonzero. The results are found in tables 4.2 and 4.3.

Table 4.2: Summary of estimation accuracy for continuous response in low dimensions. We report the means of three accuracy metrics (CCA, FD and TC) with their standard deviations in parentheses. The results are based on 100 replications.

Sample and Error Type	Model	Method	(d,n)	FD	TC	CCA
Gaussian-X, Gaussian-Error	Model 1	GEV	50, 200	2.27 (.247)	.433 (.062)	**.974 (.018)**
		SIR	50, 200	3.22 (.293)	.196 (.073)	.793 (.061)
		LSIR	50, 200	**2.16 (.220)**	**.460 (.055)**	.973 (.017)
	Model 2	**GEV**	50, 200	**2.08 (.120)**	**.478 (.030)**	**.985 (.014)**
		SIR	50, 200	2.66 (.225)	.335 (.056)	.859 (.049)
		LSIR	50, 200	2.11 (.136)	.474 (.034)	.971 (.019)
	Model 3	**GEV**	50, 200	**2.11 (.179)**	**.472 (.045)**	**.982 (.020)**
		SIR	50, 200	3.19 (.348)	.202 (.087)	.733 (.138)
		LSIR	50, 200	2.19 (.199)	.452 (.050)	.965 (.030)
Gaussian-X, Elliptical-Error	Model 1	GEV	50, 200	2.38 (.303)	.404 (.076)	**.962 (.027)**
		SIR	50, 200	3.47 (.237)	.131 (.059)	.683 (.111)
		LSIR	50, 200	**2.30 (.262)**	**.424 (.065)**	.961 (.024)
	Model 2	**GEV**	50, 200	**2.08 (.183)**	**.478 (.046)**	**.983 (.018)**
		SIR	50, 200	2.82 (.243)	.296 (.061)	.813 (.060)
		LSIR	50, 200	2.13 (.171)	.469 (.043)	.965 (.023)
	Model 3	**GEV**	50, 200	**2.18 (.278)**	**.456 (.070)**	**.972 (.068)**
		SIR	50, 200	3.21 (.351)	.199 (.088)	.734 (.128)
		LSIR	50, 200	2.24 (.278)	.439 (.070)	.958 (.056)
Elliptical-X, Gaussian-Error	Model 1	GEV	50, 200	2.11 (.290)	.473 (.072)	**.978 (.016)**
		SIR	50, 200	3.21 (.299)	.199 (.075)	.786 (.070)
		LSIR	50, 200	**2.04 (.310)**	**.490 (.077)**	.961 (.017)
	Model 2	**GEV**	50, 200	**2.25 (.261)**	**.437 (.065)**	**.953 (.056)**
		SIR	50, 200	2.99 (.327)	.253 (.082)	.770 (.094)
		LSIR	50, 200	2.33 (.300)	.418 (.075)	.929 (.060)
	Model 3	**GEV**	50, 200	**2.75 (.434)**	**.312 (.108)**	**.828 (.137)**
		SIR	50, 200	3.50 (.286)	.126 (.067)	.558 (.154)
		LSIR	50, 200	2.85 (.437)	.287 (.109)	.801 (.144)

Table 4.3: Summary of estimation accuracy for continuous response in high dimensions. We report the means of three accuracy metrics (CCA, FD and TC) with their standard deviations in parentheses. The results are based on 100 replications.

Sample and Error Type	Model	Method	(d,n)	FD	TC	CCA
Gaussian-X, Gaussian-Error	Model 1	**GEV**	1000, 500	**3.06 (.133)**	**.236 (.033)**	**.898 (.028)**
		LSIR	1000, 500	3.13 (.123)	.217 (.031)	.889 (.029)
	Model 2	**GEV**	1000, 500	**2.69 (.164)**	**.329 (.041)**	**.907 (.027)**
		LSIR	1000, 500	2.79 (.152)	.303 (.038)	.895 (.027)
	Model 3	**GEV**	1000, 500	**3.13 (.256)**	**.216 (.064)**	**.828 (.076)**
		LSIR	1000, 500	3.23 (.223)	.194 (.056)	.812 (.084)
Gaussian-X, Elliptical-Error	Model 1	**GEV**	1000, 500	**3.25 (.238)**	**.187 (.060)**	**.827 (.133)**
		LSIR	1000, 500	3.31 (.235)	.171 (.059)	.813 (.137)
	Model 2	**GEV**	1000, 500	**2.84 (.291)**	**.291 (.073)**	**.879 (.058)**
		LSIR	1000, 500	2.97 (.247)	.259 (.062)	.851 (.085)
	Model 3	**GEV**	1000, 500	**3.19 (.296)**	**.203 (.074)**	**.817 (.078)**
		LSIR	1000, 500	3.25 (.232)	.186 (.058)	.804 (.088)
Elliptical-X, Gaussian-Error	Model 1	**GEV**	1000, 500	3.12 (.176)	**.220 (.044)**	**.866 (.036)**
		LSIR	1000, 500	**3.11 (1.01)**	.183 (.044)	.830 (.068)
	Model 2	**GEV**	1000, 500	**3.37 (.193)**	**.158 (.048)**	**.777 (.070)**
		LSIR	1000, 500	3.49 (.174)	.127 (.044)	.743 (.088)
	Model 3	**GEV**	1000, 500	**3.87 (.088)**	**.032 (.022)**	**.417 (.138)**
		LSIR	1000, 500	3.91(.081)	.023 (.020)	.353 (.159)

4.2.1 Heavy Noise Slice Inverse Regression

Here we show our adapted Huber loss GEV method when applied to the SIR problem with increased noise. We use the same models in the continuous response section with all the same conditions, with the exception of the coefficient of the noise term being raised from 0.5 to 1. As well we included an additional model

$$\text{Model 4: } y = (\boldsymbol{x}^\top \beta_1 + 3)^2 + 2|\boldsymbol{x}^\top \beta_2 + 3| + \varepsilon|\boldsymbol{x}^\top \beta_2|;$$

and the scenario of having elliptical features for $\boldsymbol{x}$ and elliptical noise. We compare this to both SIR and LSIR.

Table 4.4: Summary of estimation accuracy for Huber loss estimation in low dimensions with high noise. We report the means of three accuracy metrics (CCA, FD and TC) with their standard deviations in parentheses. The results are based on 100 replications.

Sample and Error Type	Model	Method	(d,n)	FD	TC	CCA
Gaussian-X, Gaussian-Error	Model 1	**GEV**	50, 200	**3.25 (.321)**	**.188 (.080)**	**.699 (.153)**
		SIR	50, 200	3.83 (.125)	.04 (.031)	.286 (.114)
		LSIR	50, 200	3.56 (.318)	.108 (.080)	.573 (.167)
	Model 2	**GEV**	50, 200	**2.00 (.238)**	**.498 (.080)**	.992 (.047)
		SIR	50, 200	2.27 (.346)	.432 (.087)	.979 (.102)
		LSIR	50, 200	2.01 (.234)	.497 (.058)	**.997 (.041)**
	Model 3	**GEV**	50, 200	**2.41 (.332)**	**.397 (.083)**	**.992 (.075)**
		SIR	50, 200	3.55 (.249)	.111 (.062)	.560 (.131)
		LSIR	50, 200	2.43 (.398)	.390 (.100)	.915 (.089)
	Model 4	**GEV**	50, 200	**2.43 (.376)**	**.391 (.094)**	**.913 (.158)**
		SIR	50, 200	3.61 (.227)	.099 (.058)	.523 (.178)
		LSIR	50, 200	2.53 (.434)	.367 (.108)	.900 (.176)
Gaussian-X, Elliptical-Error	Model 1	**GEV**	50, 200	**3.28 (.374)**	**.181 (.096)**	**.651 (.176)**
		SIR	50, 200	3.84 (.116)	.040 (.029)	.290 (.096)
		LSIR	50, 200	3.60 (.303)	.100 (.076)	.545 (.159)
	Model 2	**GEV**	50, 200	**2.02 (.260)**	**.493 (.065)**	.991 (.046)
		SIR	50, 200	2.38 (.356)	.406 (.089)	.971 (.093)
		LSIR	50, 200	2.02 (.270)	.492 (.068)	**.995 (.040)**
	Model 3	**GEV**	50, 200	**2.47 (.309)**	**.381 (.077)**	**.915 (.097)**
		SIR	50, 200	3.62 (.242)	.096 (.061)	.511 (.159)
		LSIR	50, 200	2.54 (.364)	.366 (.091)	.911 (.105)
	Model 4	**GEV**	50, 200	**2.59 (.416)**	**.353 (.104)**	**.890 (.169)**
		SIR	50, 200	3.71 (.250)	.073 (.063)	.431 (.180)
		LSIR	50, 200	2.69 (.489)	.329 (.122)	.874 (.193)
Elliptical-X, Gaussian-Error	Model 1	**GEV**	50, 200	**3.17(.389)**	**.207 (.097)**	**.727 (.203)**
		SIR	50, 200	3.83 (.117)	.043 (.029)	.309 (.097)
		LSIR	50, 200	3.44 (.249)	.141 (.062)	.666 (.176)
	Model 2	**GEV**	50, 200	**2.17 (.072)**	**.456 (.018)**	.955 (.006)
		SIR	50, 200	2.75 (.148)	.312 (.037)	.887 (.008)
		LSIR	50, 200	2.19 (.045)	.451 (.011)	**.966 (.002)**
	Model 3	**GEV**	50, 200	**2.58 (.301)**	**.355 (.075)**	.881 (.065)
		SIR	50, 200	3.54 (.302)	.114 (.075)	.621 (.170)
		LSIR	50, 200	2.63 (.419)	.341 (.105)	**.888 (.117)**
	Model 4	**GEV**	50, 200	**2.96 (.341)**	**.261 (.085)**	**.742 (.103)**
		SIR	50, 200	3.68 (.248)	.080 (.062)	.463 (.192)
		LSIR	50, 200	3.06 (.429)	.234 (.107)	.741 (.136)
Elliptical-X, Elliptical-Error	Model 1	**GEV**	50, 200	**3.27 (.368)**	**.183 (.092)**	**.670 (.188)**
		SIR	50, 200	3.84 (.112)	.041 (.028)	.287 (.092)
		LSIR	50, 200	3.51 (.275)	.121 (.069)	.621 (.175)
	Model 2	**GEV**	50, 200	**2.19 (.070)**	**.454 (.017)**	.953 (.006)
		SIR	50, 200	2.81 (.187)	.296 (.047)	.878 (.011)
		LSIR	50, 200	2.20 (.071)	.450 (.018)	**.964 (.003)**
	Model 3	**GEV**	50, 200	**2.66 (.268)**	**.334 (.067)**	.860 (.062)
		SIR	50, 200	3.61 (.237)	.097 (.059)	.553 (.161)
		LSIR	50, 200	2.73 (.334)	.317 (.084)	**.871 (.078)**
	Model 4	**GEV**	50, 200	**3.00 (.337)**	**.249 (.084)**	**.725 (.088)**
		SIR	50, 200	3.70 (.198)	.076 (.049)	.447 (.172)
		LSIR	50, 200	3.12 (.434)	.218 (.108)	.712 (.127)

Table 4.5: Summary of estimation accuracy for Huber loss estimation in high dimensions with high noise. We report the means of three accuracy metrics (CCA, FD and TC) with their standard deviations in parentheses. The results are based on 100 replications.

Sample and Error Type	Model	Method	(d,n)	FD	TC	CCA
Gaussian-X, Gaussian-Error	Model 1	**GEV**	1000, 500	**3.55 (.100)**	**.112 (.025)**	.282 (.113)
		LSIR	1000, 500	3.65 (.116)	.087 (.029)	**.464 (129)**
	Model 2	GEV	1000, 500	**2.12 (.042)**	**.470 (.010)**	**.670 (.157)**
		LSIR	1000, 500	2.22 (.079)	.445 (.020)	.448 (.152)
	Model 3	**GEV**	1000, 500	**2.68 (.160)**	**.329 (.040)**	**.647 (.110)**
		LSIR	1000, 500	2.92 (.192)	.271 (.048)	.389 (.130)
	Model 4	**GEV**	1000, 500	**3.13 (.215)**	**.216 (.054)**	**.578 (.102)**
		LSIR	1000, 500	3.31 (.177)	.173 (.044)	.320 (.130)
Gaussian-X, Elliptical-Error	Model 1	**GEV**	1000, 500	**3.67 (.107)**	**.082 (.027)**	**.384 (.108)**
		LSIR	1000, 500	3.80 (.112)	.050 (028)	.266 (.114)
	Model 2	GEV	1000, 500	**2.13 (.044)**	**.467 (.011)**	**.665 (.159)**
		LSIR	1000, 500	2.23 (.080)	.441 (.022)	.472 (.160)
	Model 3	**GEV**	1000, 500	**2.79 (.167)**	**.302 (.042)**	**.614 (.105)**
		LSIR	1000, 500	3.03 (.191)	.234 (.048)	.377 (.125)
	Model 4	**GEV**	1000, 500	**3.17 (.208)**	**.206 (.052)**	**.586 (.097)**
		LSIR	1000, 500	3.32 (.188)	.170 (.047)	.327 (.137)
Elliptical-X, Gaussian-Error	Model 1	**GEV**	1000, 500	**3.52 (.151)**	**.121 (.038)**	**.519 (.106)**
		LSIR	1000, 500	3.61 (.134)	.098 (.034)	.337 (.125)
	Model 2	**GEV**	1000, 500	**2.66 (.240)**	**.334 (.060)**	**.572 (.123)**
		LSIR	1000, 500	3.10 (.249)	.224 (.062)	.385 (.139)
	Model 3	**GEV**	1000, 500	**3.40 (.188)**	**.149 (.047)**	**.476 (.099)**
		LSIR	1000, 500	3.59 (.164)	.101 (.041)	.334 (.115)
	Model 4	**GEV**	1000, 500	**3.84 (.083)**	**.040 (.021)**	.228 (.075)
		LSIR	1000, 500	3.94 (.064)	.015 (.016)	**.236 (.118)**
Elliptical-X, Elliptical-Error	Model 1	**GEV**	1000, 500	**3.68 (.116)**	**.083 (.023)**	**.420 (.085)**
		LSIR	1000, 500	3.76 (123)	.059 (.031)	.309 (.125)
	Model 2	**GEV**	1000, 500	**2.68 (.213)**	**.330 (.053)**	**.589 (.116)**
		LSIR	1000, 500	3.14 (.225)	.216 (.064)	.456 (.135)
	Model 3	**GEV**	1000, 500	**3.44 (.193)**	**.138 (.048)**	**.495 (.080)**
		LSIR	1000, 500	3.61 (.153)	.097 (.038)	.394 (.108)
	Model 4	**GEV**	1000, 500	**3.84 (.087)**	**.039 (022)**	.228 (.076)
		LSIR	1000, 500	3.94 (.070)	.015 (.018)	**.230 (.123)**

4.3 Linear Discriminant Analysis

In this section, we investigated the performance of GEV method under high-dimensional Linear Discriminant Analysis (LDA) framework for both binary and multi-class classification problems by applying them to simulated data generated under two types of within group covariance matrices: block Toeplitz suggested by [WT11] and Sparse precision matrix as described above. For comparison, we also included the ℓ_1-penalized linear discriminant analysis (LDA-$\ell 1$) [WT11] us-

ing R package penalizedLDA and the direct approach for discriminant analysis [MZY12, ZMY18] implemented in R packages dsda and msda for binary or multi-class cases respectively. To serve as a benchmark, we also included the Oracle classifier derived from the population parameters Σ_w and Σ_b. For each simulation, we generate $100K$ samples with $d = 500$ features, where K is the number of classes. We consider simulation settings for binary and multi-class cases as follows:

- Binary case: we set $\mu_1 = \mathbf{0}$ and $\mu_{2j} \sim N(0.3, 0.5)$ for $j \in \{1, \ldots, 20\}$ and $\mu_{2j} = 0$ otherwise. For the block Toeplitz, the block diagonal matrix, Σ_w, consists five equal size blocks with the (i, j)th element of each block equals to $0.7^{|i-j|}$. In term of the sparse precision matrix, we simulated the K-nearest-neighbor networks as describe above. Both the covariance structures were used to mimic the biological gene networks with sparse conditional dependency structure [WT11, XLV16]. We then simulate $x_i \sim N(\mu_k, \Sigma_w)$ for $i \in C_k$.

- Multi-class case: we consider $K = 3$ classes and set $\mu_1 = \mathbf{0}$, $\mu_{2j} = \mathbf{0} \sim N(0.3, 0.5)$ for $j \in \{1, \ldots, 20\}$, $\mu_{3j} = N(-0.5, 0.5)$ for $j \in \{21, \ldots, 40\}$ and $\mu_{kj} = 0$ otherwise. With the same covariance structure as in the binary cases, we simulate the data as $x_i \sim N(\mu_k, \Sigma_w)$ for $i \in C_k$.

In Table 4.6, we reported the prediction accuracy based on 100 replicates.

Table 4.6: Summary statistics reporting performance of the GEV, LDA$-\ell_1$, Direct and Oracle methods. We report the means of the FD with its standard deviation in parentheses. The results are based on 100 replications

	Σ_w	Type	LDA-ℓ_1	Direct	GEV	Orcal
Binary	Toeplitz	Error	75.12 (23.36)	31.96 (16.61)	**30.58 (17.04)**	14.54 (9.77)
	NN	Error	60.12 (8.53)	54.88 (13.73)	**52.70 (11.71)**	23.08 (4.64)
Multi-class	Toeplitz	Error	155.88 (24.26)	70.14 (31.54)	**60.02 (28.30)**	27.34 (14.09)
	NN	Error	103.1 (7.60)	83.3 (7.90)	**40.68 (24.46)**	35.06 (17.04)

4.4 Canonical Correlation Analysis

In this subsection, we assess the performance of GEV under sparse CCA framework by applying it to several simulated datasets. In all settings, we let X and Y have same dimension $d = q$, $\Sigma_x =$

$\Sigma_y = \Sigma$. Following the formulation in [CGRZ13], we model $\mathrm{Cov}(X,Y) = \Sigma_{xy}$ as

$$\Sigma_{xy} = \Sigma_x U \Lambda V' \Sigma_y, \tag{4.2}$$

where $U = (U_1, U_2)$ and $V = (V_1, V_2)$ are $d \times 2$ matrices, and Λ is a 2×2 diagonal matrix with $\lambda_1 = 1$ and $\lambda_2 = 0.7$. We set the nonzero rows of U_1, U_2, V_1 and V_2 at $\{1,2,\ldots,6\}, \{7,\ldots,12\}, \{d-5,\ldots,d\}$ and $\{d-11,\ldots,d-6\}$. The values at the nonzero rows are sampled uniformly from $(-1,-0.5) \cup (0.5,1)$ and then are normalized with respect to Σ such that $U'\Sigma U = I$ and $V'\Sigma V = I$. To capture different dependency structures, we consider the following three settings.

- Identity with $\Sigma = I_p$.
- Toeplitz with $\Sigma = (\sigma_{ij})$ where $\sigma_{ij} = 0.3^{|i-j|}$ for all $i, j \in [d]$.
- Sparse precision matrix with $\Omega = \Sigma^{-1}$ being sparse. Specifically, we generated the sparse precision matrix through nearest-neighbour networks algorithm in [LG06] with number of neighbors, $m = 5$.

We compared the performance of our methods and the Penalized Multivariate Analysis method (PMA) proposed by [WTH09] via examining the Frobenius norm distance (FD) measuring the distance between the true and estimated subspaces. The sparsity tuning parameters in PMA were chosen using permutation as suggested by the R package PMA, while the tuning parameter λ in GEV was selected by cross-validation. Results of the simulations are reported in Table 4.7. Summary statistics are based on 100 replicate trails under each of the six conditions. In general, the GEV method results in smaller Frobenius norm distance for both U and V. Under all settings, the GEV method outperforms the PMA methods yielding greater improvements as the dependence structures become more complicated.

Table 4.7: Summary statistics reporting performance of the GEV and PMA methods. We report the means of the FD with its standard deviation in parentheses. The results are based on 100 replications

(p,q,n)	Σ	$U-$PMA	$V-$PMA	$U-$GEV	$V-$GEV
(50, 50, 300)	Identity	0.589 (0.085)	0.543 (0.126)	**0.515 (0.120)**	**0.496 (0.116)**
	Toeplitz	0.839 (0.089)	0.833 (0.080)	**0.623(0.106)**	**0.607 (0.099)**
	NN	1.365 (0.132)	1.353 (0.141)	**0.774 (0.137)**	**0.778 (0.171)**
(500, 500, 2000)	Identity	0.255 (0.076)	0.399 (0.043)	**0.208 (0.074)**	**0.297 (0.071)**
	Toeplitz	0.696 (0.030)	0.745 (0.023)	**0.426 (0.059)**	**0.424 (0.060)**
	NN	1.031 (0.152)	1.031 (0.076)	**0.433 (0.079)**	**0.420 (0.090)**

4.5 Application to Tumor-Infiltrating Lymphocytes Data

To demonstrate our approach's potential utility, we apply the GEV-SIR algorithm to the Tumor-Infiltrating Lymphocytes (TILs) data inferred from The Cancer Genome Atlas (TCGA) Ovarian serous cystadenocarcinoma (Ovarian Cancer) and Lung Squamous Cell Carcinoma (Lung cancer) using CIBERSORT [NLG$^+$15]. Compelling clinical evidence suggests that the presence of effector immune cells, such as CD8$^+$ T cells and plasma cells, is positively associated with superior survival in patients with ovarian cancer. Notably, an inflamed tumor microenvironment, which is characterized by the infiltration of CD8$^+$ T cells, also attracts plasma cells. A higher percentage of plasma cell infiltration is significantly correlated with the highest levels of CD8$^+$, CD4$^+$, and CD20$^+$ TILs, and superior clinical outcomes in patients with ovarian cancer [SJ15]. Indeed, a pan-cancer analysis also identified plasma cells as a novel prognostic factor for superior survival [WN18]. However, the mechanism of plasma cell homing to the tumor bed remains unclear. Identifying oncogenic signaling pathways that shape the plasticity of plasma cell recruitment and differentiation holds promise to better classify patients based on their immune-editing profiles.

We extracted the expression of 2,000 genes with the largest variance among samples. We first determine the number of dimensions using eigen decomposition. As shown in Figure 1, GEV-SIR favors $d = 1$ because of a large gap between the first and the second largest eigenvalue. The tuning parameter λ is then selected via the cross-validation procedure. Figure 4.2 shows a strong monotonic relationship between the GEV-SIR score and the plasma cell abundances. To validate

the derived GEV-SIR score's predictability, we used the TCGA Lung cancer data as a test set. Specifically, we selected the same 2,000 genes as in Ovarian cancer and projected them into the estimated Ovarian cancer GEV-SIR direction. The right panel in Figure 1 demonstrates a similar monotonically decreasing pattern between the GEV-SIR direction and plasma cell recruitment in the Lung cancers.

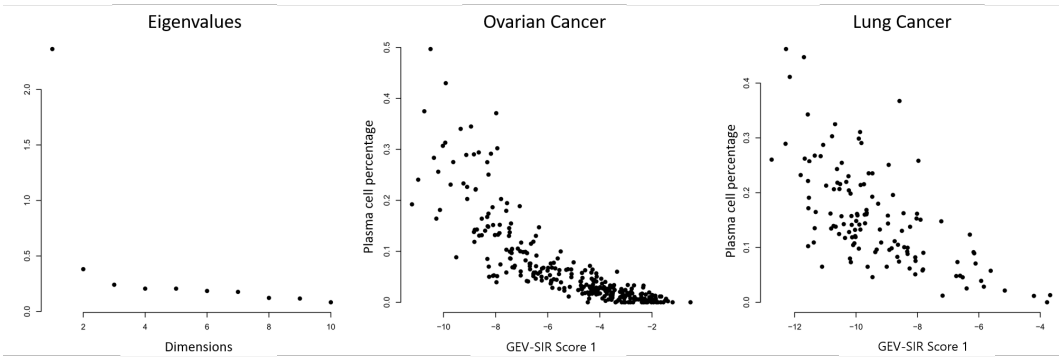

Figure 4.2: Relationship between plasma cells and GEV-SIR direction. The left panel shows the distribution of eigenvalues of $\hat{\Omega}$. The scatter plots in the middle and right panels show the relationship between the tumor infiltrated plasma cell and the GEV-SIR direction.

To better understand the reduced dimensions' biologic significance, we performed a GO pathway enrichment analysis using GSEA [STM$^+$05]. In the reduced dimension, gene clusters which regulate immune cell differentiation (p value < 0.001), effector function such as enzyme activity (p value < 0.001), regulation of apoptosis (p value < 0.05), and chemotaxis signaling (p value < 0.05), are significantly enriched. Among the strongest pathways that are positively associated with plasma cell recruitment in the second GEV-SIR direction are the defense response (p value < 0.001) and type 1 Interferon (IFN-I) network (p value < 0.05). The defense response pathway is informed by immune detection of danger-associated molecular patterns (DAMPs) and pathogen-associated molecular patterns (PAMPs). In the tumor immune detection, cancer cell damage-associated DAMPs, such as DNA, could alert immune cells and promote an "inflamed" tumor microenvironment [GGK10], which is amenable for plasma cell recruitment. IFN-I signatures have been emerging as a central signaling pathway that facilitates anti-tumor immunity [ID14]. IFN-I functions by binding to its receptor on target cells and launch a large transcriptome

consisting of interferon-stimulated genes (ISGs), among which are chemokines, such as CXCL9, CXCL10, and CXCL12. CXCL9 and CXCL10 are essential mediators of effector immune cell chemotaxis [SCR14]. Downregulation of these chemokines severely compromises anti-tumor immunity. Importantly, CXCL12 potently promotes plasma cell recruitment [DPN14].

4.6 Application to Single-Cell RNAseq Data

To demonstrate the ability of GEV-SIR handling noisy data, we next utilize GEV-SIR to analyze a dataset of human embryonic stem cells grown over a 27-day time course from [MvDW$^+$19]. The [MvDW$^+$19] dataset comprises expression measurements of 33694 genes over 31,000 cells through single-cell RNAseq (scRNAseq) technology, where cells were sampled at the following differentiation time intervals: (Day 0-3), (Day 6-9), (Day 12-15), (Day 18-21), and (Day 24-27). Unlike the measurement from bulk tissue as in the tumor-infiltrating lymphocytes data, the scRNAseq data suffers from high noise level, contamination with outliers, and large proportion of missing values due to the limited initial mRNA in each cell. Taking the developmental time as response and gene expression as predictors, we aim to reveal the driving factors/pathways for the differentiating process of embryonic cells. Following the same preprocessing procedure as in [MvDW$^+$19], we applied our GEV-SIR method to the normalized scRNAseq data and showed the two dimension embedding in Figure 4.3. Our analysis successfully captured the smooth transition of the embryonic differentiation process with GEV-SIR1 direction capturing the difference between all developmental interval and GEV-SIR2 direction mainly reflecting the differences between the last two intervals and the first three.

Figure 4.3: GEV-SIR analysis of embryoid body scRNAseq data.

CHAPTER 5

GRAPHICAL NEURAL NETWORKS FOR MULTI-MODAL DATA INTEGRATION

With the emergence of joint platforms for single-cell sequencing, the data produced by methods like scRNA-seq and scATAC-seq can be combined for multi-modality cell sequencing, attributing to each cell mRNA and DNA data. This new collection of data provides unique challenges in how best to incorporate the large amount of multi-modal data for data analysis purposes. Both data streams are very high dimension with sample sizes often on the same order or smaller than the number of features, placing the data analysis problem in the HDLSS scenario. Furthermore, the sequencing data is notoriously sparse, where the vast majority of features may be zero in a typical dataset due to technical error from dropout [SNL$^+$17], which can be even more pronounced in multi-modal data [LHH20]. One main goal for multi-modal data is achieving data integration, which is any method that combines the heterogeneous data better for downstream tasks. One way to achieve this task is by performing a joint embedding of the features of the two modalities in a shared low-dimensional space. Such an embedding ideally captures a meaningful representation of the complex cellular states from different types of measurements.

Deep learning techniques have recently been used to solve the task of multi-modal date integration for single-cell data to great success [GZP21, AAB$^+$20]. However, most of the current fail to take into account high-order interactions among cells or different modalities, and instead treat each cell as a separate input. This higher-order information is essential giving structure to the data that allows for learning a proper low-dimensional representation of high-dimensional and sparse cell features. Graph neural networks (GNNs) [LDJ$^+$21, KW17] give unique tools for capturing the desired higher-order information required for data integration. GNNs aggregate information from neighborhoods to update node embeddings iteratively, which allow for the encoding high-order structural information through multiple aggregation layers. In addition, GNNs smooth the features by aggregating neighbors' embedding, which provides an extra denoising mechanism [MLZ$^+$21]. Hence, by modeling the interactions between cells and their features as a graph, we can adopt

GNNs to exploit the structural information.

We implement a GNN framework for multimodal data integration designed in concert with [WDJ+22] called scJEGNN for single-cell Joint Embedding GNNs. We apply this model to benchmark datasets provided by NeurIPS 2021 [LBC+21] for a multimodal since-cell data integration competition and compare its performance to competitor submissions.

5.1 Problem Statement

The two modalities we operate on are GEX as mRNA data, and ATAC as DNA data. Each modality is represented as a matrix $\mathbf{X}_i \in \mathbb{R}^{n \times d_i}$, $i = 1,2$, where n is the number of cells and d_i denotes the feature dimension for each cell. In our application the GEX has dimension $d_1 = 13,431$ while the ATAC has dimension $d_2 = 116,490$, while the total sample size is $n = 42,492$. The data is also highly sparse; only 9.75% of GEX and 2.9% of ATAC features are nonzero on average. The data has expert annotation giving cell type labels for each cell with a total of 22 different cell type classes, and 2 different real values indicated cell-cycle developmental stages. As well, in our application, two modalities are sequenced with a total of 10 batches, introducing the possibility of large batch effects occurring.

The goal then is to learn an embedded representation of the cells in $\mathbb{R}^{d_3}$ that best leverages the underlying information of the two modalities in order to preserve cell info and remove spurious batch effects on the representations. This evaluation of how well the embedding represents pertinent biological information is calculated by a collection of metrics $\mathcal{M} : \mathbb{R}^{n \times d_3} \to \mathbb{R}^k, \mathcal{M}(\mathbf{X}) = (m_1(\mathbf{X}), \ldots, m_k(\mathbf{X}))$, where k metrics are given by $m_i : \mathbb{R}^{n \times d_3} \to \mathbb{R}$, $i \in [k]$, with higher values indicating better performance of the embedding. The problem can be formally defined as

Given modality $\mathbf{X}_1 \in \mathbb{R}^{n \times d_1}$ *and modality* $\mathbf{X}_2 \in \mathbb{R}^{n \times d_2}$*, learn three mapping functions* $f_{\theta_1}, f_{\theta_2}$ *and* f_{θ_3} *parameterized by* θ_1, θ_2 *and* θ_3 *to learn a representation* $\mathbf{H} \in \mathbb{R}^{n \times d_3}$

$$\mathbf{H} = f_{\theta_3}\left(CONCAT(f_{\theta_1}(\mathbf{X}_1), f_{\theta_2}(\mathbf{X}_2))\right) \tag{5.1}$$

that best maximizes the coordinates of $\mathcal{M}(\mathbf{H})$*. Here* $f_{\theta_1}(\mathbf{X}_1) \in \mathbb{R}^{n \times d_1'}$ *and* $f_{\theta_2}(\mathbf{X}_2) \in \mathbb{R}^{n \times d_2'}$ *correspond to new representations learned from modality* $\mathbf{X}_1$ *and* $\mathbf{X}_2$*, and for CONCAT* : $(\mathbb{R}^{n \times d_1'} \times$

$\mathbb{R}^{n\times d_2'}) \to \mathbb{R}^{n\times(d_1'+d_2')}$ *the function that concatenates the rows of two matrices together.*

For our application, we have $k = 3$ for the number of metrics measuring the performance of the embedding $\mathbf{H}$. The three measurements are are given by a cell-type conservation metric, a cell-cycle conservation metric, and a batch removal metric:

- **NMI cluster/label:** The Normalized mutual information (NMI) [MGH11] compares the overlap of two clusterings. The NMI is applied to the integrated data to compare the cell type labels with an automated clustering (based on Louvain clustering). NMI scores of 0 or 1 correspond to uncorrelated clustering or a perfect match, respectively. Automated Louvain clustering is performed at resolution ranges from 0.1 to 2 in steps of 0.1, and the clustering output with the highest NMI with the label set is used.

- **Cell-cycle conservation:** The cell-cycle conservation score evaluates the amount of variance explained by cell-cycle per batch prior to integration versus the amount of variance after integration. The relative differences of var_{before_i} and var_{after_i} per batch i are aggregated into a final score between 0 and 1, via

$$CC_{conservation} = \frac{1}{B}\sum_{i}^{B}\left(1 - \frac{|\text{var}_{before_i} - \text{var}_{after_i}|}{\text{var}_{before_i}}\right),$$

  where B gives the number of batches. Values near 0 indicate little conservation of variance explained by the cell-cycle, while values near 1 indicate nearly perfect conservation.

- **Batch ASW:** The Average Silhouette Distance (ASW) is used to quantify batch mixing by taking into account the incompatibility of batch labels per cell type cluster. The Batch ASW considers the absolute silhouette width, on batch labels per cell. Here, 0 indicates that batches are thoroughly mixed, but any variation from 0 indicates the presence of a batch effect. The metric re-scales this score so that higher values imply better batch mixing and uses the equation below to determine the per-cell type label, j:

$$\text{batch}ASW_j = \frac{1}{|\mathcal{C}_j|}\sum_{i\in\mathcal{C}_j} 1 - |s(i)|$$

where $\mathcal{C}_j$ is the set of cells with the cell label j and $|\mathcal{C}_j|$ denotes the number of cells in that set. To obtain the final batchASW score, the label-specific batchASW_j scores are averaged:

$$\text{batch}ASW = \frac{1}{C}\sum_{j}^{C}\text{batch}ASW_j$$

where C is the number of unique cell labels. A batchASW value of 1 indicates optimal batch mixing, and a value of 0 indicates fully separated batch clusters.

5.2 Method

In this section, we introduce the scJEGNN framework for multimodal data integration. An illustration of the framework is shown in Figure [ref fig]. Specifically, our framework can be divided into four stages: data preprocessing, graph construction, cell-feature graph convolution, and an autoencoder architecture for the final embedding.

5.2.1 Data Preprocessing

Both modalities, $\mathbf{X}_1$ for GEX and $\mathbf{X}_2$ for ATAC, go through some standard preprocessing steps regularly done in single-cell sequencing tasks. The below sequence of operations describe both f_{θ_1} and f_{θ_2}. First the matrices are ℓ_1-normalized, meaning that each row vector (cell) is divided by the total sum of the absolute values of all its features, normalizing the weight of each cell's total gene expression output. Then the data is log-transformed, so that each normalized value $\bar{\mathbf{X}}_{ij}$ is updated to the value

$$\log(\bar{\mathbf{X}}_{ij} * 10^4 + 1).$$

These values are then divided by the standard deviation of each column, which normalizes the variation of each feature. Lastly both modalities go through an initial dimension reduction using the Latent Semantic Indexing (LSI), which is a type of transformation based on the SVD decomposition. For some choice of k, the transformation simply chooses the top k left singular vectors, and projects the data to dimension k where each coordinate is the inner product with the k vectors. Here we choose different values for k giving d_1' and d_2' for $\mathbf{X}_1$ and $\mathbf{X}_2$. In our experiment, we found

choosing $d_1' = 100$ for GEX and $d_2' = 65$ for ATAC gave the best performance. Then the data is concatenated giving an output $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times (d_1' + d_2')}$. We simplify notation and denote $d' = d_1' + d_2'$ as the combined feature dimension.

5.2.2 Graph Construction

Given the preprocessed $\widehat{\boldsymbol{X}}$, we construct a graph that the GNN can be applied to. We construct a cell-feature bipartite graph, depicted in Figure 5.1, where the cells and their biological features are treated as different nodes, giving us a collection of cell nodes and a collection of feature nodes. The edges are designed to be strictly between the two collections, so that an edge connecting a cell node i to a feature node j directly represents the value of the cell's feature given by $\widehat{\boldsymbol{X}}_{ij}$. This requires a weighted edge graph instead of the usual 0-1 adjacency matrices. As we will see, given a proper choice of node embedding values, this graph will be able to propagate information from cells to pertinent feature nodes, and likewise feature nodes can also propagate their information to the cell nodes that express them highly.

We denote the bipartite graph as $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$. In this graph $\mathcal{U}$ is the set of nodes representing the n cells $\{u_1, \ldots, u_n\}$ and cV is the set of nodes representing the d' features $\{v_1, \ldots v_{d'}\}$ with one node for each feature dimension of the input data. The set $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{V}$ gives the edges in the graph between the nodes $\mathcal{U}$ and $\mathcal{V}$ which describe the relations between the cells and the features. The graph can be denoted by the weighted adjacency matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \widehat{\mathbf{X}} \\ \widehat{\mathbf{X}}^\top & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+d') \times (n+d')}$$

where $\mathbf{0}$ is a matrix with all zeros, and $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times d'}$ is the input feature matrix of cells. $\mathbf{A}$ is designed to give the structure of a bipartite graph since nodes of the cell or feature sets only have possible edges between nodes of the other set, not within the same set. The initial embeddings of the feature and cell nodes are given by matrices $\mathbf{V}$ and $\mathbf{U}$ respectively, where each row gives the vector of one node embedding. The feature nodes $\{v_1, \ldots, v_d\} \in \mathbb{R}^d$ are initialized as one-hot vectors, so that each v_i has a one in index i and zeros elsewhere, making $\mathbf{V} \in \mathbb{R}^{d \times d}$ an identity matrix. The

cell nodes are initialized as zero vectors in the same dimension, so that $\mathbf{U} \in \mathbb{R}^{n \times d}$ has all zero entries. The lack of prior information on the cells leads to this uniform initialization for those nodes, and the one-hot embedding works well with the chosen graph convolution to recreate the gene expression and propagate it into the cell node embeddings. Then our cell-feature graph can be denoted $\mathcal{G} = (A, \mathbf{V}, \mathbf{U})$.

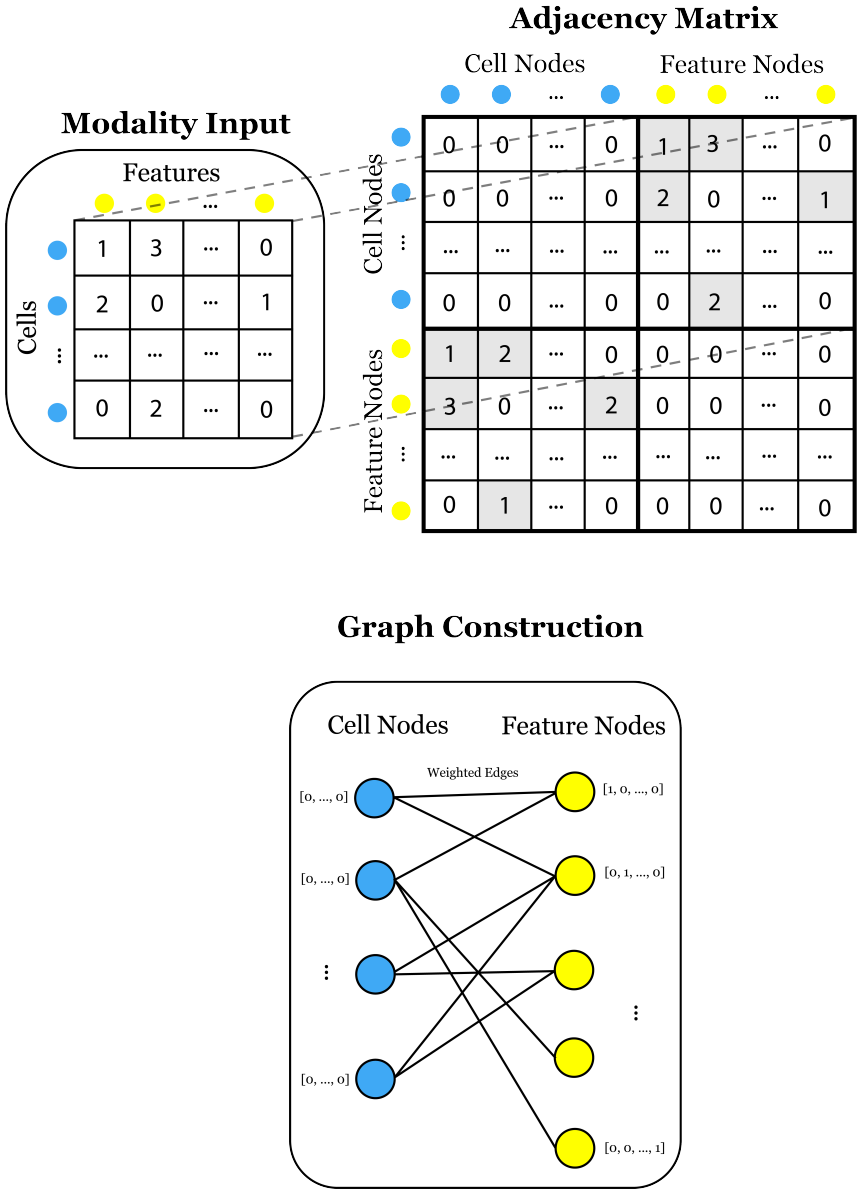

Figure 5.1: scJEGNN graph construction process. The input data determines the value of the weighted edges between the cell nodes and feature nodes, values of zero indicate no edge.

5.2.3 Graph Convolution

Given the constructed cell-feature graph, we wish to choose a graph convolution that captures higher-order structural information from the links between nodes to create better cell node representations. In each layer of a GNN, the embedding of a node is updated according to the propagated value of its neighbor, given by the edge weight times the neighbor node. In the field of GNNs we call this type of information propagation "message passing" [GSR+17] between neighbors. While a the two different node types could yield different message passing methods for each type, we use the same updates for both. To illustrate our method, we give notation of the updates applied to nodes in $\mathcal{U}$, and the updates for nodes in $\mathcal{V}$ are completely analogous. Let $\mathbf{H}^l = \{h_1^l, \ldots, h_n^l\}$, $h_i^l \in \mathbb{R}^{d'}$ be the input node embeddings in the l^{th} layer, where h_i^l corresponds to node u_i. Then the output embedding to the l^{th} layer can be expressed as

$$h_i^{l+1} = \text{Update}(h_i^l, \text{Agg}(h_j^l | j \in \mathcal{N}_i)),$$

where $\mathcal{N}_i$ is the set of first-order neighbors of node u_i, $\text{Agg}(\cdot)$ indicates an aggregation function on neighbor nodes' embeddings, and $\text{Update}(\cdot)$ is an update function the generates a new node embedding from the previous one and the aggregation output.

While there are many choices for both aggregation and update functions in GNNs, we choose the common and simple Graph Convolution Network (GCN) [KW17] model for these layer updates. In general the GCN creates a message $m^{i,l}$ for node i at layer l as follows:

$$m^{i,l} = \sigma\left(b^l + \sum_{j \in \mathcal{N}_i} \frac{e_{ji}}{c_{ji}} h_j^l W^l\right)$$

where j varies over neighbors of u_i in $\mathcal{V}$, e_{ji} denotes the edge weight between u_i and v_j, W^l and b^l are trainable parameters, $\sigma(\cdot)$ is an activation function, and c_{ji} is a normalization term defined as

$$c_{ji} = \sqrt{\sum_{k \in \mathcal{N}_j} e_{jk}} \sqrt{\sum_{k \in \mathcal{N}_i} e_{ki}}.$$

After generating the messages from neighborhoods we update the embedding for nodes in $\mathcal{U}$ as

$$h_i^{l+1} = h_i^l + m^{i,l}.$$

This simple residual mechanism adds the previous layer to the newly updated embedding, which enhances self information, by combining the node embedding with its aggregated neighborhood information.

We choose to decouple the propagation and transformation of the node embeddings. This means that we set $W^l = \mathrm{Id}_{d'}$ as the identity matrix and $b^l = \mathbf{0}$ for all layers l. As well the activation function σ is set to the identity. The choice to remove the learnable parameters and activation function may seem like a big limitation on the transformation, but recent work [WSZ$^+$19, HDW$^+$20] has shown that if later transformations occur (as in our model), the the performance of the model is often improved from the use of simplified GCN layers, and the computation efficiency is greatly increased. [HDW$^+$20] in particular found that if later transformations occur after simplified GCN layers, they are able to produce representations with the same level or better performance than they would with the earlier trainable parameters. This choice also means that the hidden layers keep same dimension throughout, so the end output yields $\mathbf{H}_{\mathcal{U}}^L \in \mathbb{R}^{n \times d'}$ where $\mathbf{H}_{\mathcal{U}}^L$ is the hidden layer representation of the cell nodes in the last layer L. In our application we found $L = 3$ to have the best performance. We take advantage of this consistent hidden layer size by completing the GNN output with a weighted summation of all the hidden layers, giving

$$\widehat{\mathbf{H}} = \sum_{i=1}^{L} w_i \cdot \mathbf{H}_{\mathcal{U}}^i.$$

Our GCN model is seen in Figure 5.2, showing the summation of the GCN layers $\widehat{H}$ being used as input for the final autoencoder layer.

It is worth noting that the simplified GCN computation and choice of node embeddings leads to a recreation of each cell's original representation for the first and second update. If a cell i has features given by the i^{th} row of $\widehat{\mathbf{X}}$, then the cell node update would equal $u_i = \sum_{k=1}^{d} \widehat{\mathbf{X}}_{i,k} \mathbf{e}_k$, where $\mathbf{e}_k$ the k^{th} unit vector with 1 in coordinate k and 0's elsewhere. This value is also the output of the second layer for the cell nodes due to the lack of an update for the feature nodes on the first layer from the all zero values of the initial cell node embeddings. After the second layer the cell nodes update in a novel manner, weighing messages higher from features that are expressed at a

greater value in that cell. This leads to increasingly similar cell embeddings from cells that have high coexpression of features.

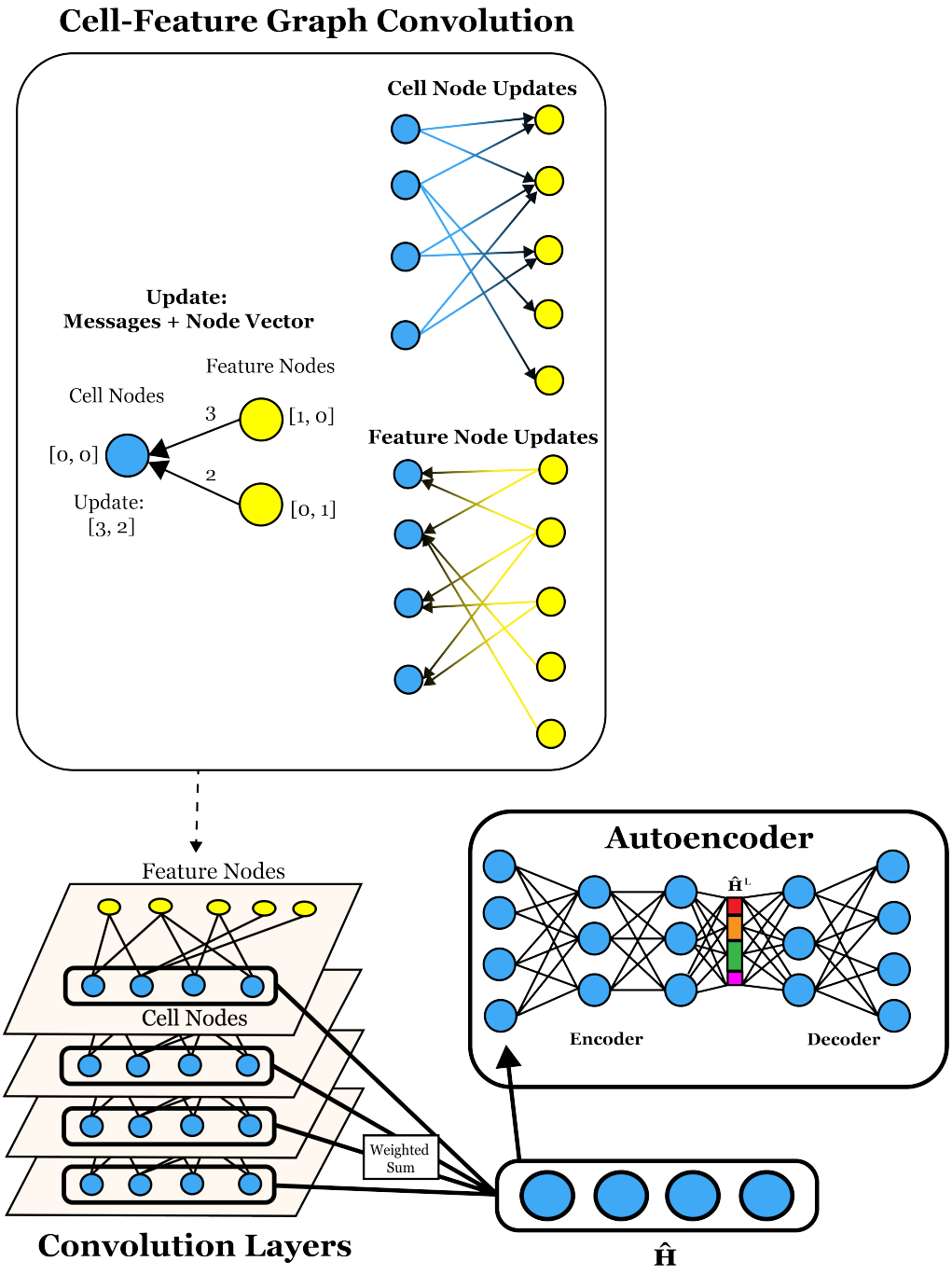

Figure 5.2: scJEGNN graph convolution. Multiple convolution layers propagate information from the weighted edges to update cell and feature nodes.

5.2.4 Autoencoder

In order to train the final cell embeddings, we use an autoencoder model, presented in Figure 5.3 to achieve desired joint embedding of the data. The autoencoder consists of an encoder layer E and decoder layer D, both modeled as fully connected perceptions. The encoding layer E takes in $\widehat{H}$ and each layer except the last is computed as

$$\widehat{\mathbf{H}}^{l+1} = \mathrm{DO}(\mathrm{BN}(\sigma(\widehat{\mathbf{H}}^{l}W^{l})), p),$$

where $\widehat{\mathbf{H}}^{l}$ is the output of the l^{th} layer, $W^{l}\mathbb{R}^{d_l \times d_{l+1}}$ is a trainable linear transformation, σ is an activation function, BN is the batch normalization function [IS15], and DO is a dropout function with parameter $0 < p < 1$. Batch normalization operates by normalizing the empirical mean and variance of each batch. The use of the batch normalization function is a well-established technique in deep learning to improve training. The dropout function randomly zeroes some of the elements of the input matrix with probability p using samples from a Bernoulli distribution, and also improves training by forcing information redundancy in the connections between layers. The last layer removes the batch normalization and dropout leaving

$$\widehat{\mathbf{H}}^{L} = \sigma(\widehat{\mathbf{H}}^{L-1}W^{L}).$$

We choose $L = 4$, $p = .2$, and reduce the dimension of the input iteratively with $d_1 = 150$, $d_2 = 120$, $d_3 = 100$, to the final embedding dimension of 39. For all layers σ is chosen to be the ReLU function, $\mathrm{ReLU}(x) = \max(0, x)$. The decoder layer is a simple two-layer transformation of the joint embedding back to the original dimension d', giving

$$D(\widehat{\mathbf{H}}^{L}) = \sigma(\sigma(\widehat{\mathbf{H}}^{L}W^{1})W^{2})$$

where $\sigma = \mathrm{ReLU}$ and $W^{1} \in \mathbb{R}^{L \times d_1}$ and $W^{2} \in \mathbb{R}^{d_1 \times d'}$.

The goal of the autoencoder is take in the GNN output $\widehat{H}$, and learn a low-dimensional representation that properly captures the biological information we care about. In order to due so the model is trained via latent feature regularization, which forces chosen latent features to predict for

cell type, cell-cycle phase score, and to blur the batch features in order to remove spurious batch effects. We combine these with the usual reconstruction loss that is used to train autoencoders to produce salient features in the encoder representation. Thus the autoencoder is trained with both supervised and self-supervised losses: three supervised losses are applied to the output of the encoder that gives the final joint embedding, and one self-supervised loss is applied to the output of the decoder. The hidden layer size of $\widehat{\mathbf{H}}^L$ is specifically designed to accommodate enough features for each of these supervised losses. We allocate 22 features for the number of cell types, 10 features for the number of batches, 2 features for the cell-cycle score, and 5 extra features to allow for additional pertinent information to improve the reconstruction loss, giving us the total of 39. The losses are then summed to give us a total loss $\mathcal{L}$. In detail we have

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{cell type}} + \mathcal{L}_{\text{batch}} + \mathcal{L}_{\text{cell-cycle}}.$$

The reconstruction loss is simply mean squared error giving

$$\mathcal{L}_{\text{recon}}(D(E(\widehat{H})) = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\mathbf{X}} - D(E(\widehat{H}))\right)^2.$$

The cell type loss is a cross-entropy loss function on the classification task; if $\widehat{H}^L_{i,1:J}$ gives us weights corresponding to the J cell classes for input i, and $C_i \in \{1,\ldots,J\}$ denotes class of cell i, then the loss is

$$\mathcal{L}_{\text{cell type}}(\widehat{H}) = \sum_{i=1}^{n}\sum_{j=1}^{J}\mathbf{1}_{C_i=c_j}\log(\hat{Y}_j)$$

where $\mathbf{1}_{C_i=c_j}$ is equal to 1 if $C_i = c_j$ and 0 otherwise, and

$$\hat{Y}_j = \frac{e^{\widehat{H}^L_{i,j}}}{\sum_{k=1}^{J} e^{\widehat{H}^L_{i,j}}}$$

is the softmax function. The loss function cL_{batch} for the batch effect is also a cross-entropy loss, but it attempts to remove batch effects by training the classifier to learn random batch labels. To do so we use a uniform distribution to generate batch labels for the input, and use the same function above on the 10 batch features in $\widehat{H}^L$. The last function $\mathcal{L}_{\text{cell-cycle}}$ is a simple mean squared error loss on each cells cell-cycle phase score compared to the 2 allocated features in $\widehat{H}^L$ that are supposed to capture the values.

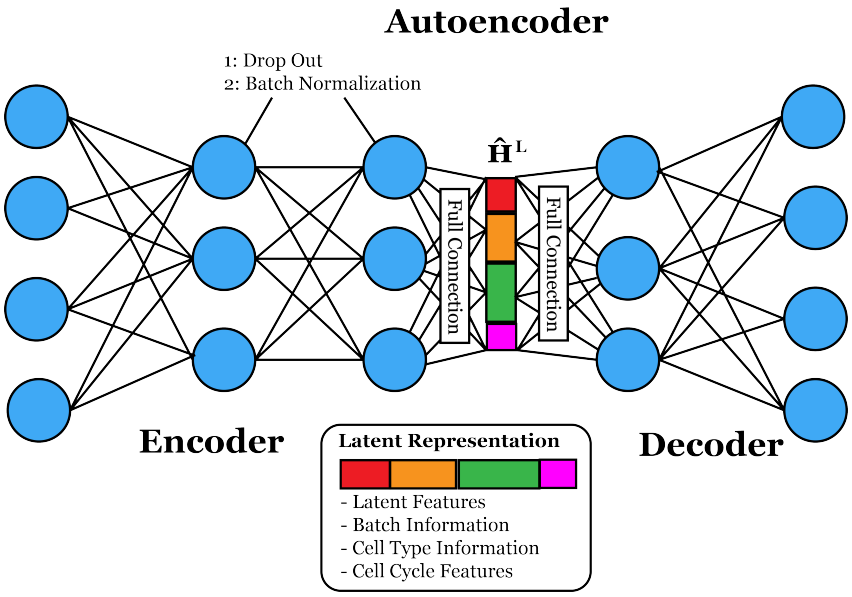

Figure 5.3: scJEGNN Autoencoder architecture. Each layer is fully connected, and the encoder layers feature drop out and batch normalization steps.

5.3 Experimental Results

Table 5.1: Performances for Joint Embedding Task

Model	NMI Cluster/label	Cell-Cycle Conservation	Batch ASW	Average Metric
Baseline	.6502	.8259	.7178	.7313
GLUE	.7754	.8355	.9100	.8403
Amateur (JAE)	0.7723	.9195	.8898	.8610
scJEGNN	**.8057**	**.9204**	**.9112**	**.8791**

We demonstrate the effectiveness of our framework scJEGNN in the joint embedding task for GEX and ATAC, and show that the model outperforms the competitor submissions to the competition on the three evaluation metrics. Of the 25 teams that submitted models to be evaluated to the NeurIPS 2021 Joint Embedding task competition, we choose the top two performing teams and show their performance along side our own. Team Amateur submitted JAE, an autoencoder that we designed our own autoencoder from, with the same latent feature regularization but additional residual connections and layers, and no graphical component. GLUE was an autoencoder model as well guided by an external knowledge graph. We additionally provide a baseline model provided

by simply evaluating a concatenation of the two modalities after dimension reduction by principle component analysis (PCA). In Table 5.1 we can see that our model significantly outperforms the other models, with an improvement over 0.1 according to the average metric.

CHAPTER 6

CONCLUSION

In this work we have developed two methods for finding low-dimensional representations of high-dimension data. The first is given by a unified framework for generalized eigenvalue problems in the GEV estimator. This sparse projection regression framework is a reformulation of an intractable Rayleigh quotient problem and achieves great computational efficiency. We established nonasymptotic error bounds on the proposed estimators for the applications of SIR and LDA, and showed these rates are minimax optimal. We showed application of GEV to the CCA problem, and adapted the algorithm for a robust Huber-loss based formulation. We tested our framework on both synthetic and real datasets and demonstrated the algorithm's superior performance compared with other state-of-the-art methods in high dimensional data. The second method is the scJEGNN, a graphical neural network tailored to data integration for HDLSS single-cell sequencing data. We showed that with the unique model, the GNN is able to leverage structural information of the biological data relations in order to perform a joint embedding of multiple modalities of single-cell gene expression data.

6.1 Future Work

GEV. One obvious goal is to show the same statistical consistency the GEV estimator obtains for SIR and LDA is also true for the application of CCA. The adaptation of the GEV estimator to CCA required changing the structure of $\boldsymbol{U}$ from a collection of products of eigenvalues and eigenvectors of $\boldsymbol{\Omega} = \text{cov}(\mathbb{E}[\mathbf{x}|y])$, to the combination of the right and left singular vectors of $\boldsymbol{\Sigma}_{xy} = \mathbf{O}_x \mathbf{D} \mathbf{O}_y^\top$. Setting $\mathbf{U} = (\mathbf{O}_x^\top, \mathbf{O}_y^\top)^T/\sqrt{2}$ requires additional work to show that the difference $\|\mathbf{U} - \widehat{\mathbf{U}}\|_{\infty,\infty}$ is bound above by the desired rate of $C\sqrt{\frac{\log(d)}{n}}$. Similarly, the robust version of the GEV estimator using the Huber loss also has the possibility of proving strong theoretical rates of convergence. This work requires a new derivation of the bound on $\|\nabla \mathcal{L}(\mathbf{W}^*)\|_{\infty,\infty} \leq \lambda/2$ due to the change in the gradient of the loss. The derivative of the Huber norm is more complicated than the Frobenius norm

and leads to a piecewise defined function with additional multiplications of Σ. Lastly an extension of the GEV estimator to a nonlinear dimension reduction technique is also likely possible. An arbitrary manifold can be approximated locally by linear spaces which can be estimated using K-nearest neighbors from the sample data. Given these connected affine spaces, we can apply GEV to each to get a collection of projections, which we can carefully combine to project the data to a lower dimensional space. This likely requires much more stringent requirements about the sample size in each portion of the linear approximation.

GNNs for single-cell tasks. The cell-gene graph of the scJEGNN has the means to be used in a number of tasks in singe-cell data analysis. The representation gained for the cells after going through multiple convolution layers may lead to much better estimates for methods like K-nearest neighbors, which is relied on in many methods that perform imputation. If naively applied, the KNN estimate on the initial highly sparse data is likely to be unreliable, and if the estimate could be improved by applying to the updated cell representations, the downstream steps taken for imputation could be drastically improved. In addition, these representations could be used for other common node-based tasks like classification and clustering, which lead to cell annotation methods and biological clustering in the single-cell world, or for graph-based tasks, which would lead to methods for disease prediction given cell populations from distinct patients. The methods using this graphical model can be further enhanced with some key alterations to the graph structure. The bipartite graph structure can be extended to a fuller graph that has edges between the gene nodes and edges between cell nodes. Gene node edges are important to represent gene pathways, which indicate any number of gene causal relations including gene co-expression or regulatory networks. In spatial transcriptomics data [Rus16], single-cell sequencing data is given new geometric context so that each small cluster of cells is placed in a 2 or 3-dimensional grid. This spatial data can be included in the cell-gene graph with additional edges between cells (or group of cells) indicating adjacency relations. These collection of methods utilizing this GNN model seem promising for the variety of applications listed, and are being actively developed together into a full GNN-based

package for single cell data analysis methods.

BIBLIOGRAPHY

BIBLIOGRAPHY

[AAB+20] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John Marioni, and Oliver Stegle. Mofa+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome biology, 21(1):1–17, 2020.

[AHB+16] Benedict Anchang, Tom Hart, Sean Bendall, Peng Qiu, Zach Bjornson, Michael Linderman, Garry Nolan, and Sylvia Plevritis. Visualization and cellular hierarchy inference of single-cell data using spade. Nature Protocols, 11(7):1264–1279, 2016.

[AHMM18] Lun Aaron, Laleh Haghverdi, Michael Morgan, and John Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. Nature Biotechnology, 36(5):421–427, 2018.

[APYG19] Cédric Arisdakessian, Olivier Poirion, Xun Yunits, Breck Zhu, and Lana Garmire. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. Genome Biology, 20:1–14, 2019.

[BHS+18] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nature Biotechnology, 36(5):411–420, 2018.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, pages 1–122, 2011.

[BRT09] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. The Annals of Statistics, 37:1705–1732, 2009.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1):183–202, 2009.

[CGRZ13] Mengjie Chen, Chao Gao, Zhao Ren, and Harrison H. Zhou. Sparse CCA via Precision Adjusted Iterative Thresholding. arXiv:1311.6186 [math, stat], November 2013. arXiv: 1311.6186.

[CHWE11] Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll. Sparse discriminant analysis. Technometrics, 53(4):406–413, 2011.

[CL98] Chun-Houh Chen and Ker-Chau Li. Can SIR be as popular as multiple linear regression? Statistica Sinica, 8:289–316, 1998.

[CMW13] Tianwen Tony Cai, Zongming Ma, and Yihong Wu. Sparse PCA: Optimal rates and

adaptive estimation. The Annals of Statistics, 41(6):3074–3110, 2013.

[Con18] Tabula Muris Consortium. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature, 562(7727):367–372, 2018.

[Coo98] Ralph Dennis Cook. Regression graphics. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, 1998.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision, (40):120–145, 2011.

[DPN14] Yvonne Döring, Christian Pawig, Lukas Weber, and Heidi Noels. The cxcl12/cxcr4 chemokine ligand/receptor axis in cardiovascular disease. Frontiers in Physiology, 5, 2014.

[ENK16] Anders Eklund, Thomas Nichols, and Hans Knutsson. Cluster failure: Why fmri inferences for spatial extent have inflated false-positive rates. Proceedings of the National Academy of Sciences, (113):7900–7905, 2016.

[FFT12] Jianqing Fan, Yang Feng, and Xin Tong. A road to classification in high dimensional space: the regularized optimal affine discriminant. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(4):745–771, September 2012.

[FS81] Jerome Friedman and Werner Stuetzle. Projection pursuit regression. Journal of the American Statistical Association, 76(376):817–823, 1981.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GGK10] Sergei Grivennikov, Florian Greten, and Michael Karin. Immunity, inflammation, and cancer. Cell, 140(6):883–899, 2010.

[GSR+17] Justin Gilmer, Samuel Schoenholz, Patrick Riley, Oriol Vinyals, and George Dahl. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, ICML, 2017.

[GZP21] Boying Gong, Yun Zhou, and Elizabeth Purdom. Cobolt: Joint analysis of multimodal single-cell sequencing data. bioRxiv, 2021.

[HBR+17] Adam Haber, Moshe Biton, Noga Rogel, Rebecca H Herbst, Karthik Shekhar, Christopher Smillie, Grace Burgin, Toni M Delorey, Michael R Howitt, Yarden Katz, Itay Tirosh, Semir Beyaz, Danielle Dionne, Mei Zhang, Raktima Raychowdhury, Wendy Garrett, Orit Rozenblatt-Rosen, Hai Ning Shi, Omer Yilmaz, Ramnik J Xavier, and Aviv Regev. A single-cell survey of the small intestinal epithelium.

Nature, 551(7680):333–339, 2017.

[HDW+20] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation, page 639–648. Association for Computing Machinery, New York, NY, USA, 2020.

[HMN05] Peter Hall, James Stephen Marron, and Amnon Neeman. Geometric representation of high dimension, low sample size data. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(3):427–444, 2005.

[Hot33] Harold Hotelling. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24(6):417, 1933.

[HST11] David Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. Machine Learning, 83(3):331–353, 2011.

[Hub73] Peter Huber. Robust Regression: Asymptotics, Conjectures and Monte Carlo. The Annals of Statistics, 1(5):799–821, 1973.

[ID14] Lionel Ivashkiv and Laura Donlin. Regulation of type i interferon responses. Nature Reviews Immunology, 14(1):36–49, 2014.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, page 448–456. JMLR.org, 2015.

[KW17] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In ICLR, 2017.

[LBC+21] Malte Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann Chen, Louise Deconinck, Angela Detweiler, Alejandro Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In NeurIPS Datasets and Benchmarks Track (Round 2), 2021.

[LDJ+21] Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. Graph neural networks with adaptive residual. Advances in Neural Information Processing Systems, 34, 2021.

[Len08] Chenlei Leng. Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. Computational Biology and Chemistry, 32(6):417–425, 2008.

[LG06] Hongzhe Li and Jiang Gui. Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. Biostatistics (Oxford, England), 7(2):302–317, April 2006.

[LHH20] Jeongwoo Lee, Do Young Hyeon, and Daehee Hwang. Single-cell multiomics: technologies and data analysis methods. Experimental Molecular Medicine, 52(9):1428—-1442, 2020.

[Li91] Ker-Chau Li. Sliced inverse regression for dimension reduction. Journal of the American Statistical Association, 86(414):316–327, 1991.

[Li07] Lexin Li. Sparse sufficeint dimension reduction. Biometrika, 94(3):603–613, 2007.

[LN06] Lexin Li and Christopher Nachtsheim. Sparse sliced inverse regression. Technometrics, 48(4):503–510, 2006.

[LSB+18] Jacob Levine, Erin Simonds, Sean Bendall, Kara Davis, El ad Amir, Michelle D Tadmor, Oren Litvin, Harris Fienberg, Astraea Jager, Eli Zunder, Rachel Finck, Amanda Gedman, Ina Radtke, James R Downing, Dana Pe'er, and Garry Nolan. Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. Cell, 162(1):184–197, 2018.

[LWL+20] Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. Nature Communications, 11(1), 2020.

[LYS18] Hanna Levitin, Jinzhou Yuan, and Peter Sims. Single-cell transcriptomic analysis of tumor heterogeneity. Trends Cancer, 4(4):264–268, 2018.

[LZL18] Qian Lin, Zhigen Zhao, and Jun Liu. On consistency and sparsity for sliced inverse regression in high dimensions. The Annals of Statistics, 46(2):482–1492, 2018.

[LZL19] Qian Lin, Zhigen Zhao, and Jun S. Liu. Sparse sliced inverse regression via lasso. Journal of the American Statistical Association, 114:1726–1739, 2019.

[MGH11] Aaron McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. arXiv preprint arXiv:1110.2515, 2011.

[MLZ+21] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. A unified view on graph neural networks as graph signal denoising. In Proceedings of the 30th ACM International Conference on Information Knowledge Management, pages 1202–1211, 2021.

[MP43] W.S. McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5:115–133, 1943.

[MT21] Yao Ma and Jiliang Tang. Deep Learning on Graphs. Cambridge University Press, 2021.

[MvDW+19] Kevin Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel Burkhardt, William Chen, Kristina Yim, Antonia van den Elzen, Matthew Hirn, Ronald Coifman, Natalia Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. Nature Biotechnology, 37(12):1482–1492, 2019.

[MZY12] Qing Mai, Hui Zou, and Ming Yuan. A direct approach to sparse discriminant analysis in ultra-high dimensions. Biometrika, 99(1):29–42, March 2012.

[Nes83] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. Proceedings of the USSR Academy of Sciences, 269:543–547, 1983.

[Nes04] Yu. Nesterov. Introductory Lectures on Convex Optimization. A Basic Course. 2004.

[NHS+16] Sonia Nestorowa, Fiona Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola Wilson, David Kent, and Berthold Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. Blood, 128(8):20–31, 2016.

[NLG+15] Aaron Newman, Chih Long Liu, Michael Green, Andrew Gentles, Weiguo Feng, Yue Xu, Chuong Hoang, Maximilian Diehn, and Ash Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. Nature Methods, (5):453–457, 2015.

[PTB09] Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. Statistical Applications in Genetics and Molecular Biology, 8(1):1–34, 2009.

[Qui20] Peng Qui. Embracing the dropouts in single-cell rna-seq analysis. Nature Communications, 11(1), 2020.

[Roc70] Ralph Tyrell Rockafellar. Convex analysis. Princeton University Press, Princeton, 1970.

[Ros58] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958.

[Rus16] Nicole Rusk. Spatial transcriptomics. Nature Methods, 13(710), 2016.

[RWM+18] Bushra Raj, Daniel E Wagner, Aaron McKenna, Shristi Pandey, Allon M Klein, Jay Shendure, James A Gagnon, and Alexander Schier. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. Nature Biotechnology, 36(5):442–450, 2018.

[RWRY11] Pradeep Ravikumar, Martin Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. Electronic Journal of Statistics, 5:935–980, 2011.

[RZL+21] Jiahua Rao, Xiang Zhou, Yutong Lu, Huiying Zhao, and Yuedong Yang. Imputing single-cell rna-seq data by combining graph convolution and autoencoder neural networks. iScience, 24(5):102393, 2021.

[SCR14] William Schneider, Meike Chevillotte, and Charles Rice. Interferon-stimulated genes: a complex web of host defenses. Annual Review of Immunology, 32:513–545, 2014.

[SDB+18] William Stephenson, Laura Donlin, Andrew Butler, Cristina Rozo, Bernadette Bracken, Ali Rashidfarrokhi, Susan Goodman, Lionel Ivashkiv, Vivian Bykerk, Dana Orange, Robert Darnell, Harold Swerdlow, and Rahul Satija. Single-cell rna-seq of rheumatoid arthritis synovial tissue using low-cost microfluidic instrumentation. Nature Communications, 9(1):791, 2018.

[SJ15] Phillip Santoiemma and Daniel Powell Jr. Tumor infiltrating lymphocytes in ovarian cancer. Cancer Biology Therapy, 16(6):807–820, 2015.

[SNL+17] Valentine Svensson, Kedar Natarajan, Lam-Ha Ly, Ricardo Miragaia, Charlotte Labalette, Iain Macaulay, Ana Cvejic, and Sarah Teichmann. Power analysis of single-cell rna-sequencing experiments. Nature Methods, 14:381–387, 2017.

[SSZ+17] Uri Shaham, Kelly Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. Removal of batch effects using distribution-matching residual networks. Bioinformatics, 33(16):2539—-2546, 2017.

[SSZM16] Dan Shen, Haipeng Shen, Hongtu Zhu, and JS Marron. The statistics and mathematics of high dimension low sample size asymptotics. Statistica Sinica, 26(4):1747, 2016.

[STM+05] Aravind Subramanian, Pablo Tamayo, Vamsi Mootha, Sayan Mukherjee, Benjamin Ebert, Michael Gillette, Amanda Paulovich, Scott Pomeroy, Todd Golub, Eric Lander, and Jill Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceeds of the National Academy of Sciences of the United States of America, 102(43):15545–15550, 2005.

[SYG+05] Franco Scarselli, Sweah Liang Yong, Marco Gori, Markus Hagenbuchner,

Ah Chung Tsoi, and Marco Maggini. Graph neural networks for ranking web pages. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pages 666–672, 2005.

[TBH+17] Po-Yuan Tung, John Blischak, Chiaowen Joyce Hsiao, David Knowles, Jonathan Burnett, Jonathan Pritchard, and Yoav Gilad. Batch effects and the effective design of single-cell gene expression studies. Scientific reports, 7, 2017.

[THNC03] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. Statistical Science, 18(1):104–117, 2003.

[TSY20] Kai Tan, Lei Shi, and Zhou Yu. Sparse sir: Optimal rates and adaptive estimation. The Annals of Statistics, 48(1):64–85, 2020.

[vDSN+18] David van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose Carr, Cassandra Burdziak, Kevin Moon, Christine L Chaffer, Diwakar Pattabiraman, Brian Bierie, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe'er. Recovering gene interactions from single-cell data using data diffusion. Cell, 174(3):716–729, 2018.

[WAH+19] Jingshu Wang, Divyansh Agarwal, Mo Huang, Gang Hu, Zilu Zhou, Chengzhong Ye, and Nancy Zhang. Data denoising with transfer learning in single-cell transcriptomics. Nature Methods, 16:875—-878, 2019.

[Wai19] Martin Wainright. High-Dimensional Statistics: A Non-Asymptotic Viewpoint. Cambridge University Press, 2019.

[WCZZ18] Tao Wang, Mengjie Chen, Hongyu Zhao, and Lixing Zhu. Estimating a sparse reduction for general regression in high dimensions. Statistics and Computing, 28:33–46, 2018.

[WDJ+22] Hongzhi Wen, Jiayuan Ding, Wei Jin, Xie Yuying, and Jiliang Tang. Graph neural networks for multimodal single-cell data integration, 2022.

[WKI08] Ami Wiesel, Mark Kliger, and Alfred Hero III. A greedy approach to sparse canonical correlation analysis. arXiv preprint arXiv:0801.2748, 2008.

[WMC+21] Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu. scGNN is a novel graph neural network framework for single-cell RNA-seq analyses. Nature Communications, 12(1882), 2021.

[WN18] Maartje Wouters and Brad Nelson. Prognostic significance of tumor-infiltrating b cells and plasma cells in human cancer. Clinical Cancer Research, 24(24):6125–

6135, 2018.

[WPL15] Lan Wang, Bo Peng, and Runze Li. A high-dimensional nonparametric multivariate test for mean vector. Journal of the American Statistical Association, (110):1658–1669, 2015.

[WSZ+19] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Proceedings of the 36th International Conference on Machine Learning, pages 6861–6871, 2019.

[WT11] D M Witten and R Tibshirani. Penalized classification using Fisher's linear discriminant. Journal of Royal Statistical Society, Series B, 73:753–772, 2011.

[WTH09] Daniela Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics (Oxford, England), 10(3):515–534, 2009.

[XLV16] Yuying Xie, Yufeng Liu, and William Valdar. Joint estimation of multiple dependent Gaussian graphical models with applications to mouse genomics. Biometrika, 103(3):493–511, September 2016.

[XTLZ02] Yingcun Xia, Howell Tong, W. K. Li, and Li-Xing Zhu. An adaptive estimation of dimension reduction space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64(3):363–410, 2002.

[Yu97] Bin Yu. Assouad, Fano, and Le Cam. Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics, pages 423–435, 1997.

[ZMY18] Hui Zou, Qing Mai, and Yi Yang. Multiclass Sparse Discriminant Analysis. Statistica Sinica, 2018.

[ZNC+19] Hamim Zafar, Nicholas Navin, Ken Chen, , and Luay Nakhleh. SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data. Genome research, 29(11):1847–1859, 2019.

[ZWT19] Feng Zhang, Yu Wu, and Weidong Tian. A novel approach to remove the batch effect of single-cell data. Cell Discovery, 5(46), 2019.