University of Southampton

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
ELECTRONICS AND COMPUTER SCIENCE

Outlier Detection and Subspace Learning via Structured Low Rank Approximation with Applications to *Omic* Data

by

Omar Essam Shetta

Thesis for the degree of Doctor of Philosophy

September 2020

Abstract

Dimensionality reduction is crucial when dealing with data with very high dimensionality and low number of samples. This is the case with genomic data where sequencing many genes is much easier than gathering many different samples. The main problem with high-dimensional data is that statistical inference and traditional pattern recognition techniques would break down or give misleading results. Therefore, we need to reduce the dimensionality of the data before extracting any useful information from it. A widely used dimensionality reduction technique is Principal Component Analysis (PCA). However, it is known from the literature that this method breaks down in the presence of even a small number of outliers in the data. We have reason to believe that outliers are present in genomic data due to shortcomings from the used experimental equipments, sensor malfunctions, and mistakes in the sample gathering processes. Moreover, outliers could be samples that are of interest in the problem that is being investigated, and need to be retained for further investigation.
In this work we will investigate low rank approximation methods that are robust to outliers, much of which have been already introduced in the machine learning community, and they are formulated as convex optimization problems. The main advantage of the convexity of this problems, is that it can be solved iteratively in an efficient way using first order optimization algorithms. However, outlier robust low rank approximation models, such as Outlier Pursuit (OP), that is optimal for high-dimensional genomic datasets, assume that the data lies approximately along a low-dimensional linear subspace; which is a strong assumption when dealing with gene expression or any biological dataset. Inspired by previous work in the computer vision community, we exploit the usefulness of adding a graph regularization term to OP, by building a graph between the data points to model the local geometry structure of the input data. This algorithm is called Graph regularized Outlier Pursuit (GOP), and it has the beneficial advantage of being a convex optimization problem. We will show the effectiveness in outlier detection and low-dimensional visualization of both techniques on high-dimensional genomic datasets. Furthermore, we show here that GOP and OP give better outlier detection results than traditional density based methods used for anomaly detection. Moreover, we will show the enhanced visualization capability of GOP when compared to OP, PCA, and t-distributed Stochastic Neighbour embed-

ding (t-SNE).
Stemming from GOP, this work also proposes as novel method for multi-view clustering based on subspace learning, dubbed Convex Graph regularized Robust Multi-view Subspace Learning (CGRMSL). CGRMSL is robust to outliers and incorporates the non-linearities present in the different views. Moreover, the proposed multi-view method is also based on a convex objective function which guarantees a global optimal solution. We will investigate the power of this novel method on cancer multi-*omic* datasets for applications such as: cancer subtype clustering and cancer subtype discovery.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Omar Shetta, declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as: [5]

Signed: .....

Date: .....

Acknowledgements

I would like to start by thanking both my supervisors Niranjan and Sri for their support and contribution throughout the PhD journey. Special thanks to them for mentoring and showing me, through leading by example, on how to be a successful academic. I also want to express my gratitude to all the academics and students of the VLC, it has been a great environment to work and interact in. Everyone has been very friendly and it has made the lab a great place to come to everyday. I could not imagine that I would meet such amazing people or make such good friends during my PhD. Im going to miss the friendly VLC football matches, squash, and our lunch time conversations.
Now for my family, I want to thank my: mother, father, and brother, for supporting me throughout my lifetime, especially during this last 4 years. I would not have made it this far without them.

Chapter 1

Introduction

High-dimensional datasets have significantly more dimensions compared to the number of available samples. With this type of dataset, statistical and machine learning methods are not effective in giving reliable predictions and often not even applicable. Therefore, machine learning methods that reduce the dimensionality of the data are of crucial importance. They will reduce the dimensionality of the data making it fit for clustering, visualization or any kind of statistical data analysis. One such example of high-dimensional data is gene expressions. The gene expressions of a patient are measured using high-throughput technologies; this gives an insight on the patient's specific genomic profile. Identifying clinically relevant disease subgroups are aided by monitoring genomic measurements [6, 7]. Moreover, genomic profiling is one of the key approaches used to find potential biomarkers and therapeutic targets for distinct cancer types [8]. In the previous years the bioinformatics community have made a huge effort in making available a large quantity of genomic data, such as the Gene Expression Omnibus (GEO) [9] and the Cancer Genome Atlas(TCGA) [10]. They provide easy access to thousands of normalized datasets for most cancer types.
Machine learning techniques, being supervised or unsupervised, can be applied efficiently to genomic data to extract useful information, as an example, robust regression, which is a supervised learning technique, has been applied recently with much success in [11, 12]. This work shows that the concentrations of proteins in cells could be predicted from mRNA levels; this was used to extract outliers and show that they are post-transnationally regulated. However, this method based on regression will break down if applied directly to the gene expression data because of its higher number features (genes) compared to its number of samples.
The reason for the high dimensionality in the genomic datasets is due to the fact that, recently novel high-throughput technologies have emerged to sequence the human genome. These high-throughput technologies are capable to sequence approximately 40 000 genes per sample. Moreover, gathering samples from different patients is a time consuming and non-economic task. As a consequence most genomic datasets

have a sample size much smaller than the number of genes. Thus, simple statistical techniques, such as regression, where the analytical solution relies on taking the inverse of a covariance matrix of a high-dimensional dataset, cannot be directly applied [13]. In addition, complex models that are heavily parametrised such as deep learning models, can overfit the dataset's high-dimensional space when not enough samples are provided. Therefore, the dimensionality of the data needs to be reduced before applying any inferential technique.
Available approaches which reduce dimensionality are: feature selection and feature extraction. Feature selection consists in choosing a subset of genes that best describe the effects of all genes. On the other hand feature extraction finds a low-dimensional space. Each feature of the low-dimensional space is the *extracted* feature; it is constructed by a function of the datasets original features. In the case of linear feature extraction, the *extracted* features are a linear combinations of all the original features. Performance of feature extraction and feature selection are mainly data dependent, this is discussed in [14] and several others.
In this work we will focus on a specific category of feature extraction methods, being unsupervised low rank feature extraction. Low rank feature extraction is categorized as a *linear* dimensionality reduction technique. The main advantage of linear dimensionality reduction techniques are two-fold: (1) They are easily interpretable models as the amount of parameters to be learned is small and the whole goal of such technique is to find a compact representation of the original data. (2) The low rank approximation recovered from such methods is the reconstruction of the low-dimensional space in the original space. A widely known linear low rank feature extraction method extensively used in genomics data analysis is Principal Component Analysis (PCA) [13, 15]. However, there are two main problems with PCA: (1) It does not capture the non-linear structure of the data. Thus, any non-linear structure inherent in the data will be lost in the recovered low rank approximation. (2) It is fragile to even a small proportion of outliers.

In this thesis we aim at making two contributions summarised as follows:

1. The first is to address the aforementioned drawbacks of PCA without losing the ease of interpretability. For this purpose, we have devised a novel low rank approximation method that takes into account the non-linear structure of the data, and is robust to outlier samples. It is formulated as a convex optimization framework that is most suitable to high-dimensional genomic datasets. This newly devised method will be referred to as Graph regularized Outlier Pursuit (GOP).

2. The second contribution of our thesis is to extend the new convex low rank approximation method to take into account datasets with multiple views, known

as multi-view datasets, and in the case of genomic datasets these are known as multi-*omic* datasets. The novel multi-*omic* low rank approximation model shows promising results in cancer subtype discovery. This novel method is Convex Graph regularised Subspace Learning (CGRMSL).

Before we explain the motivation behind both our contributions, we need to briefly introduce the *omic* datasets which will be investigated in this thesis; and explain the concept of linear low rank matrix approximation, as it is a building block that our work stems from.

1.1 *Omic* Data Introduction

With the new advances in high-throughput technology large amounts of molecular data measurements are available, this has helped the biomedical research field to tackle different kinds of diseases. The suffix *omic* to a molecular term signifies the study of that molecule in large quantities. The first *omic* field that has emerged is the genomic field. Genomics is the study of the entire genome. A genome is the complete set of genes of an organism; it is gathered using high-throughput DNA sequencing techniques. It emerged in the 1980s and it has been prominent since then. The advancements made in genomics have paved the way to other *omics* which are now significantly found in the academic field [16]. Different aspects of the genome can be measured by using different sequencing assays leading to different *omic* datasets [17]. The big data fields, that stem from sequencing assays measuring products of transcription (transcripts) and translation (proteins), are referred to as transcriptomics and proteomics respectively.
Transcriptomics is the genome-wide study of RNAs stemming from the biological process of transcription. Transcription is the intermediate biological process of the central dogma of molecular biology that codes a gene from DNA to RNA. The central dogma of molecular biology is the two step process of transcription and translation, that code a gene, in the form of DNA, to a protein as a sequence of amino acids. The quantitative branch of transcriptomics measures how much of a transcript is present, or in other words, how much of a gene is expressed. Collecting the expressions of all the genes in a sample gives rise to the gene expression datasets. The large number of transcripts in biological samples are sequenced using Next Generation Sequencing (NGS) technologies, such as: Microarray sequencing and RNA-sequencing (RNA-seq). In addition to protein coding RNAs, the NGS technologies are capable of sequencing even short RNAs, such as microRNAs. MicroRNAs regulate a number of different target genes, hence participate in the regulation of biological processes. Abnormalities in the regulation of microRNAs are often the cause of pathologies [18]: one such example is cancer formation and development [19]. Thus, microRNAs can

be used as biomarkers or drug targets.
Proteomics is the large scale study of proteins. Proteins are constructed by translating mRNA into a sequence of amino acids, which then fold to form functional biological structures called proteins. Proteins are complex molecules that play a key role in controlling the enzyme activity, protein renewal and transport, and maintaining the cells structure. The proteome is sequenced using Mass Spectroscopy (MS) technology. However, quantitative proteomics methods based on MS only measures a small proportion of the proteome [20]. This makes it hard to be integrated with other *omics* or used by itself to analyze biological or pathological processes.
Another important *omics* is Epigenomics, the genome wide study of reversible modifications to DNA, such as DNA methylation. DNA methylation can be caused by both environmental and genetic factors; it can last for a long period of time, and when caused by genetic factors these can be inherited [21]. More importantly, genome-wide studies have shown that methylated regions of DNA can be indicators for cancer [22].
The sequencing technologies mentioned above are bulk sequencing technologies. They take a sample which is comprised of a large number of cells, and the specific sequencing assay measures the molecules of the whole group of cells. This averages the measurement over the whole group of cells and can lose individual cell information. Recently optimized Next Generation Sequencing (NGS) technologies have emerged that sequence information of each individual cell provides a higher resolution to capture cellular differences; thus, giving a better understanding of the function of individual cells [23]. For example, tumors samples are known to be heterogeneous, meaning that different types of cells are present in one specific tumor. This ability to sequence each tumor cell will help in understanding the tumor ecosystem. Many single-cell sequencing technologies have recently emerged across the different *omics*, such as single-cell genomics, epigenomics, and transcriptomics. They have led to novel findings in the cancer research field, specifically in that of: metastasis, cancer evolution, therapy resistance, and tumor microenvironmen [24, 25].
Both bulk and single-cell *omic* datasets are high-dimensional. As mentioned before, this results from the ease of the new high-throughput technology to sequence the whole genome and by the more costly procedure of gathering samples from patients. If the genomic datasets is denoted by the matrix M, then is has dimensions $p \times n$, with p being the measured molecular feature and n is the number of samples; the dataset is high-dimensional meaning that $p \gg n$.

1.2 Notation

Throughout this thesis we will follow a few conventions that are stated below. In some sections the reader will be reminded of some of them. When needed, more specific notation will be introduced.

1. Vectors are represented as bold lower case letters, such as $\mathbf{x}$. The vector of all zeros is expressed as $\mathbf{0}$.

2. Matrices are represented as upper case letters, such as X. When data matrices are mentioned the convention is that $M \in \mathbb{R}^{p \times n}$ has n data samples with p features. For any matrix X, $X_{i,j}$ represents the $(i,j)^{\text{th}}$ entry, $\boldsymbol{X}_{i,:}$ is the i^{th} row vector, and $\boldsymbol{X}_{:,i}$ denotes the i^{th} column vector. The transpose of X is expressed as X^T.

3. Multi-view data is a collection of data matrices, with each view represented by a data matrix. The v^{th} view of a multi-view data structure is represented as M_v. For the v^{th} view of any multi-view data structure, the $(i,j)^{\text{th}}$ entry is denoted as $X_v^{i,j}$, the i^{th} column vector is expressed as $\boldsymbol{X}_{\boldsymbol{v}}^{:,i}$, and the i^{th} row vector is represented as $\boldsymbol{X}_{\boldsymbol{v}}^{i,:}$.

4. Scalars are represented as lower case italic letters, such as α.

1.3 Linear Low Rank Matrix Approximation

Linear low rank matrix approximation methods seek to find the nearest low rank approximation to the input data matrix. The general linear low rank approximation model follows this decomposition:

$$M = L + \mathcal{N}.$$

Where $M \in \mathbb{R}^{p \times n}$ is the input data matrix with n samples that have p features, $L \in \mathbb{R}^{p \times n}$ is the low rank approximation, and $\mathcal{N} \in \mathbb{R}^{p \times n}$ is the noise matrix. The assumption of linear low rank methods is that the given data matrix is inherently low rank and is corrupted by a specific noise model. The difference between linear low rank approximation methods is the noise matrix corruption model. In this thesis three types of corruption models will emerge:

- Solving the decomposition with $\mathcal{N}$ being a dense matrix with entries sampled from and i.i.d Gaussian random variable results in classical PCA (**Refer to Appendix A.3 for proof**). Throughout this thesis the PCA low rank decomposition will be known as:

  $$M = L + N.$$

  (Noise matrix for the PCA low rank decomposition will be denoted from now on as N).

- Solving the decomposition with with $\mathcal{N}$ being a sparse matrix or Laplacian distributed. Both conditions are equiavalent because when $\mathcal{N}$ is Laplacian dis-

tributed the sparse structure is indirectly induced thorugh the minimzation of the l_1 norm. Maximizing the log likelihood of the data M with $\mathcal{N}$ being Laplacian distirbuted is the same as minimizing the l_1 norm of the noise matrix (**Refer to Appendix A.4 for proof**). It is known, from the sparse coding literature, that minmizing the l_1 norm gives a sparse solution. The sparse structure on $\mathcal{N}$ gives a low rank and sparse decomposition problem; well known as Robust PCA (RPCA) [2]. Throughout this thesis the RPCA low rank decomposition will be indicated as:

$$M = L + E.$$

(Noise matrix for the RPCA low rank decomposition will be denoted from now on as E).

- When $\mathcal{N}$ is a column sparse matrix, corruptions are column-wise. This gives the known method Outlier Pursuit (OP) [3]. Throughout this thesis the OP low rank decomposition will be known as:

$$M = L + C.$$

(Noise matrix for the OP low rank decomposition will be denoted from now on as C).

1.3.1 Classical Principal Component Analysis (PCA)

PCA has the aim to find directions that maximize the variance of projections of the data points. One way to find them is by computing the empirical covariance matrix of the data matrix, then finding its eigenvectors. PCA is also formulated as a low rank approximation problem; where we need to find a low rank matrix that best approximates the data, assuming that the data lies near a low-dimensional subspace. More precisely, if we stack the data points as column vectors of a data matrix $M \in \mathbb{R}^{p \times n}$ with p dimensions and n samples, the column vectors of M should approximately lie onto a low-dimensional subspace with dimension $r \ll \min(p, n)$. This is expressed mathematically as:

$$M = L + N,$$

where L is low rank with rank-r, and N is a dense matrix with entries sampled from i.i.d Gaussian random variables. Classical PCA estimates the best low rank approximation (in the l_2 sense) of M. This can be formulated as an optimization

problem :

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & ||M - L||_F^2 \\ \text{subject to} \quad & \text{rank}(L) \leq k. \end{aligned}$$

(Here, $||M||_F^2 = \sum_i^n ||\mathbf{M}_{:,i}||_2^2 = \sum_{ij}^n M_{i,j}^2$ is the Frobenius norm of M; that is the sum of the l_2 norm squared of all the columns of M, or the sum of all elements squared of M).

This problem can be solved efficiently and reliably by the Singular Value Decomposition (SVD). If the SVD of M is $M = USV^T$, where U and V are matrices with orthonormal columns representing the column space and the row space respectively, and S is a diagonal matrix with elements in its diagonal being positive and in decreasing order called singular values. The low rank matrix L that satisfies the previous minimization problem is expressed as follows

$$L = U_{m\times k}\, S_{k\times k}\, V_{n\times k}^T$$

where $U_{m\times k}$ is the matrix with the first k columns of U, $V_{n\times k}$ is the matrix with the first k columns of V, and $S_{k\times k}$ is the square diagonal matrix with the first k singular values of S.

Although PCA is such a well-defined problem with an efficient tool to solve it, it also carries the drawback that in the presence of a few outliers in the data matrix the low rank estimate is highly corrupted. This is because the outliers will lie far away from the low-dimensional subspace and their l_2 norm squared will be very large compared to non-outlier samples; this will force the low-dimensional subspace to fit closer to the outliers. From Figure 1.1 the effect of as few as two outliers is seen on a synthetic datasets with 50 samples generated from a 2-dimensional Gaussian distribution. From the Figure the black vector is the first principal component which is heavily skewed toward the outliers; and the blue one is the original principal component without the outliers.

To solve this fragility to outliers, two different robust corruption models to outliers have emerged in the machine learning community: sparse noise (E) used in RPCA [2], and column sparse noise (C) [3] used in OP. Both RPCA and OP will be introduced in more detail in Chapter 2.

1.4 Motivation: First Contribution

Principal Component Analysis (PCA) is a linear low rank feature extraction method that is widely used for data compression [15]. PCA has been largely applied on ge-

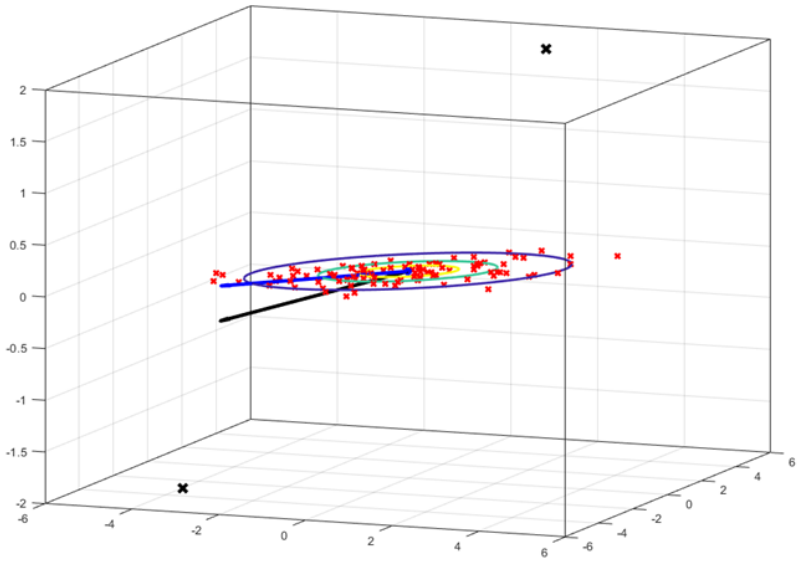


Figure 1.1: PCA fragility to outliers. Figure shows that the first principal component (**black** vector) is skewed towards the outliers even though they are a much smaller proportion compared to the normal data.

nomic data to reduce dimensionality, as an example in [26–28].

There are many applications of PCA in genomic data analysis that include the following areas. (1) Data visualization [29]; when the data is in very high dimensionality, as in the case of genomic data, this application comes of use. With PCA the data is projected onto two or three dimensions, which makes visualization possible. Several data examples are found in [27], where data is visualized in two dimensions. (2) Clustering analysis; by projecting the data onto the first few PCs, most of the variation of the genomic data can be contained by these lower dimensions. Then the first few PCs can be used instead of the whole data to cluster genes or samples [30]. (3) Regression Analysis; predictive models for disease outcomes are widely used in pharmacogenomics studies to predict response to treatments. As the genomic data has much higher dimensionality than sample size, normal linear regression analysis will result in erroneous estimates [13]. It has been shown in [31] and references therein, that it is possible to first use PCA to find the first few PCs and then use standard regression taking the first few PCs as predictors.

Although PCA is extensively used in Bioinformatics, it has some serious drawbacks when the datasets contain outliers as seen in Figure 1.1. Even the presence of one outlier can drastically affect the output of PCA [2, 32–34]. Such outliers may arise from sensor failures, mislabelling of samples or malicious tampering. In general, outliers can be a small proportion of a dataset that is functionally different from the majority of a population. Taking cancer for example, the outlier samples may result

in some interesting special instance of the disease [35]. Hence, it is not always the case that the outliers need to be removed. As a result, robust implementations of PCA that detect outliers and (robustly) estimate PCs, become of great importance. The early beginnings of robust PCA, started by using robust estimations of covariance matrices [36–38]. But they had two main problems: the methods were not resistant to a large number of outliers, and are limited to datasets of small to moderate dimensionality [39]; therefore, not applicable to higher dimensions. A second approach to robust PCA is to find directions that maximise a robust estimate of the variance, such as Projection-Pursuit [40]. This can be applied to datasets of higher dimensionality, but it suffers from the fact that it is a non-convex problem and can become computationally intractable with the increase in problem size [3]. Thus, two state-of-the-art robust PCA methods that solve a low rank approximation problem are RPCA and OP [2,3]. They were briefly introduced in Subsection 1.3 and will be further discussed in Chapter 2. Both these methods solve convex optimization problems; guaranteeing that first order optimization problems will find a global solution. Moreover, both have computationally efficient solutions that work in high-dimensional settings.

1.4.1 Corruption Models on Synthetic Data

So far we have introduced three corruption models: dense noise (N), sparse noise (E) and column sparse noise (C). The three different low rank dimensionality reduction models that assume these noise structures are: PCA, RPCA and OP respectively. The dense noise matrix, N, is assumed to have its entries sampled from a zero mean Gaussian distribution with isotropic covariance, $N_{i,j} \sim \mathcal{N}(0, \sigma^2)$. The classical PCA problem minimizes the l_2 norm squared of the reconstruction error of each sample; which is equivalent to maximising the log likelihood of the data, assuming the aforementioned Gaussian noise model (**Refer to Appendix A.3 for proof**). The estimation of the mean of the Gaussian probability distribution is skewed considerably to accommodate the outlier samples as the squared term will give the outlier samples a much higher weight than the rest of the normal samples. A probability distribution that is more robust to outliers is the Laplacian distribution, which is the distribution assumed by the RPCA problem 2.13. By minimizing the l_1 norm of the reconstruction error of each sample, the model is assuming that the entries in the noise matrix E are sampled from a zero median and b Mean Absolute Deviation (MAD) Laplacian distribution, $E_{i,j} \sim \text{Laplace}(0, b)$ (**Refer to Appendix A.4 for proof**). The RPCA model assumes sparsity on all the entries of noise matrix E. On the other hand, a more effective corruption model for outliers is the column sparse model assumed by OP. The $l_{1,2}$ norm of OP groups columns as one object and the sparsity is induced on a whole column; this stems from the definition of the $l_{1,2}$ norm. To show the robustness of RPCA and OP to outliers, three synthetic datasets with

different dimensions are set up. The three datasets are constructed with the same sample size of $n = 500$ and with different dimensions of low, moderate, and high dimensionality with p= 2, 40, and 800 respectively. Each dataset is generated by first sampling the main 500 samples from a Gaussian distribution with zero mean and positive definite covariance matrix C, $\boldsymbol{M}_{:,i} \sim \mathcal{N}(\mathbf{0}, C)$ for all samples i. Then, n_{out} outliers are sampled from a different Gaussian distribution with a mean vector $\boldsymbol{\mu}$ with large entries and isotropic covariance matrix, $\boldsymbol{Y}_{:,i} \sim \mathcal{N}(\boldsymbol{\mu}, I)$ for all samples i. For each dimensionality reduction method we are going to investigate how much the reconstructions are skewed towards the outliers. The reconstructions of PCA, RPCA and OP on the 1st principal component are computed for each dataset for a range of six different number of outliers, $n_{out} = 5, 10, 15, 20, 25, 30$. The angle θ between the 1st principal component of a given method and the 1st principal component of PCA without outliers is used as metric to evaluate the deviation towards outliers for each method. Figure 1.2 shows the illustration comparing the reconstructions of the three different methods on the 2-dimensional synthetic data with 10 injected outliers. It is seen that PCA is considerably skewed towards the outliers, whereas RPCA and OP are not affected. Figure 1.3 shows the angular deviation θ in the 1st principal component for the three methods at increasing numbers of injected outliers. We can see that with increasing dimensions the 1st principal component of RPCA is being skewed towards the outliers. Whereas, OP is robust in all three dimensions including the high-dimensional case.

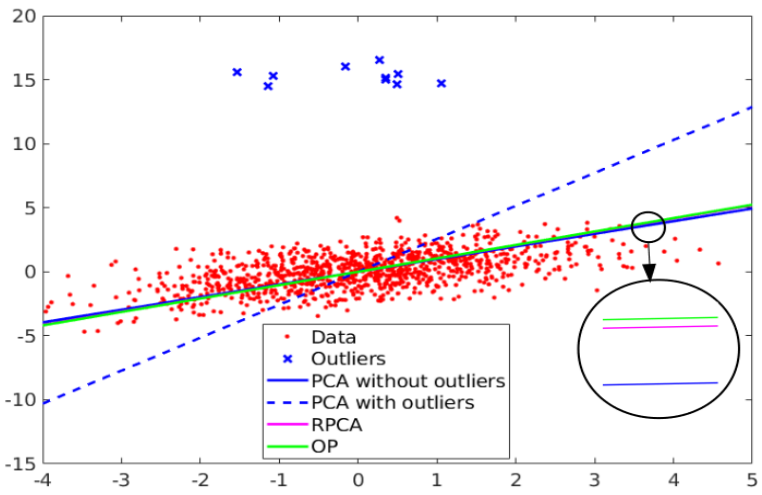
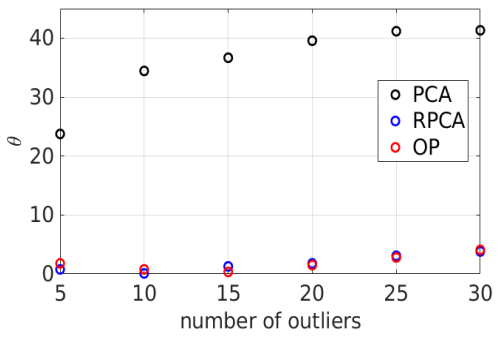
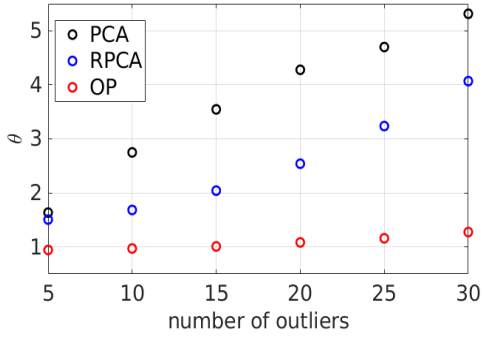


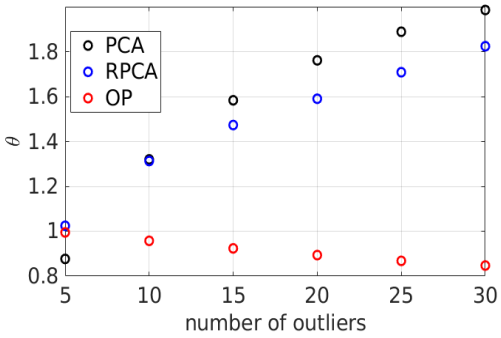
Figure 1.2: Comparison of three different corruption models: PCA, RPCA and OP on a 2-dimensional synthetic dataset. It is seen that the reconstruction estimated by PCA is skewed towards the outliers. Whereas, RPCA and OP are both robust to the outliers.



(a) Data matrix dimensionality; p =2.



(b) Data matrix dimensionality; p =40.



(c) Data matrix dimensionality; p =800.

Figure 1.3: Comparison of three different corruption models: PCA, RPCA and OP on three different dimensions. θ is the angle between the 1st principal component of each method with the 1st principal component of PCA without the outliers. We can see that PCA has the largest θ. Whereas, RPCA becomes less robust with increasing dimensions, and OP is robust in in all dimensions including the high-dimensional setting.

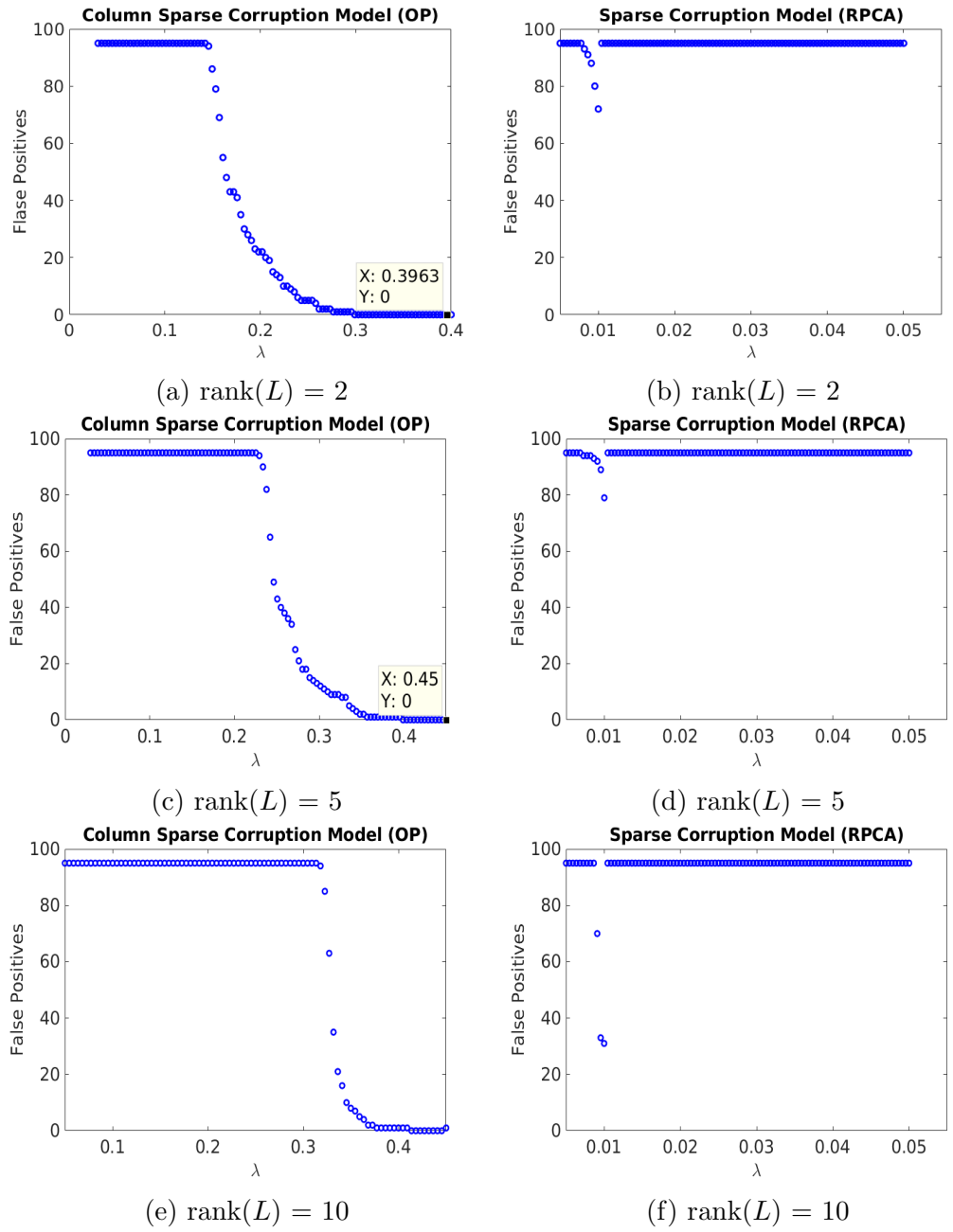
1.4.2 OP vs RPCA in High-Dimensional Spaces

In this thesis we are interested in high-dimensional datasets such as genomic datasets. In high-dimensional spaces the *column sparse* corruption model is a more effective model to detect outliers compared to the *sparse* corruption model of RPCA. We show this by setting up a synthetic experiment to compare the outlier detection capability of both models. The clean part of the synthetic dataset, $L \in \mathbb{R}^{p \times n_{clean}}$, is constructed by $L = AB$, with $A \in \mathbb{R}^{p \times r}$ and $B \in \mathbb{R}^{r \times n_{clean}}$ sampled from a standard Gaussian distribution. With L constructed in this way it lies on an r-dimensional subspace. The corrupted samples or outliers, are injected by concatenating matrix L by $C \in \mathbb{R}^{p \times n_{out}}$. C is constructed with every column being an identical copy of an adversarially generated vector. For our high-dimensional synthetic dataset we choose $p = 2000$, $n_{clean} = 95$, $n_{out} = 5$, and three different ranks of L: $r = 2, 5, 10$. After applying RPCA and OP, the reconstruction error is computed for each sample by

$e_i = ||\mathbf{C}_{:,i}||_2$. Afterwards, samples are ranked in descending order of their reconstruction error; and the ones with high error, e_i, are considered outliers. The metric used to evaluate outlier detection performance is the false positives encountered before detecting all of the known 5 outliers (in this case the positive class is the outlier class, and the threshold of detection is set to the smallest reconstruction error of the 5 outliers). As seen from Figure 1.4, for the three different ranks of L, the *column sparse* corruption model detects zero false positives after tuning the regularization parameter λ (λ is a regularization parameter that needs to be tuned in RPCA and OP). In contrast, the *sparse* corruption model detects many false positives even after sweeping though a suitable range of λ. This shows that the OP *column sparse* corruption model is more efficient in detecting outliers in high-dimensional datasets such as genomic datasets that are investigated in this thesis.

However, a crucial drawback of OP is that it cannot model the non-linearities of the data, and it can not model datasets that lie on a manifold. This becomes a disadvantage when it comes to complex biological data, such as genomic datasets, where the relationships between the variables can be non-linear [41].

In recent years graph regularized dimensionality reduction models have emerged, that inject into the learning model the intrinsic manifold structure of the data in the form of a spectral graph. This graph regularization will smooth the low-dimensional embedding or low rank matrix that is to be learned onto the intrinsic manifold structure of the data.



(a) rank$(L) = 2$ (b) rank$(L) = 2$

(c) rank$(L) = 5$ (d) rank$(L) = 5$

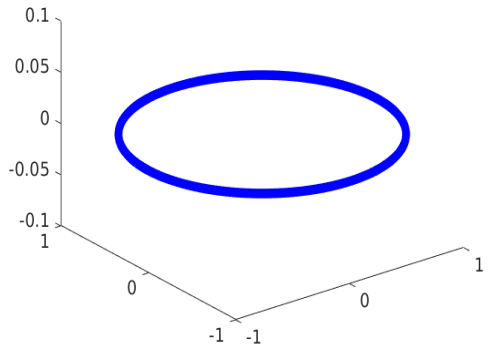
(e) rank$(L) = 10$ (f) rank$(L) = 10$

Figure 1.4: Comparing *column sparse* and *sparse* corruption models on synthetic high-dimensional dataset, with $p = 2000$ features and $n = 100$ samples, at three different ranks of L; $r = 2, 5, 10$. Metric used to evaluate outlier detection performance is false positives encountered before detecting all known outliers. λ is a regularization parameter that needs to be tuned in both RPCA and OP algorithms. It is shown that the OP corruption model is more effective in detecting outliers in high-dimensional spaces compared to the RPCA corruption model.

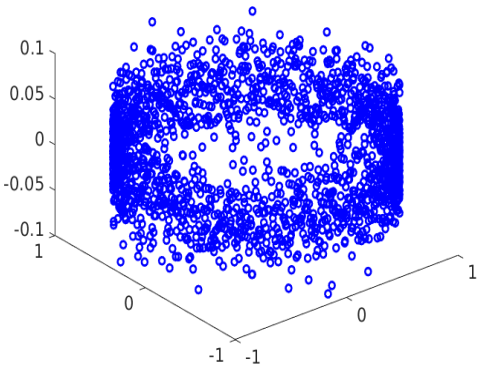
1.4.3 Usefulness of the Graph Regularizer

Here we will show the effectiveness of our method, graph regularized OP (GOP), to capture the non-linear structure of the data by constructing a synthetic example. GOP seeks a graph structured linear low rank approximation to the data with a *column sparse* corruption model. It applies a linear model to compute the low rank decomposition of the data: $M = L + C$. However, the non-linear structure is captured in the low rank matrix L by a spectral graph which models the geometric structure inherently present in the data (refer to Subsection 2.2.1 for graph construction). A synthetic non-liner dataset is generated, Figure 1.5(a) shows the original 2-dimensional structure of the data which is a circular structure. Gaussian noise is then added in the 3^{rd} dimension (Figure 1.5)(b) to corrupt the original structure. The aim here is to recover the original 2D non-linear structure of the data. As expected OP, which is a linear subspace method, collapses all the data points into the center of the space, failing to recover the underlying structure of the data (Figure 1.5(c)). In contrast, our method recovers successfully the non-linear de-noised structure of the data (Figure 1.5(d)), showing the effectiveness of incorporating a graph regularizer. The topological circular structure of the data is maintained through the graph regularizer; and the low-rank approximation filters the Gaussian noise. Thus, GOP is able to recover the clean 2-dimensional circular structure of the dataset. One could argue that this can also be achieved by non-linear dimensionality reduction algorithms, such as the Laplacian Eigenmap (LE) [42] and the Isometric feature Mapping (ISOMAP) [43]. However, these methods can only find the low rank manifold structure of the data in a low-dimensional space. The difference is that graph regularized linear low rank methods would find the principal component and principal directions allowing the low rank manifold structure of the data to be represented in the original space. This is crucial as it will allow reconstruction errors to be computed which is are needed for identifying outliers.

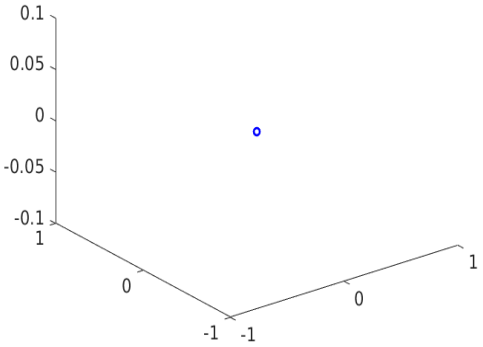
In the light of the shown properties of the *column sparse* corruption model on high-dimensional datasets (in Subsection 1.4.2), and the usefulness of a graph regularizer, we will combine, in the first aspect of this thesis, these two properties. We will show that this detects outliers effectively and recovers the underlying low rank manifold structure of genomic data while retaining the interpretability of linear low rank models.



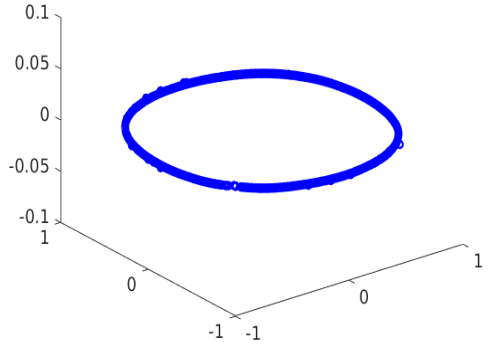
(a) Original 2D Circular Data.



(b) Circular Data with Added Noise in 3rd dimension.



(c) **OP**: Low rank matrix.



(d) **Our method (Graph Regularized OP)**: Low rank matrix.

Figure 1.5: Illustrating effectiveness of graph regularizer. (d) Shows that the graph regularized linear low rank method, (GOP), is capable of capturing the non-linear structure of the constructed 2D circular data shown in (b). By contrast (c) shows that, with the absence of the graph regularizer, the linear low rank method by itself, OP, fails to extract the non-linear structure of the data.

1.5 Motivation: Second Contribution

Several methods for supervised and semi-supervised learning are found in the biomedical literature [44,45] that classify tumor samples either as benign or malignant or into different molecular subgroups. To exploit the usefulness of such methods, reliability and availability of labels is of great importance. In the case of *omic* data gathering tissue samples and their labels is a high-cost process, thus limiting their availability. Therefore, supervised and semi-supervised methods of cancer classification suffer

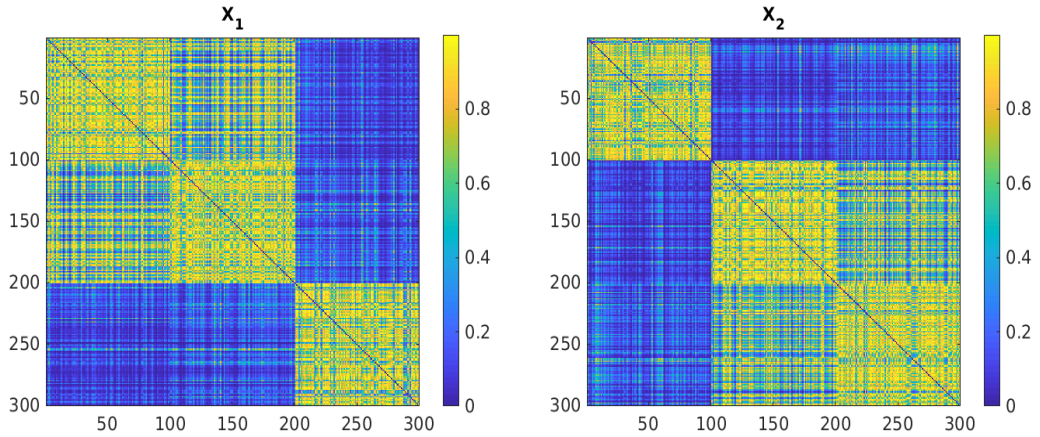
from limited sample sizes and potentially missing labels.
Clustering techniques have proven useful with *omic* data, as it is unbounded by the availability of labels. Moreover, clustering for bioinformatic data is a useful pattern discovery technique, which is the initial step taken towards data exploring [46]. Clustering is especially of great use in the emerging field of precision medicine in discovering disease subtypes [47].
Clustering single *omic* data separately has proven useful in exploring useful patterns in the data. Nevertheless, exploring more than one *omic* data for the same set of samples has shown better capability in extracting more complex structures. Specifically, molecular subtype discovery has been shown to greatly benefit from multi-view clustering compared to the single-view counterpart [48]. Multi-*omic* clustering is more advantageous than the single *omic* counterpart for several reasons. First, multi-*omic* clustering can take into account multiple molecular levels, such as transcriptomic and proteomic levels. Second, *omic* data can aggregate information from different organismal levels, such as gene expression and microRNA expression. Third, *omic* data can filter the biological and experimental noise present in the data.
Multi-view clustering methods have been widely studied in the machine learning community; these methods can also be used on multi-*omic* data [49–55]. State-of-the-art multi-view clustering techniques are formulated as non-convex optimization methods [50–53, 55, 56], which can only guarantee convergence to local optimal solutions that are computationally expensive. There are a few multi-view methods formulated as convex optimization problems; but these methods do not take into account the non-linear manifold structure of the data and are not robust to outliers.
Therefore, we have formulated a robust to outliers convex multi-view subspace learning method that seeks to find a shared low-dimensional subspace; and that takes into account the non-linear manifold structure of the different data views and is robust to outlier samples in each view. This is all modelled in a single convex optimization problem.
In the next subsection we motivate the application of multi-view methods as compared with single-view methods in multi-view data cluster identification.

1.5.1 Synthetic Example: Multi-View Subspace Learning Compared to Single-View Subspace Learning

Here we will show the effectiveness of our convex multi-view subspace learning method compared to single-view subspace learning. We generate a synthetic multi-view high-dimensional dataset. The dataset has two views with each view having $p = 1000$ features of the same set of $n = 300$ samples/instances. The samples are generated to have three distinct clusters, with each cluster having 100 instances. Each view has two of the three clusters overlapping and the third is clearly separate. The two views

are constructed to have complementary information. Each view is synthesized using the following procedure: a mixture of three 2-dimensional Gaussians is modelled with three distinct means, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$ and $\boldsymbol{\mu_3}$, and with all three covariance matrices being the identity matrix. Each Gaussian component has 100 samples which gives a total of 300 samples. Finally, the v^{th} view is constructed by projecting the low-dimensional dataset $Q_v \in \mathbb{R}^{2\times 300}$ (v can have a value of 1 or 2) into a 1000 dimensional space: $X_v = U_v Q_v$. With $U_v \in \mathbb{R}^{1000\times 2}$ being a randomly generated projection matrix, and $X_v \in \mathbb{R}^{1000\times 300}$ being the v^{th} synthesized view. The location parameters for the first view are: $\boldsymbol{\mu_1} = \{1, 2\}$, $\boldsymbol{\mu_2} = \{1, 4\}$ and $\boldsymbol{\mu_3} = \{6, 6\}$. For the second view they are: $\boldsymbol{\mu_1} = \{6, 6\}$, $\boldsymbol{\mu_2} = \{1, 2\}$ and $\boldsymbol{\mu_3} = \{1, 4\}$. The adjacency matrix of both views are shown in Figure 1.6.

To demonstrate the usefulness of multi-view subspace learning, we compare it to single-view subspace learning, on each view separately, and on the concatenation of both views. The single-view subspace learning method we compare against is Graph Laplacian PCA (GLPCA) [57] (It will be reviewed in Chapter 2). The 2-dimensional subspace for each single-view using GLPCA is shown in the top of Figure 1.7. It is seen that the subspace found on each view has two of the three clusters highly overlapping. Afterwards, GLPCA is applied on the concatenation of both views. The resulting subspace is shown in the center of Figure 1.7. It is seen that this naive way of integrating views does not successfully integrate the complementary information of each view. Then, we find the shared latent representation using our proposed multi-view subspace learning method, this representation is shown in the bottom part of Figure 1.7. It is seen that our convex multi-view method is capable of efficiently integrating the complementary information within each view and clearly separate the three clusters.



(a) Adjacency matrix (**1st view**). (b) Adjacency matrix (**2nd view**).

Figure 1.6: Adjacency matrix of both views of constructed synthetic dataset. X_1 and X_2 have complementary information that should separate all three clusters.

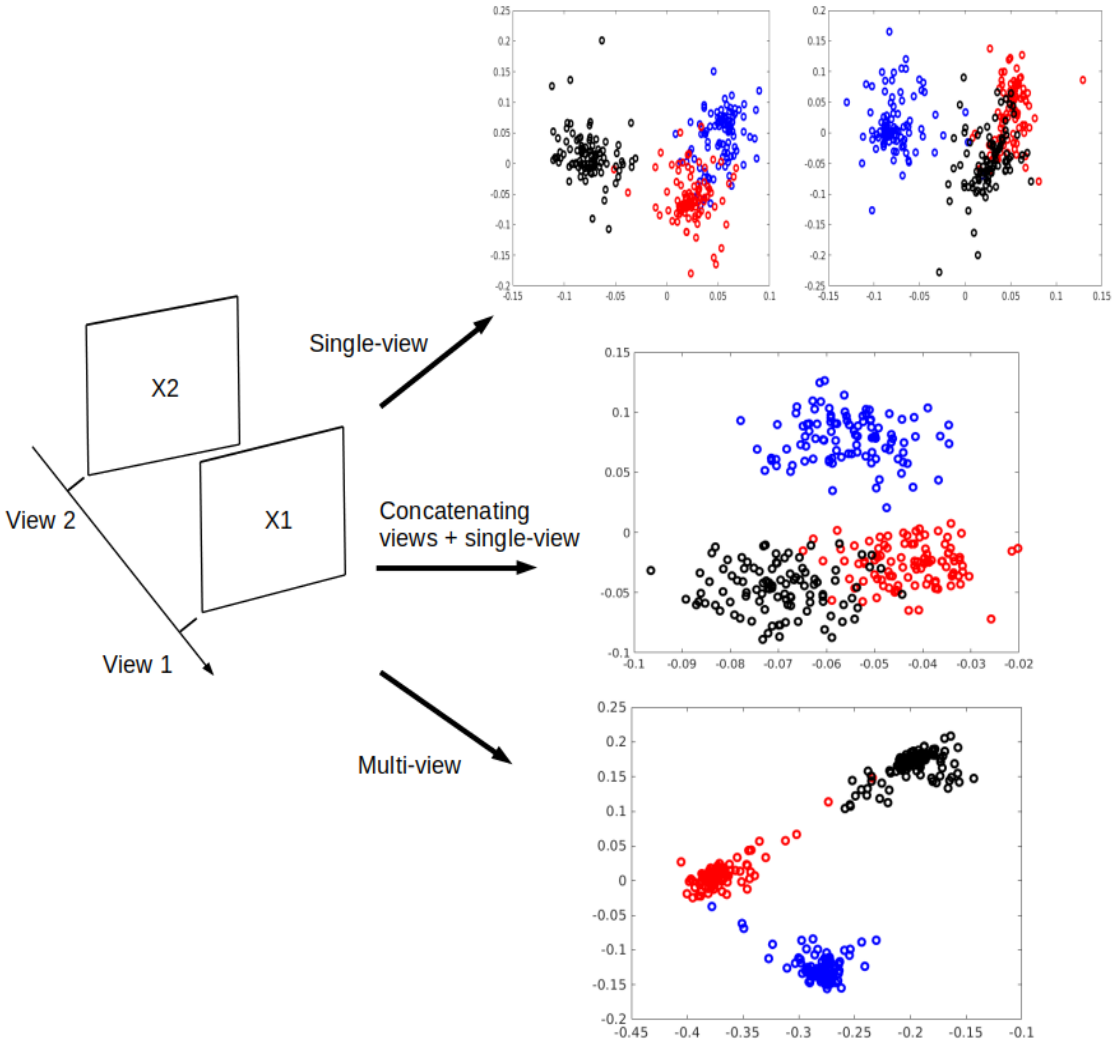


Figure 1.7: Comparing multi-view and single-view subspace learning on high-dimensional synthetic dataset. Each view of the multi-view dataset has $p = 1000$ features and $n = 300$ samples with three clusters having 100 samples each. Clearly the multi-view shared latent space is able to separate the three clusters while the single-view method on each view and on the concatenation of both views has two of three clusters overlapping.

1.6 Goals and Contributions of Thesis

1.6.1 Goals

The goal of this thesis is to devise convex subspace learning methods based on low rank matrix approximation that are robust to outliers and takes non-linearities of the data into account for both single and multi-view datasets.
Moreover, the goals of these methods is to discover in an unsupervised manner the

hidden structures of high-dimensional genomic datasets, especially cancer genomic datasets. The main tools for data pattern discovery that are addressed in this thesis are: 1) Clustering, 2) Visualization, and 3) Outlier/Anomaly detection. Clustering is achieved by applying k-means clustering on the low-dimensional subspace found by our methods. Visualization is a by-product of subspace learning, when the low-dimensional subspace is of two or three dimensions. Moreover, as the proposed subspace learning methods are designed to be robust to outliers, outlier detection is also achieved by inspecting the reconstruction error of each sample. The devised subspace learning methods take into account these three unsupervised data discovery aspects in a single framework.

1.6.2 Contributions

To achieve the goals of this thesis, we propose two subspace learning methods based upon graph regularized robust low rank matrix approximation. The contributions of our work are:

1. A novel convex subspace learning method that is most suitable for high-dimensional genomic datasets dubbed as Graph Regularized Outlier Pursuit (GOP). This method solves two main drawbacks of linear low rank models, by being robust to outliers, and modelling the nonlinearity of the data in the low rank approximation without losing the interpretability of the linear methods (more about this in Chapter 4). The novelty of our proposed technique is in formulating a subspace learning method that has the previous properties and is optimal in detecting outlier samples in high-dimensional data settings. Besides, our method is formulated as a convex optimization problem, which enables simple first-order convex optimization methods to converge to a global solution of the problem. A method similar to ours, that takes into account drawbacks of linear methods, is RPCA on graphs [58] (Reviewed in Chapter 2). This method is optimal for image like corruptions which does not transfer to genomic data structures.

   This work has been published; cited as:
   Shetta, O. and Niranjan, M., 2020. Robust Subspace Methods for Outlier Detection in Genomic Data Circumvents the Curse of Dimensionality. Royal Society Open Science, 7(2).

2. The second contribution of this thesis is Convex Graph regularized Robust Multi-view Subspace Learning (CGRMSL). It emerges from the previous contribution and it is designed to find a shared low-dimensional space for multi-view datasets. The novelty of this method in the context of other state-of-the-art

multi-view subspace learning/clustering methods, is that it is formulated as a convex optimization problem, that takes into account the non-linearity of the different views of the data and it is robust to outlier samples. Thus, it can detect outliers which can then be discarded or analyzed further.

This work is under revision:
Shetta, O., Niranjan, M., and Dasmahapatra, S. Convex Multi-View Clustering via Robust Low Rank Approximation with Application to Multi-Omic Data. Submitted to IEEE Transactions on Computational Biology and Bioinformatics.

The two proposed methods are motivated by the problems encountered in genomics data; being their high dimensionality and presence of outliers due to experimental errors. In this thesis we will illustrate our work on genomic data. However, the methods are generic to high-dimensional datasets that require robustness to outliers. The devised methods address the problem of pattern discovery in high-dimensional data by integrating in a single framework the three aspects of: clustering, visualization and outlier detection. Also, both methods are formulated as convex optimization methods, which guarantee a global solution and a computationally efficient algorithms.

1.6.3 Thesis Structure

The structure of this thesis is as follows:

- **Chapter 1** (***Introduction***): It introduces *omic* datasets and the problem of linear low rank matrix approximation. It discusses the motivation behind the work for both robust single-view subspace learning (first contribution) and robust multi-view subspace learning (second contribution).

- **Chapter 2** (***Literature Review***): It reviews the following: 1) outlier detection methods that emerged in the statistical and machine learning literature. 2) Graph regualrized single-view linear low rank methods. 3) Multi-view and multi-*omic* clustering and subspace learning methods.

- **Chapter 3** (***Gradient Based Methods***): This chapter provides a detailed background of gradient based methods that are used in robust subspace methods and in the subspace methods that we propose.

  The coming three chapters are the main contribution of this thesis.

- **Chapter 4** (***Graph Regularized Outlier Pursuit (GOP)***): It introduces the proposed subspace learning method for genomic datasets. It is part of the first contribution of this thesis.

- **Chapter 5** (***Robust Subspace Methods for Outlier Detection in Genomic Data Circumvents the Curse of Dimensionality***): This chapter highlights the importance of our proposed method (GOP) for genomic datasets both on bulk and single cell measurements. It is part of the first contribution of this thesis.

- **Chapter 6** (***Convex Multi-View Clustering via Robust Low Rank Approximation with Application to Multi-Omic Data***): This chapter is part of the second contribution of this thesis and introduces the proposed method for multi-view subspace learning (CGRMSL); showing its superiority to other state-of-the-art multi-view subspace learning methods in terms of clustering and potential cancer subtype discovery.

- **Chapter 7** (***Conclusion and Future Work***): This final chapter includes concluding remarks and ideas on extensions of our work.

Chapter 2

Literature Review

In this chapter we will review three different groups of problems:

First: We review outlier detection in its three different aspects of supervised, semi-supervised, and unsupervised. We introduce the problem of unsupervised outlier detection, which the robust versions of PCA stem from: RPCA [2], and Outlier Pursuit (OP) [3]. The low rank matrix decomposition of these two robust PCA methods are based on two tightly related lines of work. The first being rank minimization problems such as matrix completion [59]; where a low rank matrix is exactly recovered from a data matrix (that is assumed to be low rank) with few observations. The second is finding a robust solution to linear systems of equations in the presence of arbitrary and unknown corruptions [60]. Both these lines of work will be reviewed before introducing RPCA and OP.
Second: We review single-view Graph Regularized linear low rank approximation methods found in the literature.
Third: We review the different multi-view clustering, and multi-view subspace learning methods found in the literature.

2.1 Outlier Detection: Review

Outlier detection, is a class of problems aiming to find patterns in the data that do not follow an expected behaviour. In general, these non-conforming patterns are usually referred to as anomalies, outliers, surprises or contaminates depending on the application domain. In most application domains, the words outliers and anomalies are used in the context of outlier detection, and these two terms are often used in an interchangeable manner. Outliers in the data could be samples of interest, where useful information can be extracted from, or can be samples that are corrupted and thus can mislead the general trend of the data. In the latter case, the outliers will disrupt the statistical methods used to extract patterns in the data.

Outlier and anomaly detection have been researched in the statistics community starting from the 19 th century [61]. Over the course of time, the study of outlier detection techniques have spread to other research communities which gave a wide variety of techniques.
The main categories of outlier and anomaly detection methods are:

- Supervised: Data labels are present. Normal and outliers samples are labelled.
- Semi-supervised: Only normal samples are labelled.
- Unsupervised: Data labels are absent.

These methods will be reviewed in the following subsections.

2.1.1 Supervised

Supervised outlier detection methods assume that a training dataset is available with normal and outlier samples labelled as distinct classes. These type of methods are first trained using the training datasets which contains the labelled normal and outlier samples. This gives a trained predictive model to distinguish between normal versus outlier samples. Using the trained model, inferences can be made on unseen test samples to determine which class they belong to.
In the supervised outlier detection literature, many machine learning classification methods are used as predictive models, such as: Neural networks, SVMs, Linear Discriminant Analysis (LDA), Bayesian classifiers and many more. Another flavour of classification methods devised in the outlier detection literature are one-class classification methods.
Outlier detection methods based on one-class classification are trained using only samples that are labelled *normal*. These methods learn a boundary surrounding the normal samples; if a test sample is found to lie outside the learnt boundary it is classified as an outlier. Most famous examples of such methods are: one-class SVMs [62] and one-class Kernel Fisher Discriminants [63, 64]. In the literature, there also exists work that uses neural networks to tackle the problem of one-class classification to detect anomalies, such as: program intrusion detection [65, 66], fraud detection [67] in mobile phone networks, detecting anomalies in jet engine vibrations [68].
There are two major problems with the supervised outlier detection approach. First, the presence of much more normal samples than the outlier samples; this imbalance in the class distributions creates issues, that are tackled in the machine learning community [69–74]. Second, obtaining accurate labels for the outlier class is usually a difficult problem.

2.1.2 Semi-Supervised

Semi-supervised outlier detection based methods rely on fitting a model on normal data instances to learn a notion of *normality*. Then, test instances are compared against the model to detect if they are normal or anomalous. As we have discussed in the previous supervised subsection, some of supervised machine learning methods can be adapted to work in a semi-supervised manner by only using the normal data samples for training.

One popular category of methods that are distinct in the semi-supervised setting are statistical anomaly detection methods, where probability density models are fitted to the *normal* data; then, the reciprocal of the likelihood of a test sample is used as its anomaly score. Statistical outlier detection methods act on the following assumption:

Assumption: *Normal samples lie in high probability regions of a statistical model, while outliers lie in low probability regions of the statistical model.*

Statistical outlier detection methods come in two different configurations: parametric, and non-parametric statistical models. Both have been applied to detect outliers. Parametric methods make assumption about the distribution of the data, thus estimate the parameters of the assumed distribution. In contrast, non-parametric methods make no assumption about the distribution of the data.

Parametric Statistical Methods

Parametric methods assume that *normal* instances of the data are generated by a parametric distribution with probability density function $f(\mathbf{x}, \Theta)$ for each instance $\mathbf{x}$, parametrized by parameters Θ. The inverse of the probability density function $f(\mathbf{x}, \Theta)$ of instance $\mathbf{x}$ is used as the anomaly score. Parametric statistical outlier detection methods can be divided into two categories: 1) Single Parametric Distribution, 2) Mixture of Parametric Distributions.

1. (*Single Parametric Distribution*) They mainly assume that the data is generated from a Gaussian distribution. The parameters of location (mean) and scatter (standard deviation) are estimated by finding the maximum of the log likelihood of the data. The anomaly score is the distance of a sample from the estimated mean. Then samples that are greater than a threshold are considered to be outliers. Depending on the type of distribution assumed different measure of distances to the mean are used.

   For multivariate datasets that assume a Gaussian distribution, the Mahalanobis distance of a test sample $\mathbf{x}$ to the estimated mean $\boldsymbol{\mu}$ is used as a measure of distance [75].

   Datasets that assume a t-distribution, the t-test is used to detect outliers. In this test a normal sample is compared against a test sample using the t-test; if the test measures a significant difference between the normal and test sample,

then the test sample is labelled as an outlier. A multivariate form of the t-test is the *Hotteling* t^2-test [76], which has been used to detect outliers in several bioavailability studies.
The use of a χ^2 statistic has been used in the cyber security literature by [77] to detect malicious attacks on operating systems. The normal data here is assumed to be generated from a multivariate Gaussian distribution. The χ^2 statistic is expressed as follows:

$$\chi^2 = \sum_{i=1}^{p} \frac{(x_i - \mu_i)^2}{\mu_i}.$$

Where x_i is the value of the i^{th} element of the test sample $\mathbf{x}$, μ_i is the i^{th} element of estimated mean vector $\boldsymbol{\mu}$ from the normal dataset, and p is the number of variables. If the χ^2 is greater than a threshold, then the observed sample $\mathbf{x}$ is an outlier.

2. (*Mixture of Parametric Distributions*) This type of outlier detection methods model the *normal* data as a mixture of parametric distributions. Then a test sample $\mathbf{x}$ is considered to be an outlier if its likelihood $f(\mathbf{x}, \Theta)$ is lower than a specified threshold.
Mostly Gaussian mixture methods have been used for such category of techniques [78]. They have been used in many applications such as: air-frame strain detection [79,80], detecting masses in mammograms [81,82], detecting intrusion in networks [83,84], and detecting anomalies in biomedical signals [85]. Moreover, authors in [86] used a mixture of Poisson distributions to model the normal data.

Non-Parametric Statistical Methods

Outlier detection methods in this category model the data using non-parametric statistical models. This type of models do not assume a specific distribution for the probability density function of the data. However, the density of the data is estimated from the available data itself. There are mainly two subcategories for this group of methods: 1) histogram based, and 2) kernel function based.

1. (*Histogram based*) This type of methods simply model the density of the normal data using histograms. For multivariate datasets a histogram is constructed for each feature of the normal data. Then to test for the anomaly of a test sample, an anomaly score is measured for each feature of the test sample. It is measured as the inverse of the height of value of the feature for the specific instance that is being tested. Then the anomaly score of each feature are accumulated to form an anomaly score for the specific test instance. This type of method has been used in practice to detect anomalies in many applications such as: detecting

structural damages [87–89], detecting anomalous topics in text data [90], web attack detection [91, 92], detecting network intrusions [83, 84, 93], and fraud detection [94].

2. (*Kernel Function Based*) These methods use kernel functions to approximate the probability density of the normal data. Samples that lie in the low likelihood region of the estimated density are considered outliers [95]. Such non-parametric methods have been used to detect network intrusions [96], masses in mammograms [82], and novelties in oil flow [97].

Summary

Semi-supervised outlier detection methods can only be used in applications where normal data is reliable and available. Parametric statistical outlier detection methods can either fit a single distribution or mixture distributions. Single distribution methods are the linear in computational complexity with respect to the number of samples and the number of features. Mixture of distribution methods use an iterative estimation algorithm: the Expectation Maximisation (EM) algorithm, which has computational complexity linear in time per iteration of the EM. However, it could have slow convergence as it depends on the problem and the convergence criterion. Non-parametric methods suffer much more from the curse of dimensionality compared to the parametric methods, as they make less assumption about the data. Also, they need many more samples than dimensions to estimate accurately the density of the normal data. Furthermore, kernel density methods can have quadratic computational complexity with regard to the number of samples in the data.

2.1.3 Unsupervised

Unsupervised outlier detection methods are more popular as they do not need labelled samples as the supervised outlier detection methods. They can be divided into the following categories: nearest-neighbour based, clustering based, dimensionality reduction based, and robust statistics based.

Nearest-Neighbour Based Outlier Detection

These methods act based on a specific assumption.

Assumption: *Normal samples lie close to each other in dense regions, while outliers lie far apart from their nearest neighbour.*

For nearest-neighbour based methods a similarity metric needs to be defined to measure the distance between any two data samples. For continuous datasets mainly the Euclidean distance is used as a similarity metric but other metrics can be used [98]. In the literature the k-nearest neighbour methods are widely used to detect outlier

samples in the data. The k-nearest neighbour methods define an anomaly score to each sample as its distance to its k^{th} neighbour. These methods have been applied in different domains to detect outliers, such as [86] to detect land mine from satellite images and anomalies in DC field windings of large turbines [99]. Authors in [99] used a threshold on the anomaly score to assign outliers. On the other hand one could choose the n samples with largest anomaly score to be the outliers [100].

Clustering Based Outlier Detection

Clustering is a data mining and machine learning tool used to group similar data samples into clusters [98, 101]. Although outlier detection and clustering seem to be fundamentally different problems, many clustering based outlier detection techniques can be found in the machine learning literature. Clustering based outlier detection techniques can be divided into three different categories.

1. First category, is based on the following assumption:
   Assumption: *Normal data samples are able to be clustered, while outliers can not be clustered.*
   Techniques belonging to this category apply clustering algorithms to the data, and any samples that do not belong to any cluster are considered to be outliers. This is achieved by clustering techniques that do not force every data sample into a cluster. Examples of such methods are: DBSCAN [102], ROCK [103], and SNN clustering [104]. Disadvantage of clustering methods following the above mentioned assumption is that they are optimized to cluster the data; thus, they are not optimal in finding anomalies.

2. Second category relies on the following assumption:
   Assumption: *Normal data samples are close to the nearest cluster centroid, while outliers are far away from the nearest cluster centroid.*
   Clustering methods that follow this assumption detect outliers by using a two-step procedure. The first step consists in applying a clustering algorithm to find cluster centroids. The second step will measure the distance of each data sample to its nearest cluster centroid; this is considered the anomaly score of each data sample. Smith et al. in [105] studied popular clustering algorithms, such as K-means clustering, Self organising maps (SOM), and Expectation Maximization to detect anomalies, following the previously described two step procedure.
   Methods that follow this particular assumption can be used in a semi-supervised setting, where the training dataset is clustered and then each test sample is compared against the clusters to get an anomaly score [90, 106, 107].
   The problem with this category of clustering based outlier detection methods is that they will fails to detect outliers that are clustered together. This brings us to the third category.

3. Third category assumes the following:
   Assumption: *Normal data samples belong to large clusters, while outliers belong to small clusters*
   This category of clustering based methods works in the following way. A clustering algorithm is applied to the data to detect clusters. The samples that belong to the minority cluster are considered to be the outliers. Several forms of this category have been developed [108–110]; they take into consideration both the distance of a sample from the centroid of the cluster it belongs to and the size of the cluster. The problem with these methods is that they are not optimized to detect outliers as they are mainly focusing on clustering the data.

Robust Statistics Based Outlier Detection

Robust statistics methods are designed to have high *breakdown value*. The definition of the *breakdown value* in robust statistics is the smallest proportion of the samples in the dataset that need to be corrupted to skew the estimate considerably. Thus, if an estimation method has a high breakdown value then it is more robust against outliers in the data. Robust statistics methods can be either univariate or multivariate.
The most common univariate methods of estimating location and scatter are: the median and the Median Absolute Deviation (MAD). Although they both act in the univariate case, they can still be used to detect outliers in multivariate datasets. Authors in [35] used both the Boxplot method which takes the median as a robust estimate of location, and MAD to detect outliers in genomic datasets. A sample is considered anomalous when it has more than a pre-defined number of outlier features. Multivariate robust statistics methods seek to robustly estimate the location vector and the scatter (covariance) matrix. There are several methods, such as the Minimum Covariance Determinant (MCD) [111, 112], M-estimator [36], S-estimator [113], Ellipsoidal peeling [114], and iterative deletion [115]. However, these methods suffer from the pitfall that their *breakdown value* is inversely proportional to the dimensionality of the data, making them unusable for high-dimensional datasets.

Dimensionality Reduction Based Outlier Detection

Dimensionality reduction methods are based on compressing the data from its original space to a reduced subspace where the structure of the data is expressed. Both linear and non-linear dimensionality reduction methods exist. Linear methods capture the highest variability in the data, and non-linear methods

capture the non-linear manifold structure of the data. Dimensionality reduction outlier detection methods can be used in two different modalities. The first modality is based on finding a lower dimensional embedding/projection of the data where the outlier and normal samples are separated. The second modality is based on using the reconstruction error of the data samples as the anomaly score, here the assumption is that outlier samples will have higher reconstruction errors than the normal samples.
An example of the first modality is found in [116]; it uses both linear and non-linear dimensionality reduction methods to detect anomalies in transportation corridors. Many other techniques use PCA to project the data into a lower dimensional subspace. An example of such kind of techniques is [117], where the authors analyze the projection of the data onto the low variance principal components. Such method has been used to detect outliers in astronomical catalogues [118].
For the second modality, methods such as PCA, Autoencoders(AE) and Variational Autoencoders have been used to detect outliers. PCA has been used to detect intrusions in computer networks [119]. Both authors of [120, 121] have studied AEs to detect outliers, and have shown that outliers have higher reconstruction error than the normal samples. VAEs have been studied by [122] to detect outliers, they used as an anomaly score the reconstruction probability as opposed to the reconstruction error. This is due to the fact that VAEs optimize a probabilistic objective function as opposed to the Euclidean reconstruction error, as AE and PCA. However, dimensionality reduction outlier detection methods used in the second modality are often semi-supervised; where normal samples are only used in a training set so that the learned model can detect anomalies more effectively. They are used in the semi-supervised setting, because often the applied dimensionality reduction methods are fragile to even a small number of outliers. Thus, the learned subspace is skewed towards the outliers giving them a misleading reconstruction error that makes them indistinguishable to the reconstruction error of normal samples. Research has been done in the robust statistics community to develop PCA that is robust to outliers. These robust PCA methods follow one of two categories: 1) Apply classical PCA to a robust estimate of the covariance matrix, or 2) look for directions that maximise a robust estimate of scale of the projected samples. Robust estimates of the covariance matrix have high *breakdown values* when the dimensionality of the data increases, as discussed in the previous subsection. Methods of the second category *breakdown values* are not affected by high dimensionality, such as Projection Pursuit (PP) [40]; however, they are non-convex and are combinatorial in their computational complexity, making them intractable when the dimensionality of the data increases.

One efficient way to solve this problem was introduced by Wright et al. in [1], which formulated the low rank matrix recovery problem and dubbed it '*Robust PCA*' (RPCA). Robust PCA will be described after introducing linear inverse problems with arbitrary corruption and rank minimization problems, as it stems from these two lines of work.

2.1.4 Linear Inverse Problems

Linear inverse problems arise in many applications such as signal and image processing, astrophysics, optics and statistical inference, just to name few. The linear inverse problem lets us study the discrete linear system in the form:

$$A\mathbf{x} = \mathbf{b} + \mathbf{w}, \tag{2.1}$$

were $A \in \mathbb{R}^{m\,x\,n}$ and $\mathbf{b} \in \mathbb{R}^m$ are known, $\mathbf{w} \in \mathbb{R}^m$ is an unknown noise and $\mathbf{x} \in \mathbb{R}^n$ is an unknown signal or image that needs to be estimated [123]. The standard approach to problem 2.1 is the least squares (LS) approach where the estimator $\hat{\mathbf{x}}$ is the one that minimizes the error:

$$\text{(LS):} \quad \hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}||A\mathbf{x} - \mathbf{b}||_2^2. \tag{2.2}$$

When A is a square matrix and non-singular, the LS estimator is the naive solution given by $A^{-1}\mathbf{b}$. However, in the case where the solution is not so straight forward, it is stated in the form

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b},$$

where $(A^T A^{-1})A^T$ is the pseudo-inverse of A. in many cases it happens that the matrix A is ill-conditioned, meaning that A will have a very large ratio between the largest and smallest singular value. This will lead to amplifying any errors in the target $\mathbf{b}$, leading to poor estimation of the weight vector $\mathbf{x}$. To solve this problem regularization methods are used to get a more stable solution. The main idea of regularization is to replace the ill-conditioned problem with a better conditioned problem, which has a solution that approximates the required solution. There are different regularization techniques, one of them being Tikhonov regularization [124] in which a quadratic penalty is added to the objective function in 2.2:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} ||A\mathbf{x} - \mathbf{b}||_2^2 + \lambda||L\mathbf{x}||_2^2. \tag{2.3}$$

The second term added in 2.3 is the regularization term which controls the l_2 norm of the estimate $\mathbf{x}$. The regularization parameter $\lambda > 0$ sets the trade-off between finding the required solution of 2.2 and having a small norm of the solution. L

is commonly chosen as the identity matrix, or the first or second order derivative operator [125, 126]. Although 2.3 has a closed form solution because the second term is a quadratic penalty which is differentiable, it does not give sparse solutions.
Another very important and widely used regularization method is the l_1 regularization method. Problem 2.2 is modified as:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \, ||A\mathbf{x} - \mathbf{b}||_2^2 + \lambda ||\mathbf{x}||_1, \tag{2.4}$$

where $||\mathbf{x}||_1$ stands for the sum of the absolute values of the elements of $\mathbf{x}$. The l_1 norm regularization gathered much importance, because it promotes sparsity in the solution of 2.4 as it is the tightest convex surrogate of the l_0 norm (l_0 norm being the number of zero elements). Problem 2.2 with the l_0 norm regularization term is highly non-convex, which may lead to finding suboptimal solutions; therefore, its convex surrogate (l_1 norm) is used instead to solve this issue. The objective function in problem 2.4 is not smooth; thus, not differentiable as in the case of 2.1 and 2.2. However, it is a convex optimization problem that can be easily solved, with many algorithms in hand [127,128]. In the following subsection we will introduce the general form of rank minimization problems and the famous matrix completion, which is a special case of these kind of problems.

2.1.5 Rank Minimization Problems

Let $X \in \mathbb{R}^{p \times n}$ be in the space of $p \times n$ matrices. The affine rank minimization problem consists of finding a matrix of minimum rank that needs to satisfy a system of linear equality constraints; the optimization problem is given as:

$$\min_X \big\{ \operatorname{rank}(X) : \mathcal{A}(X) = \mathbf{b} \big\}, \tag{2.5}$$

where $\mathcal{A} : \mathbb{R}^{pxn} \to \mathbb{R}^m$ is a linear map and $\mathbf{b} \in \mathbb{R}^m$. This problem in the form of 2.5 has been investigated before, in the machine learning field [129, 130]. Problem 2.5) is a non-convex and computationally intractable optimization problem. However, a convex relaxation of the rank function has been found to be the nuclear norm [131]. Now, the problem can be defined as a convex problem:

$$\min_X \big\{ ||X||_* : \mathcal{A}(X) = \mathbf{b} \big\}, \tag{2.6}$$

where the nuclear norm of X, $||X||_*$ is the sum of all singular values of X. A well known problem that has its general form as in 2.6, is the matrix completion problem. In the matrix completion problem, a random subset of matrix entries are given, and its aim is to recover the missing entries, making sure that the recovered matrix has

the lowest possible rank. It is written as:

$$\min_X \big\{ ||X||_* : X_{i,j} = M_{i,j}, (i,j) \in \Omega \big\}, \tag{2.7}$$

where M is the matrix with m available entries and Ω is the set of pairs of indices of size m. Here $\mathcal{A}(X) = X_\Omega$ and $\mathbf{b} = M_\Omega$, where X_Ω is a vector in $\mathbb{R}^{|\Omega|}$ obtained by selecting the elements of X whose indices are in Ω, same applies for M_Ω.
The equality constraint in problem 2.6 is more of an ideal scenario; in practice there is a noise term that is added to the observed variable $\mathbf{b}$, as $\mathcal{A}(X) = \mathbf{b} + \mathbf{w}$, $\mathbf{w}$ being the noise term. Notice how this is similar to the linear inverse problem, if X is restricted to be diagonal the equality constraint becomes $A\mathbf{x} = \mathbf{b} + \mathbf{w}$ (introduced in the previous subsection). To solve problem 2.6 with the noise term, we need to minimize the error $\mathcal{A}(X) - \mathbf{b}$. By relaxing the equality constraint in 2.6 we can have a formulation that is more resistant to noise, this leads to the nuclear norm regularized linear least squares problem [132]:

$$\min_X \big\{ ||\mathcal{A}(X) - \mathbf{b}||_2^2 + \lambda ||X||_* \big\}, \tag{2.8}$$

where $\lambda > 0$ is a positive parameter. Also notice that if X in 2.8 is restricted to be diagonal the problem will be same as l_1 regularized linear least square (problem 2.4). In the next subsection we will introduce the concept of Robust PCA that takes into account the concepts introduced in this subsection and subsection 2.1.4.

2.1.6 Robust Principal component Analysis (RPCA)

Robust PCA introduced by Candés et al. in [2], assumes that a data matrix $M \in \mathbb{R}^{p \times n}$ is generated by a low rank matrix $L \in \mathbb{R}^{p \times n}$, by corrupting some of its entries. This corruption is represented by adding to L a corruption matrix $E \in \mathbb{R}^{p \times n}$, which is sparse with non-zero entries that are of large values. The data matrix M is represented by the following decomposition:

$$M = L + E, \tag{2.9}$$

Notice that this case is different from the classical PCA case where the corruption matrix is dense with entries that are sampled from a Gaussian distribution.
The robust PCA problem consists in recovering L, given the data matrix $M = L + E$, where L and E are unknowns, and L is known to be low rank and E is known to be sparse. Conceptually meaning that we wish to recover the lowest rank matrix L that may have generated the data, subject to satisfying the constraint that the errors are

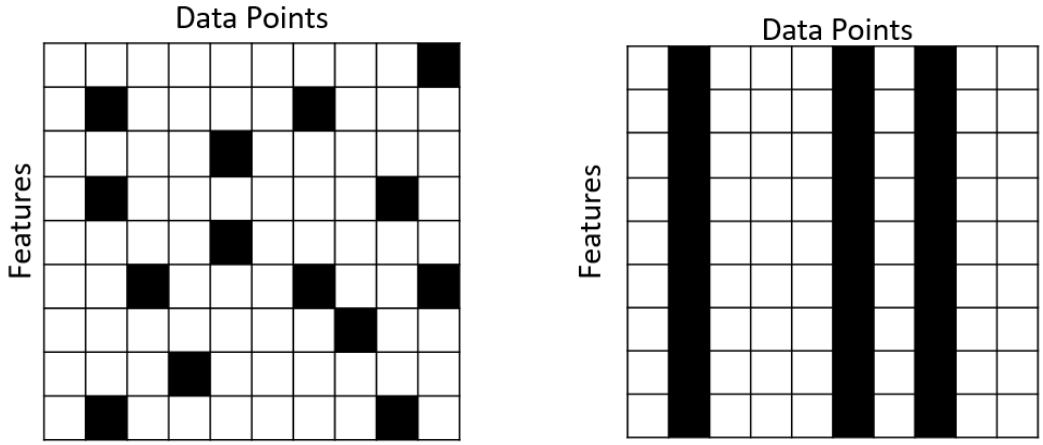
sparse $||E||_0 \leq k$. The robust PCA problem can be formulated as:

$$\min_{L,E} \quad \text{rank}(L) + \lambda||E||_0 \quad \text{subject to} : M = L + E \tag{2.10}$$

if problem 2.10 could be solved for an appropriate λ, we can hope to exactly recover the pair L_0, E_0 that generated the data M [1,2]. Unfortunately, the objective function of problem 2.10 is non-convex and its solution is not computationally feasible. However, from the previous subsection we know that there is a convex relaxation for the rank function; which is the count of the number of non-zero singular values of L; by using the nuclear norm which is the sum of the singular values of L. Moreover, there the convex relaxation of the l_0 norm (the l_0 norm is the count of the non-zero elements of E) is the l_1 norm, which is the sum of all the elements of E. Thus, the convex relaxation of problem 2.10 can be written as :

$$\min_{L,E} ||L||_* + \lambda||E||_1 \quad \text{subject to} : M = L + E \tag{2.11}$$

In this case $||E||_1$ is the sum of the absolute values of all the elements of E.
In [1, 2, 59] it has been proved that the pair L_0, E_0 can be exactly recovered under very broad conditions. These conditions are mainly two: 1) the left and right singular vectors of L should not be aligned with $\mathbf{e_i}$, the basis vectors in Euclidean space (the vector that has all entries equal to 0 except the i^{th} equal to 1). As an example, suppose that M is a rank-1 matrix constructed as $M = \boldsymbol{e_1 e_n^T}$. This will result in a matrix that has all entries as zeros except entry $(1, n)$ will have a value of one. In this case matrix M is both low-rank and sparse making its decomposition not feasible. To make the problem more significant, Wright et al. in [59] imposed that the low-rank matrix L needs not to be sparse. To impose this structure on the low-rank matrix, Wright et al. in [59] introduced the *random orthogonal model*. This considers the left are right singular vectors of the low rank matrix L as being selected uniformly at random, among all sets of r orthonormal vectors (r being the rank of L). 2) The non-zero entries of the error matrix E should be uniformly scattered through the whole matrix. In [1, 2], Wright et al. and Candès et al. proved that if the previous two conditions are satisfied one can recover the exact pair (L_0, E_0) which generated a data matrix M, as $M = L_0 + E_0$, by solving the convex optimization problem 2.11. The formulation of robust PCA in 2.11 has been widely used in many applications [2], such as video surveillance to identify activities that stand out from the background, face recognition, and collaborative filtering problems, just to name a few. Mainly in these type of applications the exact recovery of the low rank and the sparse matrix is crucial, therefore the robust PCA in 2.11) offers exact recovery and scalable optimization algorithms to find the solution.
The corruption model of E in robust PCA (2.9) is not of interest, as our purpose is



(a) (b)

Figure 2.1: Two different corruption models. **(a):** Robust PCA corruption model of Wright et al. [1] and Candès et al. [2], where the corruption matrix E is a sparse matrix with gross non-zero entries with indices chosen uniformly at random. This leads to sparse corruptions, which will have many data points with few features corrupted. **(b):** Represents the corruption model of Outlier Pursuit of [3], where the corruption matrix C is a column sparse matrix which will switch-off entire columns. The shown corruption model has a small fraction of outlier data points with features entirely corrupted. (black entries considered to be large numbers and white entries as zeros.)

to detect outlier samples. This is because having E as a sparse matrix with indices of the non-zeros entries being uniformly distributed through the matrix, means that all sample points will have some corrupted components; thus, making it not optimal for detecting outlier samples. This corruption model fits more the corruptions found in images, such as image occlusions. For the genomic datasets that we investigate in this thesis our purpose is to detect outlier samples. Therefore, a different corruption model needs to be analyzed, where most of the sample points have no corrupted components, and a few samples have most or all components corrupted. As such, we will focus on the column-wise corruption model, as it is more suited to detect outlying samples. Robust PCA with this type of corruption model has been investigated in [3] and has been called Robust PCA via Outlier Pursuit (OP). It will be introduced in the next subsection. An example of the corruption model of Robust PCA and Outlier Pursuit is shown in Figure 2.1(a) and Figure 2.1 (b) respectively.

2.1.7 Outlier Pursuit (OP)

Outlier Pursuit, has been introduced by Xu et al. in [3]. Recalling that the columns of the data matrix M represent samples and its rows represent dimensions, the aim

of Outlier Pursuit is to decompose M as:

$$M = L + C,$$

where L is low rank, and C is non-zero in only a small fraction of the columns, satisfying the corruption model shown in Figure 2.1(b). Outlier Pursuit can be written in the form of a optimization problem as simply as :

$$\min_{L,C} \text{rank}(L) + \lambda ||C||_{0,c} \quad \text{subject to} : M = L + C, \tag{2.12}$$

where $||C||_{0,c}$ stands for the number of non-zero columns of C. The non-zero columns are corrupted in most or all of their dimensions. However the objective function of problem 2.12 is non-convex and can not be solved efficiently. Therefore Outlier Pursuit solves the convex relaxation of 2.12, written as:

$$\min_{L,C} ||L||_* + \lambda ||C||_{1,2} \quad \text{subject to} : M = L + C, \tag{2.13}$$

where $||C||_{1,2}$ stands for the sum of the l_2 norms of the columns of C; $||C||_{1,2} = \sum_i ||\boldsymbol{C}_{:,i}||_2$.
The objective of Outlier Pursuit is to recover the low rank matrix L and the column sparse matrix C.
However, in a realistic case the samples will not lie exactly on a low-dimensional subspace as they will be corrupted by noise; so we need to consider the case where the decomposition is $M = L + C + N$, where N is an additional noise matrix. We can adapt optimization problem 2.13 to accommodate for approximate solutions, by replacing the equality constraint with a more relaxed constraint. This is written in an optimization form as:

$$\min_{L,E} ||L||_* + \lambda ||C||_{1,2} \quad \text{subject to} : ||M - (L + C)||_F \leq k. \tag{2.14}$$

As in the case of rank minimization with noise, one can solve for an approximate solution by including a regularizing parameter μ to problem 2.14. This becomes an unconstrained optimization problem as:

$$\min_{L,C} \quad \mu ||L||_* + \mu\lambda ||C||_{1,2} + \frac{1}{2}||M - (L + C)||_F^2. \tag{2.15}$$

The form of all the problems 2.4, 2.8, and 2.15 is convex and can be solved using proximal gradient methods [133].
The aim of OP is to solve for a low rank matrix approximation of the input data that is robust to outliers. This is achieved by modelling the outliers in the matrix C. The

benefits of OP are:

1. The low rank matrices, thus the low-dimensional representation of the data is not affected by the outliers, hence making it robust to outliers.

2. The outliers are detected, which can be either discarded or can be of interest to the application itself.

2.2 Graph Regularized Low Rank Approximation Methods: Review

The problem of incorporating the non-linear topological structure of the data in classical PCA is achieved by graph regularization. State of the art graph regularized PCA models can be divided into two different models:

- Factorized model: graph regularization is on factors.

- Non-factorized model: graph regularization is on low rank matrix.

All the methods we are reviewing below use a graph regularization with respect to a graph of samples. If the data matrix is $X \in \mathbb{R}^{p \times n}$ then the graph we regularize the model with is the Laplacian matrix $\Phi \in \mathbb{R}^{n \times n}$ of the k-nearest neighbour graph, constructed between the n samples of the data matrix X. Before reviewing state of the art graph regularized PCA methods, in the next subsection we will describe how to construct and compute the graph Laplacian Φ.

2.2.1 Graph Construction

To model the topological structure of the data the localities of each sample need to be modelled. This is achieved by using a k-nearest neighbour graph [42, 58, 134]. The graph which has nodes corresponding to samples, is constructed by first finding the k-nearest neighbours of each sample. Then for each sample we weight the edges to its k neighbours through the Gaussian kernel function $W_{i,j} = \exp(-\frac{||\boldsymbol{M}_{:,i} - \boldsymbol{M}_{:,j}||_2^2}{2\sigma^2})$. All other points that are not in the k-nearest neighbours of the sample are weighted as zero. The matrix that incorporates this information is the affinity matrix $W \in \mathbb{R}^{n \times n}$. Then the graph Laplacian matrix $\Phi \in \mathbb{R}^{n \times n}$ is defined by $\Phi = D - W$, where D is a diagonal matrix where each entry on its diagonal is the row sum of the corresponding row in W, $D_{i,i} = \sum_j W_{i,j}$.

Proposition 1 The graph Laplacian matrix Φ is symmetric positive semi-definite. (*Proof*)

The symmetry of Φ follows from the symmetry of W and D. Φ is positive semi-definite if it satisfies $\boldsymbol{z}^T\Phi\boldsymbol{z} \geq 0$ for any vector $\boldsymbol{z} \in \mathbb{R}^n$. We can rewrite $\boldsymbol{z}^T\Phi\boldsymbol{z}$ as $\frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}(z_i - z_j)^2$:

$$\begin{aligned}\boldsymbol{z}^T\Phi\boldsymbol{z} &= \boldsymbol{z}^T D\boldsymbol{z} - \boldsymbol{z}^T W\boldsymbol{z} = \sum_{i=1}^{n} D_{i,i}z_i^2 - \sum_{i,j=1}^{n} W_{i,j}z_i z_j \\ &= \frac{1}{2}\Big(\sum_{i=1}^{n} D_{i,i}z_{i,i}^2 - 2\sum_{i,j=1}^{n} W_{i,j}z_i z_j + \sum_{j=1}^{n} D_{j,j}z_{j,j}^2\Big) = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}(z_i - z_j)^2.\end{aligned} \tag{2.16}$$

From the last expression in equation 2.16 and the way $W_{i,j}$ is constructed we can see that $\boldsymbol{z}^T\Phi\boldsymbol{z} \geq 0$ is satisfied.

2.2.2 Factorized Models

Factorized models have the same form of classical PCA. In these type of models the low rank matrix approximation of the data matrix is determined by a matrix factorization. The data matrix $X \in \mathbb{R}^{p\times n}$ is factorized to $X = UQ^T$, where $U \in \mathbb{R}^{p\times r}$ and $Q \in \mathbb{R}^{n\times r}$, given that $r << p$. These methods have a graph regularization on the low-dimensional embedding of the data [57, 134, 135].

Graph Laplacian PCA (GLPCA)

Graph Laplacian PCA (GLPA) [57] solves the following optimization problem:

$$\begin{aligned}&\min_{U,Q} ||X - UQ^T||_F^2 + \alpha\,\mathrm{tr}(Q^T\Phi Q) \\ &\text{subject to: } Q^TQ = I,\end{aligned} \tag{2.17}$$

where $\mathrm{tr}(Y)$ is the sum all the diagonal elements of a square matrix Y. The graph regularization is on Q, the principal components of the data matrix X. This problem has a closed-form solution, where the columns of Q are computed by eigenvectors of the generalized Laplacian matrix: $X^TX + \alpha\Phi$. In addition, since Q is an orthonormal matrix, the optimal column basis vectors U can be computed by $U = XQ$. Note that problem 2.17 is non-convex with respect to U and Q, due to the orthonormality constraints and the product of the matrix factors: UQ^T. Nonetheless, it has a unique solution.

Robust Graph Laplacian PCA (RGLPCA)

GLPCA is sensitive to outlying samples. Therefore, the authors of [57] also devised another version of GLPCA that is robust to outlying samples. The outlier robust

problem is

$$\begin{aligned} &\min_{U,Q} ||X - UQ^T||_{1,2} + \alpha \operatorname{tr}(Q^T \Phi Q) \\ &\text{subject to: } Q^T Q = I. \end{aligned} \tag{2.18}$$

The main difference between 2.17 and 2.18 is the reconstruction error term of the objective function. The reconstruction term in 2.18 is more robust to outlier samples as it does not square the error of the individual data sample. The problem with RGLPCA is that it does not have a closed form solution as GLPCA. The solution to the RGLPCA problem is found by an iterative update method. Therefore, due to the non-convexity of the problem, the solution will be a local optimum of the objective function. 2.18.

Manifold Matrix Factorization (MMF)

The method devised by Zhang et al. [134] solves the following optimization problem:

$$\begin{aligned} &\min_{U,Q} ||X - UQ^T||_F^2 + \alpha \operatorname{tr}(Q^T \Phi Q) \\ &\text{subject to: } U^T U = I, \end{aligned} \tag{2.19}$$

this problem resembles more the classical PCA problem as it has the same orthonormality constraint on the basis matrix U. Similar to PCA, this problem is also non-convex, however is solved using a alternate iterative method.

Multiple Manifold Matrix Factorization

A direct modification to MMF described above is proposed by authors in [135], which use an ensemble of graph regularization terms. This method is called multiple manifold matrix factorization (MMMF); it solves the following optimization problem:

$$\begin{aligned} &\min_{U,Q,\boldsymbol{\alpha}} ||X - UQ^T||_F^2 + \operatorname{tr}(Q^T(\sum_i \alpha_i \Phi_i)Q) \\ &\text{subject to: } U^T U = I, \sum_i \alpha_i = 1. \end{aligned} \tag{2.20}$$

Problem 2.20 takes into account multiple graphs constructed by using different parameters, or different methods; Φ_i represents the graph Laplacian of the i^{th} graph. This type of ensemble regularization makes the model more robust to noise in the data. Note that this method finds the vector containing the sparse linear combination coefficients $\boldsymbol{\alpha}$ of the graphs Laplacian matrices during the optimization process, which adds to the non-convexity of the problem. Furthermore, problem 2.20 is solved using an iterative solver, where at each iteration an MMF problem is to be solved, this

convergence to a local optimum solution and has higher computational complexity than MMF.

2.2.3 Non-Factorized Models

This type of models have a graph regularization on the low rank matrix approximation, in contrast to the factorized models that have a graph regularization on the principal components of the data.

Manifold Matrix Factorization

This model can fall under both categories of factorized and non-factorized by only a change of variables. Using the cyclic permutation invariance property of the trace the graph regularization term of problem 2.19 can be rewritten as:

$$\mathrm{tr}(Q^T \Phi Q) = \mathrm{tr}(U Q^T \Phi Q U^T) = \mathrm{tr}(L \Phi L^T)$$

.

Therefore problem 2.19 can be rewritten as:

$$\begin{aligned} &\min_{L} ||X - L||_F^2 + \alpha \,\mathrm{tr}(L \Phi L^T) \\ &\text{subject to: } \mathrm{rank}(L) \leq r, \end{aligned} \tag{2.21}$$

this problem is equivalent to the previous problem and it is convex with respect to the variable L and it has a unique closed form solution.

Robust PCA on graphs (RPCAG)

Robust PCA on graphs is introduced by [58]. The problem is formulated in the following way:

$$\min_{L,C} ||L||_* + \lambda ||C||_1 + \alpha \,\mathrm{tr}(L \Phi L^T) \quad \text{subject to: } M{=}L{+}C. \tag{2.22}$$

It has a convex objective function, therefore a global solution can be reached through standard iterative optimization methods. The difference between problem 2.22 and problem 2.21 is the model of the noise that is taken into account. In problem 2.21 the noise is modelled to be sampled from a Gaussian distribution, which makes it sensitive to outlier samples in the dataset. On the other hand, in the RPCAG model, like RPCA, the noise is modelled to be distributed by a Laplacian probability density function. Furthermore, in RPCAG the low rank matrix L is modelled to be smooth on a manifold described by the Laplacian matrix Φ.
Proof of convexity of the objective function of problem 2.22 is shown below.

Proposition 2 The function $f(L, C) = ||L||_* + ||C||_1 + \alpha \operatorname{tr}(L\Phi L^T)$ is convex w.r.t both L, C.

(**Definition**) For a function $f : \mathbb{R}^n \to \mathbb{R}$ to be convex, then for every $\boldsymbol{x}$, $\boldsymbol{y}$ and $0 \leq \lambda \leq 1$ the inequality

$$f\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \leq \lambda f\big(\boldsymbol{x}\big) + (1-\lambda) f\big(\boldsymbol{y}\big) \tag{2.23}$$

holds.

***Proof* of Proposition 2**

The first two functions of $f(L, C)$ are the nuclear norm and l_1 norm, $||L||_*$ and $||C||_1$. Vector norm functions have the following three properties:

1. $||\boldsymbol{x}|| = 0$ only is $\boldsymbol{x} = \boldsymbol{0}$.
2. $||\alpha \boldsymbol{x}|| = |\alpha| ||\boldsymbol{x}||$. for all $\alpha \in \mathbb{R}$.
3. $||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ (triangle inequality).

We use properties 2 and 3 for vector norms to prove that they are convex and satisfy inequality 2.23

$$||\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}|| \leq ||\lambda \boldsymbol{x}|| + ||(1-\lambda)\boldsymbol{y}|| = \lambda ||\boldsymbol{x}|| + (1-\lambda)||\boldsymbol{y}||. \tag{2.24}$$

Therefore, the nuclear norm and l_1 norm are convex functions.
The third function of $f(L, C)$ is $\operatorname{tr}(L\Phi L^T)$ it can be rewritten as $\frac{1}{2}\sum_{i,j=1}^{N} ||\boldsymbol{L}_{:,i} - \boldsymbol{L}_{:,j}||_2^2 W_{i,j}$ (refer to Chapter 4). The l_2 norm squared, $||\boldsymbol{L}_{:,i} - \boldsymbol{L}_{:,j}||_2^2$, is a strongly convex function and all $W_{i,j}$ are positive scalars. Therefore, $\frac{1}{2}\sum_{i,j=1}^{N} ||\boldsymbol{L}_{:,i} - \boldsymbol{L}_{:,j}||_2^2 W_{i,j}$ is a convex combination of convex functions, this gives also a convex function.
Definition $g(\boldsymbol{x}) = a_1 g_1(\boldsymbol{x}) + a_2 g_2(\boldsymbol{x})$ with $a_i \geq 0$. $g(\boldsymbol{x})$ is a convex combination of convex functions g_1 and $g_2 : \mathbb{R}^n \to \mathbb{R}$. Now, we prove that $g(\boldsymbol{x})$ is a convex function by showing that it satisfies inequality 2.23.

$$\begin{aligned}
&g\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) = a_1 g_1\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) + a_2 g_2\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \\
&\leq a_1 \lambda g_1(\boldsymbol{x}) + a_1(1-\lambda) g_1(\boldsymbol{y}) + a_2 \lambda g_2(\boldsymbol{x}) + a_2(1-\lambda) g_2(\boldsymbol{y}) \\
&= \lambda\big(a_1 g_1(\boldsymbol{x}) + a_2 g_2(\boldsymbol{x})\big) + (1-\lambda)\big(a_1 g_1(\boldsymbol{y}) + a_2 g_2(\boldsymbol{y})\big) \\
&= \lambda g(\boldsymbol{x}) + (1-\lambda) g(\boldsymbol{y}).
\end{aligned} \tag{2.25}$$

This shows that

$$g\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \leq \lambda g(\boldsymbol{x}) + (1-\lambda) g(\boldsymbol{y}).$$

Which concludes the proof.

Matrix Completion on Graphs

The authors of [136] proposed matrix completion on graphs. The objective function of this problem is as follows:

$$\min_{L} ||A \circ (M - L)||_F^2 + \lambda ||L||_* + \alpha_r \operatorname{tr}(L^T \Phi^r L) + \alpha_c \operatorname{tr}(L \Phi^c L^T), \tag{2.26}$$

where A is a binary mask matrix, with elements that are ones for the known entries and zeros for the missing entries. Note that this problem we want to recover the missing entries in L with graph Laplacian regularization. To model accurately the row and column structure of the data, the authors regularizes L by its column structure (Φ^c) and its row structure (Φ^r). Although problem 2.26 is a formulation of matrix completion, it can also be considered a low rank matrix approximation in the same essence of PCA; when A is the matrix of all ones.

2.3 Multi-view Clustering: Review

Multi-view clustering methods are widely available in the machine learning community. They can be divided into four different categories:

- Co-training based: it uses an alternate optimization framework, where an optimization step is taken in each iteration with respect to one of the views.
- Deep learning based: it adopts the knwon neural network architectures which have shown promising results in the deep learning research field.
- Early integration: it consists in concatenating the features of the different views into a single matrix, then a single-view clustering method is used onto the obtained matrix. It is the simplest of all three categories.
- Late integration: it consists in separately clustering each view using a single-view clustering algorithm, then the different clusters are integrated into one global clustering solution.
- Intermediate integration: it consists in building a model that integrates all views and can be subdivided into the following subcategories: i) Statistical modelling based. ii) Sample similarity based. iii) Joint dimensionality reduction based.

Multi-view clustering methods belonging to these different categories will be reviewed in the following subsections.

2.3.1 Co-Training

Co-training is the earliest form of multi-view learning that attempted to integrate two views [137]. Stemming from the improved accuracy that this method has shown on semi-supervised classification tasks, authors have been inspired to investigate co-training on multi-view data. One of the earliest methods that attempt to investigate multi-view clustering is co-EM introduced by [49]. This method takes the Expectation maximisation (EM) algorithm used for k-means clustering and adapts it to a multi-view setting. It also uses the iterative EM algorithm in an alternate framework, where an optimisation step is taken in each iteration for one of the views; then, on the next iteration, an optimisation step regarding a different view is taken. This methods was one of the first to show improvements in clustering each view alone and in clustering the matrix of concatenated views [46].

2.3.2 Deep Learning Based

In the past decade, deep learning methods have shown promising results in challenging tasks, such as image recognition [138] and text translation [139]. They have also been widely used for analysing medical data [140]. Neural networks designed for multi-view application have shown promising results [141]; especially for the unsupervised representation learning setting. These representation learning methods are used to cluster and discover potential novel structures in multi-view dataset settings [142, 143]. In the multi-*omic* dataset case, authors in [144] use an auto encoder to find representations of liver cancer samples from different patients. The multi-*omic* dataset consist of three views: gene expression, DNA methylation, and miRNA expression. The three different views of the data where concatenated beforehand; then the autoencoder was used to find the low-dimensional representation of the samples. Afterwards, the features of the low-dimensional representation, that are most correlated with the survival times of the patients, are selected to be used for standard clustering. This method showed significant difference in the survival times between clusters. Liang et al. in [145] used a Deep Beleif Networks (DBN) [146] to cluster ovarian cancer patients. The multi-*omic* dataset is comprised of three views: gene expression, DNA methylation and miRNA expression. This network architecture consists in having hidden layers from each view. Then, the three hidden layers are integrated by fully connecting with the final hidden layer. Subsequently, a binary 3-dimensional representation is learnt for all the samples, the $2^3 = 8$ different possible positions in this latent representative are used as cluster labels for 8 clusters. They showed improved clustering performance compared to k-means clustering on the concatenation of the three different views.

The main drawback of deep learning algorithms is that they are effective only with datasets that have a large sample size, and a large sample size compared to the num-

ber of dimensions. Deep learning algorithms have many parameters to be learned; not having enough sample size would result in overfitting of the training data and thus giving poor generalization. This is a major issue with the current multi-*omic* datasets which are low in sample size and have much more dimensions than samples.

2.3.3 Early Integration

Early integration methods concatenate different views into one single matrix; then use single-view clustering methods to obtain the clusters. Although this method is simple to implement, it suffers from several major drawbacks. First, the dimensionality of the concatenated matrix increases substantially and negatively affects the performance of the clustering algorithms. Second, the different views will be treated as one. This will ignore the structural diversities between views. Specific early integration methods have been designed in the machine leaning community to address these drawbacks.
LRACluster [54] is an early integration technique that uses a regularized probabilistic model. Each view is modelled to have a specific latent representation of its same size using a specific probability density function depending on the type of feature present in the view. Different types of modelled features are: real, count, binary using the Gaussian, Poisson and Bernoulli. The objective function of LRACluster is described as: $\text{argmin}_{\Theta} \sum_{v=1}^{K} \mathcal{L}(\Theta_v; M_v) + \mu||\Theta||_*$. Where $\mathcal{L}(\Theta_v; M_v)$ is the negative log-likelihood of the v^{th} view M_v, Θ is the concatenation of all the Θ_v matrices, μ is a regularization parameter, and $||\Theta||_*$ is the nuclear norm of Θ. The nuclear norm regularization of Θ will induce the low rank structure in the matrix Θ which will reduce the complexity of the probabilistic model. The LRACluster objective function is convex, thus a global optimal solution is achieved by using a simple gradient descent method.
Structured sparsity [147] is another multi-view clustering early integration method. The different views $M_v \in \mathbb{R}^{p_v \times n}$ are concatenated to form a matrix $X \in \mathbb{R}^{p \times n}$ ($p = \sum_v p_v$). The objective function of structured sparsity is: $\text{argmin}_{W,F} ||X^T W + 1_n \mathbf{b}^T - F||_F^2 + ||W||_{G_1}$. Where $\mathbf{b}$ is a $c \times 1$ intercept vector and 1_n is a $n \times 1$ vector for all ones. W is a $p \times c$ matrix which contains the weights of each feature for c different clusters. F is a $n \times c$ cluster indicator matrix satisfying the constraint: $F^T F = I$. The aim for this algorithm is to find the closest projection of the data to the cluster indicator matrix F. Then a sample is assigned to a specific cluster, by choosing the position of the largest entry in the i^{th} row of the cluster indicator matrix $F_{i:} \in \mathbb{R}^{1 \times c}$. The regularization term uses the G_1 norm on the weight matrix W. The G_1 norms of W is the sum of the l_2 norm of the weights of the features in each view summed over all clusters, $||W||_{G_1} = \sum_{i=1}^{c} \sum_{v=1}^{K} ||\boldsymbol{W}_v^{:,i}||_2$. This will induce a group sparsity. Thus, if features of a specific view in W do not discriminate a specific cluster, then the values of the features of that view for that specific cluster are assigned to low values.

2.3.4 Late Integration

Late integration methods cluster each view separately; then integrate the cluster solutions to form a single consensus clustering solution. Different single-view clustering algorithms can be used to cluster each view. This makes late integration methods versatile in choosing the best clustering algorithm that best fits the structure each view.

COCA [148] is a late integration method used to study how different tumor tissues can share the same genomic signatures. The study consisted in clustering 12 cancer types by using pan-cancer TCGA data. First, each view is clustered separately into c^v clusters, then the i^{th} sample of the v^{th} view $\boldsymbol{M}_v^{:,i}$ is encoded by a binary vector $\mathbf{p_i^v} \in \mathbb{R}^{c\times 1}$, where $\mathbf{p}_i^v(j) = 1$ when j corresponds to the cluster that the i^{th} sample is assigned to, otherwise the entries are 0. Then, for each sample the binary vectors from each view are concatenated together forming the binary matrix $B \in \mathbb{R}^{b\times n}$, where $b = \sum_{v=1}^{V} c^v$. The binary matrix B is then clustered using consensus clustering [149] which will give the final overall clustering assignment of the samples.

Late integration methods that use *soft* clustering other than *hard* cluster assignments have been studied. One such example is [150], where Probabilistic Latent Semantic Analysis (PLSA) [151] is used to generate *soft* cluster assignments from the consensus binary matrix B.

Another late integration method is PINS [152]. It takes into account a binary connectivity matrix for each view $C_v \in \mathbb{R}^{n\times n}$, where $C_v^{i,j} = 1$ when sample i ($\boldsymbol{M}_v^{:,i}$) and sample j ($\boldsymbol{M}_v^{:,j}$) are connected in the v^{th} view. Then the connectivity matrices of each view C_v are averaged together to form a single connectivity matrix representing an integration of the clustering assignments of each view. The integrated connectivity matrix is subsequently clustered using different methods chosen on the basis of how much they agree with each other.

2.3.5 Statistical Modelling Based

Statistical methods rely on modelling the probability distribution of the data. The most applied intermediate integration method that is based on statistical modeling is iCluster [153].

iCluster assumes that each view of the data M_v is generated from a shared low-dimensional latent representation $Z \in \mathbb{R}^{c\times n}$. Each view is modelled as $M_v = W_v Z + N_v$, where $W_v \in \mathbb{R}^{p_v\times c}$ is a view specific projection matrix, c is the number of clusters and N_v is a noise matrix which is normally distributed. iCluster maximises the negative log likelihood of the data with an additional sparsity inducing l_1 norm regularizer on the projection matrices. The optimisation problem is solved using the EM algorithm to solve for the W_v and Z. Then the shared low-dimensional latent representation Z is clustered using k-means clustering to obtain the final clustering

solution.

2.3.6 Similarity Based

Similarity based methods use sample similarities to cluster data. In this type of multi-view clustering methods the sample similarities are determined for each view separately then the similarities are integrated. Different methods in this category integrate the view specific similarities in different ways. Three main methods have been developed using this kind of approach: 1) Spectral clustering based methods. 2) Similarity Network Fusion (SNF) [154]. 3) regularized Multiple Kernel Learning Local Preserving Projections (rMKL-LPP) [155].

Spectral Clustering Based Methods

They are generalizations of the well known spectral clustering method [156]. Single-view spectral clustering optimizes the following objective function: $\text{argmax}_U \text{tr}(U^T \Phi U)$ s.t $U^T U = I$, where $\Phi \in \mathbb{R}^{n \times n}$ is the Laplacian matrix and $U \in \mathbb{R}^{n \times c}$ is a matrix of the low-dimensional representation of the samples, and c is the number of clusters. The solution for this optimization problem is the c largest eigenvectors of Φ which are concatenated to form the matrix U. Then matrix U is clustered using k-means clustering to find c different clusters.
Authors in [56] generalized the spectral clustering objective function to work on multiple views. They propose two different objective functions with different regularizations. The first regularizes the eigenvectors of each view to be similar to a consensus; the second regularizes them to be similar to each other. The first objective function is as follows: $\text{argmax}_{U^v \forall v, U^*} \sum_{v=1}^{V} \text{tr}(U_v^T \Phi_v U_v) + \sum_{v=1}^{V} \lambda_v \text{tr}(U_v U_v^T U^* U^{*T})$ $s.t$ $U_v^T U_v = I$ $\forall v; U^{*,T} U^* = I$. This objective function tries to balance the individual spectral clustering objective and the correlation of the eigenvectors of each view U^v with the consensus eigenvectors $U^* \in \mathbb{R}^{n \times c}$. The second objective function has the same spectral clustering objective as the first term but regularizes it as follows: $\sum_{\forall v \neq m} \text{tr}(U_v U_v^T U_m U_m^T)$ .This objective function regularizes the eigenvectors of each view to be correlated with each other.
Authors in [157], instead of separately finding the latent low-dimensional representation of each view (as [56]), they optimize a single shared latent space between all views. The objective function is: $\text{argmax}_U \sum_{v=1}^{V} \text{tr}(U^T \Phi_v U)$ $s.t$ $U^T U = I$. It is the same as applying spectral clustering on the sum of the Laplacian matrices of each view $\sum_{v=1}^{V} \Phi_v$. Then, the resulting clusters are improved by assigning samples to clusters in a greedy fashion while aiming to optimize the normalized cut objective function.

Similarity Network Fusion

SNF [154] is a similarity based multi-view clustering method. This method initially constructs a network for each *omic* where the nodes correspond to samples. Then, W_v the similarity of each network is computed. Afterwards, the iterative network fusion step updates each W_v using knowledge from other networks, with the aim to make them more similar to each other at each iterative step. At the end of the iterative procedure the networks are averaged together to get a final network. The final network is then clustered using spectral clustering.
In [154] SNF was used to detect cancer subtypes in five different TCGA cancer datasets, with each cancer type having three measured molecular features: mRNA expression, DNA methylation and miRNA expression.

rMKL-LPP

rMKL-LPP introduced in [155] takes the multiple kernel learning problem for data integration and augments it to reduce the dimensionality of the data. The multiple kernel learning problem integrates different kernel by learning the coefficients of their linear combination. Different kernels here are the sample similarity matrices of each view. rMKL-LPP reduces the dimensionality of the data into a shared representation that maintains the sample similarities of each view. Then, k-means clustering is applied in the learnt lower dimensional space to find sample partitions. In [155] rMKLLPP is used on five TCGA cancer datasets, and it shows that the obtained clusters have significantly different survival times.

2.3.7 Joint Dimensionality Reduction Based

Joint dimensionality reduction based multi-view methods assume that the views of the data are each generated from a shared low-dimensional representation. Thus, the aim of these methods is to find a shared latent representation between all the views. In doing so, the learnt representation would combine complementary information that is present within the different views. The general model of learning a shared latent representation is introduced by [158].
In [158] the low-dimensional projections $Z_v \in \mathbb{R}^{r_v \times n}$ ($r_v << p$, $\forall v$) of each view M_v are assumed to be given beforehand, and the aim of their work is to devise a general framework to find a shared low-dimensional representation $Z^* \in \mathbb{R}^{r \times n}$. The general framework is:

$$\underset{Z^*, \mathcal{F}}{\operatorname{argmin}} \sum_{v=1}^{V} \lambda_v l(Z_v, f_i(Z^*)).$$

Where $\{\lambda_v\}_{v=1}^{V}$ is a set of non-negative weights, l is a distance function, and $\mathcal{F} = \{f_v\}_1^V$ is a set of functions that map Z^* to Z_v. The aims to find an optimal shared

latent space Z^* and the mapping functions f_v. Then, multi-view clustering is achieved by using a single-view clustering method on the optimal shared subsapce Z^*. Authors in [158] derived an algorithm to optimize the aforementioned general framework when using the well known Euclidean distance function; giving the multi-view subspace learning framework:

$$\underset{Z^*,B}{\operatorname{argmin}} \sum_{v=1}^{V} \lambda_v ||Z_v - B_v Z^*||_F^2.$$

Where $\{B_v \in \mathbb{R}^{r_v \times r}\}_{v=1}^{V}$ is the set of linear transformations that transform the dimensionality of the low-dimensional representations of each view Z_v to the same space of the shared low-dimensional subspace.
Several multi-view subspace learning based methods have been introduced in the literature; we will review below the most relevant ones.

Canonical Correlation Analysis (CCA)

CCA is the earliest of multi-view subspace learning methods. It is regarded as the multi-view version of PCA. Given a dataset with two views; $M_1 \in \mathbb{R}^{p_1 \times n}$ and $M_2 \in \mathbb{R}^{p_2 \times n}$, the aim of CCA is to find two projection directions $\mathbf{w}_1 \in \mathbb{R}^{p_1}$ and $\mathbf{w}_2 \in \mathbb{R}^{p_2}$, so that the correlation of the projection of each view is maximised; the projection directions are called canonical transformations, and the projections are called canonical variates. The first canonical variate of the first view is $\mathbf{z}_1 = {M_1}^T \mathbf{w}_1$ and that of the second view is $\mathbf{z}_2 = {M_2}^T \mathbf{w}_2$. CCA seeks to maximise the correlation coefficient ρ of both view:

$$\rho = \frac{\mathbf{z}_1^T \mathbf{z}_2}{||\mathbf{z}_1||_2 ||\mathbf{z}_2||_2} = \frac{\mathbf{w}_1^T C_{12} \mathbf{w}_2}{\sqrt{(\mathbf{w}_1^T C_{11} \mathbf{w_1})(\mathbf{w}_2^T C_{22} \mathbf{w}_2)}},$$

where $C_{ij} = M_i {M_j}^T$ is the cross-covariance matrix between view i and j. $\mathbf{w}_1$ and $\mathbf{w}_2$ are the directions that give the maximal correlations between the views. Then, the k^{th} pair of projected directions $\mathbf{w}_1^k$ and $\mathbf{w}_2^k$ are found, so that the corresponding pair of canonical variates ${M_1}^T \mathbf{w}_1^k$ and ${M_2}^T \mathbf{w}_2^k$ are maximally correlated, given that $\mathbf{w}_1^k$ and $\mathbf{w}_2^k$ are orthogonal to $\mathbf{w}_1^{k-1}$ and $\mathbf{w}_2^{k-1}$ respectively. ρ is invariant to the scaling of the canonical transformations $\mathbf{w}_1$ and $\mathbf{w}_2$, thus they are normalised, with the result that the variance of the first canonical variates are equal to one. The CCA objective function can be written as:

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\operatorname{argmax}} \, \mathbf{w}_1^T C_{12} \mathbf{w}_2 \\ &\text{s.t} \quad \mathbf{w}_1^T C_{11} \mathbf{w}_1 = 1, \quad \mathbf{w}_2^T C_{22} \mathbf{w}_2 = 1. \end{aligned}$$

applying the Lagrange multiplier techniques on the previous problem one can solve for $\mathbf{w}_1$ and $\mathbf{w}_2$ by solving a generalized eigenvalue problem each. Solution for $\mathbf{w}_1$ is:

$$C_{12}(C_{22})^{-1}C_{21}\mathbf{w}_1 = \eta C_{11}\mathbf{w}_1. \tag{2.27}$$

where η is the eigenvalue corresponding to $\mathbf{w}_1$. Solution for $\mathbf{w}_2$ is:

$$C_{21}(C_{11})^{-1}C_{12}\mathbf{w}_2 = \eta C_{22}\mathbf{w}_2. \tag{2.28}$$

$\mathbf{w}_1$ and $\mathbf{w}_2$ are the first eigenvectors of problems 2.27 and 2.28, The steps to obtain the generalized eigenvalue problems 2.27,2.28 are shown in the Appendix. Note that this method works only if covariance matrices C_{11} and C_{22} are non-singular. This assumption will not be true if the multi-view data is high-dimensional.
In [55] an extension to CCA to multi-view clustering is introduced. Their work utilizes CCA to find a subspace that is spanned by the means of the mixture components generating the views. This subspace has a special property: when data is projected onto it, the means of the mixing distributions will be well separated, and the data within the same generating mixture will be closer than the original space. To this extent, in [55] the projected data onto the obtained subspace is clustered using a single-view clustering algorithm. Moreover, their algorithm for finding the subspace boils down to computing the top left singular vectors of the cross-covariance matrix C_{12}. Then, to find the shared latent representation, the data in the first view is projected onto the computed left singular vectors.
CCA can only take into account two views, extensions of CCA have been formulated to take into account multiple views.

Multi-View CCA

A direct extension of CCA is Multi-view CCA (MCCA). MCCA is able to analyze the linear relationships between V different views. The objective of MCCA is to find a set of canonical transformations $\{\mathbf{w}_v\}_{v=1}^{V}$ that maximise the sum of the correlation of all pairs of the canonical variates $\{{M_v}^T\mathbf{w}_v\}_{v=1}^{V}$. The MCCA objective functions is expressed as follows:

$$\begin{aligned} &\underset{\{\mathbf{w}_v\}_{v=1}^{V}}{\operatorname{argmax}} \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk}\mathbf{w}_k \\ &\text{s.t} \quad \mathbf{w}_v^T C_{vv}\mathbf{w}_v = 1, \ \text{ for } \ v = \{1, 2, ..., V\}. \end{aligned}$$

Note that this objective function uses the same concept of CCA showed earlier; however, it takes into account the correlation of all pairs of canonical variates, therefore

extending CCA to more than two views. Using the Lagrange multiplier method on the MCCA objective, we get a multivariate generalized eigenvalue problem (MEP) [159], show as follows:

$$\hat{C}\hat{\mathbf{w}} = \Lambda H \hat{\mathbf{w}}. \tag{2.29}$$

$\hat{\mathbf{w}}$ is a p-dimensional vector ($p = \sum_v p_v$), $\hat{C}$ is a block matrix with each $(i,j)^{\text{th}}$ bock being matrix cross-covariance matrix C_{ij}, H and Λ are both block diagonal matrices; shown as:

$$\hat{C} = \begin{bmatrix} C_{11} & \dots & C_{1V} \\ \vdots & \ddots & \vdots \\ C_{V1} & & C_{VV} \end{bmatrix}, \ \hat{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_V \end{bmatrix}, \ \Lambda = \begin{bmatrix} \lambda_1 I_{p_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \lambda_V I_{p_v} \end{bmatrix}. \ H = \begin{bmatrix} C_{11} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & C_{VV} \end{bmatrix},$$

Where I_{p_i} is the identify matrix with dimensions $p_i \times p_i$, and $\mathbf{0}$ is the zero matrix. The steps going from the MCCA objective to the MEP is show in the Appendix (A.6).
The main drawback of MCCA is that it only takes into account linear relationships of the data; therefore fails to model non-linearities in the data. Moreover, it is non-trivial to find an exact analytical solution to an MCP problem [160]; only approximate solutions can be found.

Graph Regularized MCCA

The drawbacks of MCCA previously mentioned are addressed by Graph regularized MCCA (GrMCCA) [52]. To model non-linearities in the data, GrMCCA makes use of a graph regularizer for every view which induces smoothness over the manifold of each view. The canonical variates are smoothed over the corresponding Laplacian matrix of every view Φ_v. Thus, the graph regularizer for each view is: $\mathbf{w}_v^T M_v \Phi_v M_v^T \mathbf{w}_v$. The GrMCCA objective function combines the sum of the graph regularizers of every view to the MCCA objective function:

$$\begin{aligned} &\underset{\{\mathbf{w}_v\}_{v=1}^V}{\text{argmax}} \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk} \mathbf{w}_k - \eta \sum_{v=1}^{V} \mathbf{w}_v^T M_v \Phi_v M_v^T \mathbf{w}_v \\ &\text{s.t} \quad \mathbf{w}_v^T C_{vv} \mathbf{w}_v = 1, \ \text{ for } \ v = \{1, 2, ..., V\}. \end{aligned} \tag{2.30}$$

The first term in 2.30 induces maximal correlation among all the canonical variates, and the second term makes sure that all the local geometric structure of the original data is enforced into the canonical variates.
Similar to MCCA, a MEP is obtained from problem 2.30 by applying the Lagrange multiplier method. The problem with the MEP is that a global solutions is hard to find, only an approximation of the global solution is feasible [160]. Therefore, in [52]

the authors solve this issue by rewriting problem 2.30 in a different form that groups all the V different constraints. It is rewritten as follows:

$$\begin{aligned} &\underset{\{\mathbf{w}_v\}_{v=1}^V}{\text{argmax}} \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk}\mathbf{w}_k - \eta \sum_{v=1}^{V} \mathbf{w}_v^T M_v \Phi_v M_v^T \mathbf{w}_v \\ &\text{s.t} \quad \sum_{v=1}^{V} \mathbf{w}_v^T C_{vv}\mathbf{w}_v = V. \end{aligned} \tag{2.31}$$

Problems 2.30 and 2.31 are equivalent to each other. However, problem 2.31 can now be solved using a simple generalized eigenvalue problem which gives a global solution. To solve problem 2.31 the Lagrange multiplier method is used. The Lagrange function is:

$$\mathcal{L}(\{\mathbf{w}_v\}_{v=1}^V, \lambda) = \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk}\mathbf{w}_k - \eta \sum_{v=1}^{V} \mathbf{w}_v^T M_v \Phi_v M_v^T \mathbf{w}_v - \lambda\Big(\sum_{v=1}^{V} \mathbf{w}_v^T C_{vv}\mathbf{w}_v - V\Big).$$

Where λ is the Lagrange multiplier. The gradient of the Lagrange multiplier function $\mathcal{L}(\{\mathbf{w}_v\}_{v=1}^V, \lambda)$ with respect to a specific $\mathbf{w}_v$ is:

$$\frac{\delta \mathcal{L}}{\delta \mathbf{w}_v} = 0; \quad 2\sum_{k=1}^{V} C_{vk}\mathbf{w}_k - 2\eta M_v \Phi_v M_v^T \mathbf{w}_v - 2\lambda C_{vv}\mathbf{w}_v = 0,$$

this gives V different equations:

$$\sum_{k=1}^{V} C_{vk}\mathbf{w}_k - \eta M_v \Phi_v M_v^T \mathbf{w}_v = \lambda C_{vv}\mathbf{w}_v. \quad \forall v \tag{2.32}$$

Now the V equations can be rewritten in terms of block matrices as:

$$\begin{bmatrix} \hat{C}_{11} & \dots & C_{1V} \\ \vdots & \ddots & \vdots \\ C_{V1} & & \hat{C}_{VV} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_V \end{bmatrix} = \lambda \begin{bmatrix} C_{11} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & C_{VV} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_V \end{bmatrix}. \tag{2.33}$$

Where $\hat{C}_{vv}$ is:

$$C_{vv} - \eta M_v \Phi_v M_v^T.$$

Now problem 2.33 is a generalized eigenvalue problem. The concatenated canonical transformation vectors are the eigenvectors of the generalized eigenvalue problem 2.33 corresponding the eigenvalue λ. Therefore, we choose a set of r eigenvectors $\{\hat{\mathbf{w}}_i\}_{i=1}^r$ corresponding to the r largest eigenvalues of problem 2.33, where $\hat{\mathbf{w}}^T = (\mathbf{w}_1^T, ..., \mathbf{w}_V^T)$ is the concatenated vector of canonical transformations of each view. Now the projection

matrices for each view are constructed as: $P^v = (\mathbf{w}_1^v, ..., \mathbf{w}_r^v)$ for $v = \{1, 2, ..., V\}$. Once the projection matrices are constructed each view can be projected onto its r-dimensional subspace as $Z_v = {P_v}^T M_v$. Now that the r-dimensional projections are found the shared latent subspace is constructed by summing all the projections as: $Z^* = \sum_{v=1}^{V} Z_v$.

Non-Negative Matrix Factorization Based

The aim of Non-negative Matrix Factorization (NMF) is to decompose the input data matrix into a low rank matrix constructed by the product of two non-negative matrices. NMF is based on the following objective function:

$$\begin{aligned} &\underset{P,Q}{\operatorname{argmin}} \, ||M - PZ||_F^2, \\ &\text{s.t} \quad P \geq 0, Z \geq 0, \end{aligned} \tag{2.34}$$

where $P \in \mathbb{R}^{p \times r}$ is a non-negative basis matrix and $Z \in \mathbb{R}^{r \times n}$ is the non-negative low-dimensional representation of the input data matrix. Note that the product PZ gives and r rank matrix. Moreover, the NMF objective function is solved by alternating updates between P and Z, with multiplicative update rules that ensure that both P and Z remain non-negative [161]. After the low-dimensional representation Z is found any traditional clustering method can be used to partition the samples.
An extension to the standard single-view NMF model is Multi-NMF [51]. It applies NMF to each view (M_v) and then integrates all the low-dimensional representations (Z_v) by inducing similarity to a *consensus* matrix (Z^*). Multi-NMF minimizes the following objective function:

$$\begin{aligned} &\underset{\{P_v, Z_v\}_{v=1}^V}{\operatorname{argmin}} \sum_{v=1}^{V} ||M_v - P_v Z_v||_F^2 + \lambda_v ||Z_v - Z^*||_F^2 \\ &\text{s.t} \quad P_v, Z_v, Z^* \geq 0, \end{aligned} \tag{2.35}$$

where $Z^* \in \mathbb{R}^{r \times n}$ is the shared low-dimensional representation or *consensus* matrix of the multi-view data. Single-view clustering techniques are then applied on the *consensus* matrix to obtain sample partitioning. λ_v are mixture coefficients to weight the effect of each view on the shared latent representation.
The problem with this type of factorized model is that the objective function is non-convex, therefore only sub-optimal solutions can be obtained. Moreover, the non-linear structure of the data is ignored as problem 2.35 only models the linear structure of the data.

Convex Multi-View Subspace Learning

All previously mentioned multi-view clustering methods, except for LRACluster are based on non-convex objectives. This means that optimization procedure for those methods do not reach a global optimum, and the solution obtained is highly variable and depends on the initialization of the algorithm. Convex Multi-view Subspace Learning (CMSL) [162] is a convex formulation of dimensionality reduction. CMSL objective stems from CCA; it devises a convex variation of it. CMSL solves for a shared latent low-dimensional representation Z^* using a convex objective function. After Z^* is found, single-view clustering can be applied to it. This method just like CCA can only take into account two views of the data.
Another convex multi-view subspace learning method is Convex Subspace Representation Learning (CSRL) [163]. CSRL can take into account multiple data views. CSRL minimizes a convex formulation of the general multi-view subspace learning framework with an additional regularizer:

$$\underset{Z^*,\{B_v\}_{v=1}^V}{\operatorname{argmin}} \sum_{v=1}^{V} \lambda_v ||Z_v - B_v Z^*||_F^2 + ||Z^*||_{1,2}. \tag{2.36}$$

Where $||Z^*||_{1,2}$ is the $l_{1,2}$ norm of Z^*, it is the sum of the l_2 norms of the rows of Z^*; defined by: $||Z^*||_{1,2} = \sum_{i=1}^{r} ||\boldsymbol{Z}^*_{i,:}||_2$.
Both CMSL and CSRL are convex methods that can be solved using optimizers converging to global solutions. However, both these methods fail to model the intrinsic non-linearities present in the data.

2.4 Literature Review Summary

In this chapter we have reviewed the different methods of outlier detection, graph regularized linear low rank methods, and multi-view clustering methods. Before moving onto the proposed subspace learning methods and their applications on genomic datasets and multi-*omic* datasets, we introduce as background the gradient based methods used to solve the robust subspace methods; RPCA and OP.

Chapter 3

Gradient Based Methods

This chapter introduces the gradient based methods used in this thesis, in addition to relevant background that motivates and leads to each method. Proximal gradient methods and dual methods arise during the thesis as optimization algorithms. Proximal gradient methods are used by Xu et al. [3] to optimize OP, and Candès et al. [2] to optimize RPCA. Dual methods are used by Shahid et al. [58] to optimize RPCAG, and will be used to optimize our proposed methods: GOP and CGRMSL.
Proximal gradient methods are part of the family of first-order optimization methods, which date back to the 1950's. However only recently there has been renewed interest in them in the field of large scale optimization. This is due to the simple computation of only first-derivatives, in contrast to second order optimization methods where the computational expense becomes too high in large scale problems due to the computation of the Hessian matrix,$(\nabla^2 f(\mathbf{x}))$.
We will first present the simpler gradient descent method to set the path for the more elaborate proximal gradient methods. Then we will put into picture the general form of the proximal gradient method and its accelerated version known as accelerated proximal gradient method. After introducing the general form of both methods we will introduce the accelerated proximal gradient algorithm used to solve the Outlier Pursuit problem.
Dual methods are also gradient based optimization methods that use gradient ascent to optimize the dual problem. We will provide background about the dual ascent method and the augmented Lagrangian method, which will serve as precursors to the Alternating Direction Method of Multipliers (ADMM). ADMM will be then used in later chapters to optimize our proposed methods of GOP and CGRMSL.

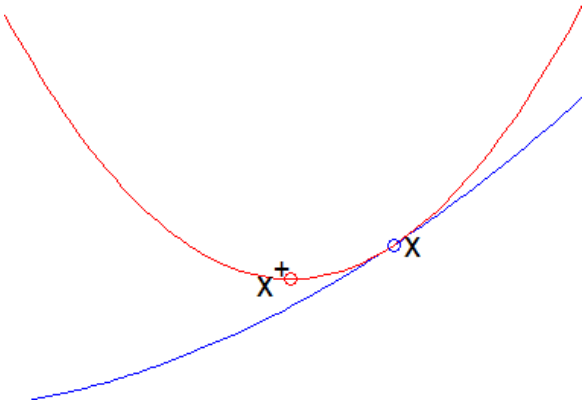


Figure 3.1: red line represents quadratic approximation at point x and blue line represents $f(y)$,

3.1 Gradient Descent

Consider the unconstrained smooth convex optimization problem:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}), \tag{3.1}$$

where f is convex and differential with $\mathbf{x} \in \mathbb{R}^n$. The gradient descent algorithm is as follows:

Algorithm 1 Gradient Descent

1: choose initial value as $\mathbf{x}^0 \in \mathbb{R}^n$
2: $\mathbf{x}^{k+1} = \mathbf{x}^k - t\nabla f(\mathbf{x}^k)$
3: repeat until convergence

where k is the iteration index and t is the step size. Note that any initial point will give a global solution, because f is a convex function, this statement will be untrue if f was non-convex.
Gradient descent can be interpreted as choosing $\mathbf{x}^{k+1}$ as the minimum of a quadratic approximation at $\mathbf{x}^k$. Moreover, consider the Taylor expansion around a point $\mathbf{x}$ by replacing the usual $\nabla^2 f(\mathbf{x})$ by $\frac{1}{t}I$, where I is the identity matrix,

$$f(\mathbf{y}) \approx \ f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{1}{2t}||\mathbf{y} - \mathbf{x}||_2^2,$$

here the term $f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$ can be seen as a linear approximation term of f, and the term $\frac{1}{2t}||\mathbf{y} - \mathbf{x}||_2^2$ is seen as a proximity term to $\mathbf{x}$ with weight $\frac{1}{2t}$. Now by choosing next point $\mathbf{x}^{k+1}$ as $\mathbf{x}^+$ and $\mathbf{x}$ as current point $\mathbf{x}^k$, the next point $\mathbf{x}^+$ can be

chosen to be the minimizer of the quadratic approximation as shown in Figure 3.1 and is defined by the following minimization problem:

$$\mathbf{x}^+ = \underset{\mathbf{y}}{\operatorname{argmin}}\, f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{1}{2t}||\mathbf{y}-\mathbf{x}||_2^2. \tag{3.2}$$

The parameter t in the previous equation sets the step size of gradient descent, choosing the correct step size is of crucial importance in first order optimization algorithms such as gradient descent and proximal gradient methods. One way to tackle this problem, is to have an extra assumption on the function f, this assumption is that ∇f is Lipschitz continuous with Lipschitz constant $L > 0$. The Lipschitz constant needs to satisfy the following inequality

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq L||\mathbf{x}-\mathbf{y}||_2 \quad \forall \mathbf{x}, \mathbf{y}.$$

With this additional assumption on f, gradient descent with fixed step size $t \leq 1/L$ satisfies,

$$f(\mathbf{x}^k) - f^* \leq \frac{||\mathbf{x}^0 - \mathbf{x}^*||_2^2}{2tk}, \tag{3.3}$$

where f^* is an optimal solution to 3.1 and $f(\mathbf{x}^k)$ is the value of f at iteration k. From inequality 3.3 we can say that gradient descent has a convergence rate of $O(1/k)$. Inequality 3.3 can be easily proved through the convergence analysis by taking into account the Lipschitz assumption and the convexity of f, outline of proof can be found in [164].

3.2 Proximal Gradient Descent

Proximal gradient methods solve the following unconstrained problem with the objective function split into two components:

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}), \tag{3.4}$$

where g is a scalar function $g : \mathbb{R}^n \to \mathbb{R}$, that is convex and differentiable with $\mathbf{x} \in \mathbb{R}^n$. h is also a scalar function that is convex and possibly non-differentiable. In the case were g and h are differentiable, then f would also be differentiable, the gradient descent update would be used to solve 3.4 (recalling the motivation of the gradient descent update that is to minimize the quadratic approximation of f around $\mathbf{x}$ as shown in equation 3.2). In our case f is not differentiable, however g is differentiable, the idea of proximal gradient descent is to have a quadratic approximation on g

around $\mathbf{x}$ and keeping h unchanged. The update would be,

$$\mathbf{x}^+ = \underset{\mathbf{y}}{\operatorname{argmin}} \quad g(\mathbf{x}) + \nabla g(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{1}{2t}||\mathbf{y} - \mathbf{x}||_2^2 + h(\mathbf{y}) \tag{3.5}$$

by ignoring the constant terms the minimization problem 3.5 can be rewritten as,

$$\mathbf{x}^+ = \underset{\mathbf{y}}{\operatorname{argmin}} \quad \frac{1}{2t}||\mathbf{y} - (\mathbf{x} - t\nabla g(\mathbf{x}))||_2^2 + h(\mathbf{y}). \tag{3.6}$$

The first term in equation 3.6 ($\frac{1}{2t}||\mathbf{y} - (\mathbf{x} - t\nabla g(\mathbf{x}))||_2^2$) can be interpreted as a term that induces the solution to stay close to the gradient update of g and the second term being $h(\mathbf{y})$ induces the minimization of h.
Before defining proximal gradient descent we need to define the proximal mapping or sometimes called the proximal operator. It is defined as,

$$\text{prox}_t^h(\boldsymbol{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2t}||\mathbf{x} - \mathbf{y}||_2^2 + h(\mathbf{y}) \tag{3.7}$$

where prox_t^h is the proximal operator of the function h with parameter t. Then the proximal gradient descent algorithm is as follows:

Algorithm 2 Proximal Gradient Descent

1: choose initial value as $\mathbf{x}^0 \in \mathbb{R}^n$
2: $\mathbf{x}^{k+1} = \text{prox}_{t^k}^h(\mathbf{x}^k - t^k \nabla g(\mathbf{x}^k))$
3: repeat until convergence

It has been proved by the convergence analysis of the proximal gradient descent in [165] that proximal gradient descent with fixed step size $t \leq 1/L$ satisfies the following inequality

$$f(\mathbf{x}^k) - f^* \leq \frac{||\mathbf{x}^0 - \mathbf{x}^*||_2^2}{2tk}. \tag{3.8}$$

Which shows that proximal gradient descent has a convergence rate of $O(1/k)$, the same as gradient descent.
However, the convergence rate can still be improved to an optimal convergence rate by adding a momentum term that is cleverly chosen, this approach is called accelerated proximal gradient which will be introduced in the next section.

3.3 Accelerated Proximal Gradient (APG)

Accelerated Proximal Gradient(APG) descent is an efficient algorithm used to solve the unconstrained problem 3.4; it has an optimal convergence rate of $O(1/k^2)$, and

the only difference from the normal proximal gradient method is that the proximity operator update is not applied to $\mathbf{x}$ but it is applied to a cleverly chosen intermediate variable. The general algorithm for APG is as follows,

Algorithm 3 Accelerated Proximal Gradient **(General form)**

1: choose initial value as $\mathbf{x}^0 = \mathbf{x}^{-1} \in \mathbb{R}^n$, $t^0 = t^{-1} = 1$
2: repeat until convergence the following
3: set $\mathbf{v}^k = \mathbf{x}^k + \frac{t^{k-1}-1}{t^k}(\mathbf{x}^k - \mathbf{x}^{k-1})$
4: set $\mathbf{x}^{k+1} = \text{prox}^h_{t^k}(\mathbf{v}^k - t^k \nabla g(\mathbf{v}^k))$
5: choose t^{k+1} satisfying

$$(t^{k+1})^2 - t^{k+1} \leq (t^k)^2 \tag{3.9}$$

In the smooth setting when h is equal to zero in 3.4, Nesterov [166] introduced an algorithm that has only one gradient evaluation per iteration, using only an interpolation strategy to achieve a convergence rate of $O(1/k^2)$. After some years, Beack and Teboulle [167] extended Nesterov's algorithm in [166] to solve the general non-smooth problem 3.4 when h is non-differentiable. They have shown that APG has the following convergence,

$$f(\mathbf{x}^k) - f^* \leq \frac{2||\mathbf{x}^0 - \mathbf{x}^*||_2^2}{t(k+1)^2}. \tag{3.10}$$

This proves the convergence rate of APG being $O(1/k^2)$. For fastest convergence the sequence t^k needs to increase as fast as possible [132]. The choice $t^k = \frac{k+2}{k}$ satisfies inequality 3.10. An alternative is to change the inequality in 3.9 to an equality and solve for t^{k+1} yielding,

$$t^{k+1} = \frac{1 + \sqrt{1 + 4(t^k)^2}}{2}, \tag{3.11}$$

and is used in [167].
Here we have introduced the general formulation of APG. In the next section we will introduce the APG algorithm for Outlier Pursuit. We can express its optimization problem as in 3.4 with a convex, smooth part f and convex, non smooth g. Moreover, Outlier Pursuit has a nice closed form solution for its proximity operator that is computationally non-expensive.

3.4 APG for Outlier Pursuit

In the Outlier Pursuit problem the function that is needed to be minimized is the objective function of 2.15

$$\underbrace{\mu||L||_* + \mu\lambda||C||_{1,2}}_{h(L,C)} + \underbrace{\frac{1}{2}||(L+C)-M||_F^2}_{g(L,C)}, \tag{3.12}$$

notice that h in this case is a function on an ordered pair $h : \mathbb{R}^{n\times m} \times \mathbb{R}^{n\times m} \to \mathbb{R}$ that is separable, with $h(L,C) = h_1(L) + h_2(C)$. It is known form the separable sum property of proximal operators that the following is true

$$\text{prox}^h(L,C) = \big(\text{prox}^{h1}(L), \text{prox}^{h2}(C)\big).$$

With this property we can solve with respect to L and C independently as

$$f(L) = \underbrace{\mu||L||_*}_{h_1(L)} + \underbrace{\frac{1}{2}||(L+C)-M||_F^2}_{g(L)}, \qquad \text{assuming C is constant} \tag{3.13}$$

$$f(C) = \underbrace{\mu\lambda||C||_{1,2}}_{h_2(C)} + \underbrace{\frac{1}{2}||(L+C)-M||_F^2}_{g(C)}. \qquad \text{assuming L is constant} \tag{3.14}$$

Note that the derivative of $g(L,C)$ with respect to L keeping C constant ($\nabla_L g(L,C)$), and the derivative of $g(L,C)$ with respect to C keeping L constant ($\nabla_C g(L,C)$), are both equal to each other and are equal to $(L+C-M)$. Now we can start by defining the proximal gradient update for problem 3.13. We know that $\nabla_L g(L,C) = L+C-M$ the proximal update step becomes

$$L^+ = \text{prox}_t^{h_1}\big(L - t(L+C-M)\big), \tag{3.15}$$

the proximity function of h_1 replaced by $h_1 = \mu||L||_*$ is

$$\text{prox}_t^{h1}(L) = \underset{Y}{\text{argmin}} \quad \frac{1}{2t}||L-Y||_F^2 + \mu||Y||_*. \tag{3.16}$$

The proximity function 3.16 has closed form solution being $\text{prox}_t^{h1}(L) = \mathcal{D}_{\mu t}(L)$ (this is proved in the Appendix). Where $\mathcal{D}_{\mu t}(L)$ is the singular value soft thresholding operator applied to L. Where $\mathcal{D}_{\mu t}(L)$ is

$$\mathcal{D}_\mu(L) = U\xi_\mu(\Sigma)V^T, \tag{3.17}$$

where $L = U\Sigma V^T$ is the SVD of L, and $\xi_\mu(\Sigma)$ is the soft-thresholding operator applied to the diagonal elements of Σ with parameter μ

$$\Sigma_{i,i} := \max\big(\Sigma_{i,i} - \mu, 0\big). \tag{3.18}$$

The proximal gradient update of 3.13 can now be expressed as

$$L^+ = \mathcal{D}_{\mu t}\big(L - t(L + C - M)\big). \tag{3.19}$$

Next, to compute the proximal gradient update to minimize 3.14 we need to find the proximity operator of $h_2 = \mu\lambda||C||_{1,2}$, which is given by

$$\text{prox}_t^{h2}(C) = \underset{Y}{\text{argmin}} \quad \frac{1}{2t}||C - Y||_F^2 + \mu\lambda||Y||_{1,2}. \tag{3.20}$$

This also has a closed form solution, $\text{prox}_t^{h2}(C) = \mathcal{C}_{\mu\lambda t}(C)$, where $\mathcal{C}_{\mu\lambda t}(C)$ is the column-wise soft-thresholding operator (it is applied to each column of C) with parameter $\mu\lambda t$. Its general form with parameter t, $\mathcal{C}_t(C)$, is defined as:

$$\boldsymbol{C}_{:,i} := \begin{bmatrix} \mathbf{0} & if \quad ||\boldsymbol{C}_{:,i}|| \leq t \\ \boldsymbol{C}_{:,i} - t\frac{\boldsymbol{C}_{:,i}}{||\boldsymbol{C}_{:,i}||_2} & if \quad ||\boldsymbol{C}_{:,i}||_2 > t \end{bmatrix}, \tag{3.21}$$

where $\boldsymbol{C}_{:,i}$ is the i^{th} column of C, (Refer to Appendix for derivation). Now the Proximal gradient update to minimize 3.14 becomes

$$C^+ = \mathcal{C}_{\mu\lambda t}\big(C - t(L + C - M)\big). \tag{3.22}$$

The APG algorithm for the Outlier Pursuit problem becomes as follows

Algorithm 4 Accelerated Proximal Gradient **(Outlier Pursuit)**

input: $M \in \mathbb{R}^{m\times n}$,λ, δ,μ_0,η

1: choose initial value of $C^0, C^{-1}, L_0, L_{-1} \in \mathbb{R}^{m\times n}$; $t^0, t^{-1} \leftarrow 1$; $\bar{\mu} \leftarrow \delta\mu_0$

2: repeat the following until convergence

3: set $V_L^k = L^k + \frac{t^{k-1}-1}{t^k}(L^k - L^{k-1})$; $V_C^k = C^k + \frac{t^{k-1}-1}{t^k}(C^k - C^{k-1})$

4: set

5: $L^{k+1} = \mathcal{D}_{\mu^k t^k}\big(V_L^k + t^k(V_L^k + V_C^k - M)\big)$;

6: $C^{k+1} = \mathcal{C}_{\mu^k \lambda t^k}\big(V_C^k + t^k(V_L^k + V_C^k - M)\big)$

7: $\mu^{k+1} = \max\big(\eta\mu_k, \bar{\mu}\big)$

8: choose t^{k+1} satisfying

$$(t^{k+1})^2 - t^{k+1} \leq (t^k)^2 \tag{3.23}$$

output: $\hat{L} = L^k$, $\hat{C} = C^k$ when k is last iteration.

Algorithm 4 is different from the general APG formulation in that it has an added step 7, which performs a continuation technique that has been previously employed by other studies [3,168], stating that practically this step greatly reduces the number of iterations. More precisely the sequence L_k, C_k, generated by Algorithm 4, gets closer to the optimal solution set of 3.12. Moreover, the smaller the $\bar{\mu}$ the closer is the solution to the optimal solution set of the linearly constrained Outlier Pursuit problem that we wished to solve before relaxing linearity 2.13. The same authors [3,168] have chosen $\mu_0 = 0.99||M||_2$,$\delta \leq 10^{-5}$ from empirical results, stating that it is a good choice for most practical purposes. Also they claim that the convergence is slow for $\eta \in (0, 0.5)$; therefore choosing $0.5 < \eta < 1$ is a feasible choice for η (η needs to be smaller than 1 because the sequence μ_k needs to be a decreasing sequence.)

3.5 Dual Methods

In this section we provide some useful background to the Alternating Direction Method of Multipliers (ADMM), by introducing the Dual Ascent method and the Augmented Lagrangian method. Both these dual optimization algorithms are precursors to ADMM.

3.5.1 Dual Ascent Method

Consider the convex optimization problem

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\, f(\boldsymbol{x}) \\ &\text{subject to: } A\boldsymbol{x} = \boldsymbol{b}, \end{aligned} \tag{3.24}$$

its Lagrangian function is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) + \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b}), \tag{3.25}$$

thus, the dual function is

$$g(\boldsymbol{u}) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = -f^*(-A^T\boldsymbol{u}) - \boldsymbol{b}^T\boldsymbol{u}, \tag{3.26}$$

with $\boldsymbol{u}$ being the dual variable, and f^* is the conjugate function of $f(\boldsymbol{x})$ (for detailed individual steps of equation 3.26 refer to Appendix A.7). Now the dual problem of 3.24 is

$$\underset{\boldsymbol{u}}{\text{maximize}}\, g(\boldsymbol{u}). \tag{3.27}$$

In the dual ascent method, the dual problem is solved using gradient ascent. The method first finds $\boldsymbol{x}^+ \in \text{argmin}_{\boldsymbol{x}}\, \mathcal{L}(\boldsymbol{x}, \boldsymbol{u})$, then computes the subgradient (not the gradient because g is assumed to be convex; not strictly convex) of the dual function $\delta g(\boldsymbol{u}) = A\delta f^*(-A^T\boldsymbol{u}) - b$. After the gradient is computed, it takes a step towards the direction of the gradient. A nice property that stems from conjugate functions is that $\boldsymbol{x} \in \delta f^*(-A^T\boldsymbol{u})$. Thus, the subgradient is computed by $\delta g(\boldsymbol{u}) = A\boldsymbol{x} - \boldsymbol{b}$; it is equal to the residual of the equality constraint (this is proved in Appendix A.8). The dual ascent method for the general problem 3.24 is as follows,

Algorithm 5 Duals Ascent Method

1: start by initializing dual variable $\mathbf{u}^0$ and step-size $t^k > 0$.
2: repeat following until convergence
3: $\boldsymbol{x}^{k+1} \in \text{argmin}_{\boldsymbol{x}}\, f(\boldsymbol{x}) + (\boldsymbol{u}^k)^T A\boldsymbol{x}$
4: $\mathbf{u}^{k+1} = \mathbf{u}^{\mathbf{k}} + t^k(A\boldsymbol{x}^{k+1} - \boldsymbol{b})$

where $t^k > 0$ is the step size. The step 3 is the $\boldsymbol{x}$-minimization step, and step 4 is the dual variable update.
A nice property of dual ascent is that if the function $f(\boldsymbol{x})$ is separable with respect

to partitioning the input vector $\boldsymbol{x}$ into N separate subvectors $\boldsymbol{x_i}$; meaning

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(\boldsymbol{x_i}).$$

Then the $\boldsymbol{x}$-minimization step splits into N separate problems that could be solved in parallel [169]. The disadvantage of dual ascent is that f can not be unbounded and needs to strictly convex to ensure convergence. These are strong conditions that are usually not met in many applications. A method that corrects this disadvantage is the Augmented Lagrangian method.

3.5.2 Augmented Lagrangian Method

The Augmented Lagrangian method was mainly developed to improve on the harsh convergence assumptions that are required by the dual ascent method. The augmented Lagrangian of problem 3.24 is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) + \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b}) + \frac{\rho}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2, \tag{3.28}$$

where ρ is a positive regularization parameter of the penalty term. The augmented dual function is denoted as $g_\rho(\boldsymbol{u})$. The augmented Lagrangian can be seen as the non-augmented Lagrangian of

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\, f(\boldsymbol{x}) + \frac{\rho}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2 \\ &\text{subject to: } A\boldsymbol{x} = \boldsymbol{b}. \end{aligned} \tag{3.29}$$

This problem and problem 3.24 are identical, in the sense that any optimal $\boldsymbol{x}$ would add a penalty term of zero to the objective. By including the penalty term, it shows that $g_\rho(\boldsymbol{u})$ can now be differentiable under mild conditions on the original problem 3.24. The gradient of $g_\rho(\boldsymbol{u})$ can be computed in the same way as the dual ascent method. That is, by first computing the $\boldsymbol{x}$-minimization step followed by the dual variable update. The Augmented Lagrangian method for solving problem 3.24 is shown in Algorithm 6.

Algorithm 6 Augmented Lagrangian Method

1: start by initializing dual variable $\mathbf{u}^0$ and step-size $p > 0$.
2: repeat following until convergence
3: $\boldsymbol{x}^{k+1} \in \text{argmin}_{\boldsymbol{x}}\, \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{u}^k)$
4: $\mathbf{u}^{k+1} = \mathbf{u}^{\mathbf{k}} + \rho(A\boldsymbol{x}^{k+1} - \boldsymbol{b})$

Algorithm 6 is similar to the dual ascent method but with the $\boldsymbol{x}$-minimization step on the augmented Lagrangian, and the regularization parameter ρ is used as the step size. An advantage of the augmented Lagrangian method is that it converges under much more general conditions than the dual ascent method, such as cases when f is unbounded and is not strictly convex. A downside of this method is that when the function f is separable, the augmented Lagrangian $\mathcal{L}_\rho$ is not separable. Hence, the $\boldsymbol{x}$-minimization step cannot be decomposed into separate problems.

3.6 Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) is an algorithm that is designed to combine the decomposability of the dual ascent method with the mild convergence assumptions of the augmented Lagrangian method. It is an optimization method that solves problems in the general form of:

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \quad \text{subject to : } A\mathbf{x} + B\mathbf{z} = \mathbf{c}, \tag{3.30}$$

with variables $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$, where $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, and $\mathbf{c} \in \mathbb{R}^p$. We define the augmented Lagrangian function with parameter $p > 0$ as

$$\mathcal{L}_p(\mathbf{x}, \mathbf{z}, \mathbf{u}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{u}^T(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) + \frac{p}{2}||A\mathbf{x} + B\mathbf{z} - c||_2^2, \tag{3.31}$$

where $\mathbf{u}$ is the vector of Lagrange multipliers, each element in the vector corresponds to the Lagrange multiplier for one of the linear constraints. ADMM iterations for the general problem 3.6.1 is as follows:

Algorithm 7 Alternating Direction Method of Multipliers (ADMM)

1: choose initial value for variables: $\mathbf{x}^0, \mathbf{z}^0, \mathbf{u}^0$, and step-size $p > 0$.
2: repeat following until convergence
3: $\mathbf{x}^{k+1} = \text{argmin}_{\mathbf{x}}\, \mathcal{L}_p(\mathbf{x}, \mathbf{z}^k, \mathbf{u}^k)$
4: $\mathbf{z}^{k+1} = \text{argmin}_{z}\, \mathcal{L}_p(\mathbf{x^{k+1}}, \mathbf{z}, \mathbf{u}^k)$
5: $\mathbf{u}^{k+1} = \mathbf{u^k} + p(A\mathbf{x^{k+1}} + B\mathbf{z^{k+1}} - \mathbf{c})$

The ADMM algorithm is proven to converge to a global solution for convex objective functions [169]. In practice, ADMM converges to modest accuracy within a few tens of iterations; this is sufficient for most problems, including the convex optimisation ones that will be solved in this thesis.

3.6.1 Convergence of ADMM

ADMM in Algorithm 7 is called a two-block ADMM because it alternatively updates two separate blocks of variables. Convergence in the two-block case is guaranteed when both functions f and g in problem are convex [169], for any step size $p > 0$. In the three-block and multi-block case of ADMM, convergence is not always guaranteed and in some cases diverges, as shown by Chen et al. in [170]. Moreover, Chen et al. in [170] also proved that the presence of a mild condition guarantees the convergence of the extension of ADMM to a multi-block form. The general form of a multi-block ADMM is as follows:

$$\begin{aligned} &\min_{\boldsymbol{x_i}\forall i} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) \\ &\text{subject to: } \sum_{i=1}^{N} A_i \boldsymbol{x}_i = \boldsymbol{b}. \end{aligned} \tag{3.32}$$

The condition proved by [170] is the following.

Condition 1 [170]: Convergence of multi-block ADMM (problem 3.32) is guaranteed when any two coefficient matrices, A_i, are orthogonal to each other. Therefore, if $N = 3$, then if any of the following conditions is true: $A_1^T A_2 = I$, $A_2^T A_3 = I$, or $A_3^T A_1 = I$, then convergence is guaranteed.

3.7 Summary

In this chapter we have introduced the Proximal Gradient method (PG) and its accelerated version APG in their general form; then we have presented the APG algorithm for Outlier Pursuit. In the next chapter we will introduce our method: Graph regularized Outlier Pursuit (GOP), and how to optimize it using the Alternating Direction Method of Multipliers (ADMM) [169].

Chapter 4

Robust Subspace Methods for Outlier Detection in Genomic Data Circumvents the Curse of Dimensionality

In the previous chapters we discussed Outlier Pursuit and how it can be solved algorithmically. However, we need to emphasize that this model does not take into account the non-linear structure of the data. This is usually a strong drawback when it comes to real datasets where the samples will most likely lie approximately on a non-linear subspace, or in other words, a manifold. This can lead to misleading or unsatisfactory results when it comes to highly non-linear datasets such as Transcriptomic and Protemoic data where features and samples can have complex relationships between each other.

As we focus on genomic data, which has a highly complex structure, we will incorporate a term in the objective function of OP, that keeps the convexity of the problem and also incorporates the complex manifold structure of the input data. Inspired by work in computer vision by Shahid et al. [58], which added a graph regularization term to the objective function of RPCA [2] to incorporate the geometric structure of the data in the recovered low-dimensional space. We introduce in this chapter Graph regualrized Outlier Pursuit (GOP), where we add a graph regularization term to the objective function of OP, with the aim to find a low-dimensional representation which respects the intrinsic geometric structure of the data. The main difference between the proposed model, GOP, and the sparse model of: RPCAG [58] and RPCA [2], is that they assumed the error matrix to be sparse, with arbitrary support, meaning that any entry in the matrix can be non-zero. This corruption model is more intuitive in an image setting where corruptions are most likely spread all around the image. However, in the case of genomic data an intuitive corruption model would be for the

error matrix C to be column sparse; meaning that few samples are heavily corrupted in most features.

4.1 Introduction

The problem of cancer classification, and identifying clinically relevant tumor subgroups are aided by monitoring gene expression [6, 7]. Moreover, gene expression profiling is one of the key approaches used to find potential biomarkers and therapeutic targets for distinct cancer types [8]. However, these large datasets are often affected by outliers. In common language, outliers are a small fraction of samples that deviate considerably from other samples in the population. Outliers can arise from errors in the experimental procedure or can be samples that are functionally different from the majority of the population. In the former case, they are discarded to prevent them from affecting downstream statistical analysis [35], and in the latter case, outliers can be further analyzed to find that they belong to a rare cell type or to a functionally distinct group of cells [171]. Therefore, machine learning techniques that are robust to outliers are of great interest, as they will be able to compute models that are not affected by abnormalities, and will be able to detect outliers. As an example, robust regression has been applied recently with much success in [11,12]. This work shows that in some cases concentration of proteins in yeast cells could be predicted from mRNA abundance in addition with sequence derived features. Extracted outliers from their model were recognized as being subject to post-translational modifications. Another approach for outlier and anomaly detection is to fit a probability density function on the data. Methods such as Gaussian mixture models and kernel density estimation have been used by [82,172] to detect novelties in different applications. However, more recently [173] has discussed that using a mixture of Gaussian components can overfit a cluster of outliers. This configuration happens frequently in real settings where the outliers have a high similarity between each other. Using a single Gaussian component works surprisingly well in practice for outlier detection tasks [173]. However, probability density fitting methods will break down, if applied directly to gene expression datasets, because they suffer from high dimensionality, as their number of features (genes) is much greater when compared to their number of samples. The problem with high-dimensional datasets is that when the number of features increases the volume of the space increases in such a rapid manner that the available samples are not sufficient to get statistically significant results. By reducing the dimensionality of the data, and keeping the same number of samples it will be possible to apply statistical techniques to extract useful information. This will solve the issues caused by the curse of dimensionality. Therefore, it is of great interest to reduce the dimensionality of gene expression datasets without losing too much useful biological information. A widely used dimensionality reduction technique is Principal

Component Analysis (PCA), which showed its importance during the past years in data analysis, especially on high-dimensional transcriptomic datasets [13]. PCA seeks to find a low-dimensional subspace that has the smallest least squares reconstruction error [15]. However, it is known to be heavily affected by outliers in the data. Even in the presence of one outlier [2, 32]. This is mainly because the least-squares error that is minimized in the PCA objective function has a quadratic term which will amplify the errors produced by the outliers in the data. This motivated many researchers over the past years to find formulations of PCA that are robust to outliers. However, many robust PCA algorithms suffer from two main drawbacks: computational intractability and degradation of performance when the dimensionality of the data increases [3]. A robust PCA method that considers these two drawbacks is Outlier Pursuit (OP) which is introduced by Xu et al. in [3]. OP considers the problem of recovering the column space of the uncorrupted points and the index of the outlier points that are present in the data by minimizing a convex objective function. This convexity makes the problem solvable by simple optimization methods that can find a global minimum of the objective function. On the contrary, the state of the art Robust PCA methods are non-convex and optimization methods will converge to local minima. However, OP does not take into account the inherent manifold structure of the data; this is also a known drawback of standard PCA. This gives misleading or unsatisfactory results when it comes to highly non-linear datasets such as Transcriptomic and Proteomic datasets where features and samples can have complex relationships between each other.

In this chapter, we focus on gene expression data which will naturally have a highly complex structure. To solve this issue, we will introduce Graph regularized Outlier Pursuit (GOP), that has a graph regularization term incorporated in its objective function 4.1, with the aim to find a low-dimensional representation that respects the intrinsic geometric structure that the data lives in. In this chapter we will evaluate how the proposed GOP performs on high-dimensional genomic datasets. It will be put in context with other methods, such as traditional probabilistic outlier detection methods that will be negatively affected in high-dimensional spaces, and with traditional dimensionality reduction techniques for subspace recovery. Results on publicly available genomic data will show that GOP robustly detects outliers whereas a density based method fails even at moderate dimensions. Moreover, we will show that GOP has better clustering and visualization performance on the recovered low-dimensional representation when compared to popular dimensionality reduction techniques.

This chapter is organised as follows: Section 4.2 introduces GOP and how it can be solved algorithmically using the ADMM optimization method. Section 4.3, introduces all the outlier detection methods used as a comparison to GOP. These are the OP algorithm, Gaussian density estimation method, and traditional non-parametric methods based on robust metrics; the last two are used as a comparing benchmark

for outlier detection. Section 4.4, shows the practicality of how to detect outliers using the robust low rank approximation methods of: GOP and OP, and how to tune the parameters of both methods. Section 4.5, introduces the high-dimensional gene expression datasets used in this study. Section 4.6, shows the outlier detection results of all methods. Section 4.7, shows that the outlier detection performance of GOP is more suitable to high-diemsnioanl genomic datasets as comapred to RPCAG of Shahid et al. [58]. Section 4.8, shows the convergence of our method, GOP, and OP on the genomic datasets. 4.9, highlights the importance of our method compared to other types of outlier detection methods. Finally, we end with concluding remarks in 4.10.

4.2 Graph Regularized Outlier Pursuit

Graph regularized Outlier Pursuit (GOP) is an outlier detection and low-dimensional subspace recovery method based on structured low rank approximation. GOP incorporates in its objective function the intrinsic manifold structure of the data in the form of a graph. The Graph regularized Outlier Pursuit problem optimizes the following objective function:

$$\min_{L,C} ||L||_* + \lambda||C||_{1,2} + \alpha \operatorname{tr}(L\Phi L^T) \quad \text{subject to: } M{=}L{+}C. \tag{4.1}$$

The equality constraint in problem 4.1 is a harsh condition to be met exactly, in fact the solution to problem 4.1 with the ADMM algorithm (Algorithm 8) converges when $||M - L - C||_F^2 \leq \delta$ (this is one of the convergence criterion of Algorithm 8), where δ is set to a number in the interval (10^{-8}, 10^{-6}). GOP seeks to find the best linear embedding of the data that is robust to outliers, while enhancing the embedding through the graph regularizer. It achieves this by pushing points closer together in the low-dimensional space if they have high affinity $W_{i,j}$ in the original input space. This preserves the intrinsic non-linear structure present in the data while finding the best robust low rank approximation to the data matrix. To best interpret the function

of the graph regularization term $\text{tr}(L\Phi L^T)$ we can rewrite it in the following way:

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{N}||\boldsymbol{L}_{:,i}-\boldsymbol{L}_{:,j}||_2^2 W_{i,j}\\
&=\frac{1}{2}\sum_{i,j=1}^{N}(\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,i}-2\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,j}+\boldsymbol{L}_{:,j}^T\boldsymbol{L}_{:,j})W_{i,j}\\
&=\frac{1}{2}\sum_{i,j=1}^{N}(2\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,i}-2\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,j})W_{i,j}\\
&=\sum_{i=1}^{N}\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,i}\sum_{j=1}^{N}W_{i,j}\quad-\quad\sum_{i,j=1}^{N}\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,j}W_{i,j}\\
&=\sum_{i=1}^{N}\boldsymbol{L}_{:,i}^T\boldsymbol{L}_{:,i}D_{i,i}\quad-\quad\sum_{i,j=1}^{N}\boldsymbol{L}_{:,j}^T\boldsymbol{L}_{:,j}W_{i,j}=\text{tr}(LDL^T)-\text{tr}(LWL^T)\\
&=\text{tr}(L(D-W)L^T)=\text{tr}(L\Phi L^T).
\end{aligned}\tag{4.2}$$

The graph regularization term expressed as $\frac{1}{2}\sum_{i,j=1}^{n}||\boldsymbol{L}_{:,i}-\boldsymbol{L}_{:,j}||_2^2 W_{i,j}$ can be better interpreted. This function will impose structure in the recovered low rank matrix L, in the sense that if two points have high affinity in the original input space the distance of the corresponding columns in L needs to be small. Moreover, the graph regularization term would enhance separability of outliers in the L matrix, when this separable structure is found by the affinity matrix W. We also need to emphasize that problem 4.1 is a convex problem, and it can be solved using Alternation Direction Method of Multipliers (ADMM) [169].
We also need to emphasize that GOP is used in this work to detect outlier samples. However, this method can further be used to detect outlier genes if only all the matrices in problem 4.1 are transposed. The ADMM algorithm for solving GOP is shown in the next subsection.

4.2.1 ADMM Algorithm for Solving Graph Regularized Outlier Pursuit

To solve GOP using ADMM we need to introduce an auxiliary variable, so that we can divide the objective function into three separate blocks. We rewrite the GOP objective function as follows:

$$\begin{aligned}
&\min_{L,C,Q}||L||_*+\lambda||C||_{1,2}+\alpha\,\text{tr}(Q\Phi Q^T)\\
&\text{subject to: } M=L+C,\quad L=Q,
\end{aligned}\tag{4.3}$$

where Q is an auxiliary variable. Now we can define the augmented Lagrangian function of 4.3:

$$\mathcal{L}_p(L, C, Q, Z_1, Z_2) = ||L||_* + \lambda||C||_{1,2} + \alpha \operatorname{tr}(Q\Phi Q^T)$$
$$+ \langle Z_1, M - L - C\rangle + \frac{p_1}{2}||M - L - C||_F^2$$
$$+ \langle Z_2, Q - L\rangle + \frac{p_2}{2}||Q - L||_F^2.$$

Where $\langle ., .\rangle$ denotes the Frobenius inner product of two matrices, if $\langle X, Y\rangle$ then it is defined as $\operatorname{tr}(X^T Y)$. Here we need to minimize the augmented Lagrangian with respect to each of the five variables sequentially. The general form of the ADMM algorithm to solve GOP is shown in Algorithm 8, where Z_1^k and Z_2^k are the Lagrange multipliers and k is the iteration index.
Step 4, 5 and 6 of Algorithm 8 have closed form solutions and they are derived in the following paragraph.
We divide the solution of the augment Largrangian of problem 4.3 by solving three subproblems sequentially, with each one being the minimization of the augment Lagrangian $\mathcal{L}$ with respect to one of the matrix variables: L, C and Q. The three subproblems are solved in the following order:

1. $L^{k+1} = \underset{L}{\operatorname{argmin}} \mathcal{L}(L, C^k, Q^k, Z_1^k, Z_2^k)$,
2. $C^{k+1} = \underset{C}{\operatorname{argmin}} \mathcal{L}(L^{k+1}, C, Q^k, Z_1^k, Z_2^k)$,
3. $Q^{k+1} = \underset{Q}{\operatorname{argmin}} \mathcal{L}(L^{k+1}, C^{k+1}, Q, Z_1^k, Z_2^k)$.

However, before showing the updates of the primal variables L, C and Q the reader is reminded about the definition of the proximity operator [174]. The proximity operator is defined by:

$$\operatorname{prox}_h(X) = \underset{Y}{\operatorname{argmin}}\, h(Y) + \frac{1}{2}||Y - X||_F^2, \tag{4.4}$$

where $h : \mathbb{R}^{p \times n} \to \mathbb{R}$ is a convex function that takes as input a matrix with dimensions p x n and outputs a real valued number. The closed form solutions of the updates are shown as follows:

1. **Updating** L (finding L^{k+1}):
   $L^{k+1} = \underset{L}{\operatorname{argmin}} \mathcal{L}(L, C^k, Q^k, Z_1^k, Z_2^k)$. Terms that are not related to L are con-

stants and thus are discarded. This gives us:

$$
\begin{aligned}
L^{k+1} &= \underset{L}{\operatorname{argmin}} ||L||_* + \langle Z_1^k, M - L - C^k \rangle + \frac{p_1}{2}||M - L - C^k||_F^2 + \langle Z_2^k, Q^k - L \rangle \\
&+ \frac{p_2}{2}||Q^k - L||_F^2. \\
&= \underset{L}{\operatorname{argmin}} ||L||_* + \frac{p_1}{2}\Big|\Big|L - (M - C^k + \frac{Z_1^k}{p_1})\Big|\Big|_F^2 + \frac{p_2}{2}\Big|\Big|L - (Q^k + \frac{Z_2^k}{p_2})\Big|\Big|_F^2. \\
&= \underset{L}{\operatorname{argmin}} \frac{||L||_*}{p_1 + p_2} + \frac{1}{2}\Big|\Big|L - \frac{p_1 R_1^k + p_2 R_2^k}{p_1 + p_2}\Big|\Big|. \\
&= \operatorname{prox}_{\frac{||L||_*}{p_1+p_2}}(\frac{p_1 R_1^k + p_2 R_2^k}{p_1 + p_2}),
\end{aligned}
$$

where $R_1^k = M - C^k + \frac{Z_1^k}{p_1}$ and $R_2^k = Q^k + \frac{Z_2^k}{p_2}$. The proximity operator of the nuclear norm function is the singular value soft-thresholding operator as derived in Appendix A.2, which is defined as $\mathcal{D}_\epsilon(X) = U\xi_\epsilon(\Sigma)V^T$, where $X = U\Sigma V^T$ is the SVD of X, and $\xi_\epsilon(\Sigma)$ is the soft-thresholding operator on the diagonal elements of Σ (as expressed in equation 3.18), with parameter ϵ. Now let $H = \frac{p_1 R_1^k + p_2 R_2^k}{p_1 + p_2}$ and $p = \frac{p_1 + p_2}{2}$. The update for L^{k+1} becomes

$$L^{k+1} = \mathcal{D}_{\frac{1}{p}}(H) = P\xi_{\frac{1}{p}}(\Omega)W^T,$$

where $H = P\Omega W^T$ is the SVD of H.

2. **Updating** C:
Using the same procedure as done before we have,

$$
\begin{aligned}
C^{k+1} &= \underset{C}{\operatorname{argmin}} \lambda||C||_{1,2} + \langle Z_1^k, M - L - C \rangle + \frac{p_1}{2}||M - L - C||_F^2. \\
&= \underset{C}{\operatorname{argmin}} \frac{\lambda}{p_1}||C||_{1,2} + ||C - (M - L + \frac{Z_1^k}{p_1})||_F^2. \\
&= \operatorname{prox}_{\frac{\lambda}{p_1}||C||_{1,2}}(M - L^{k+1} + \frac{Z_1^k}{p_1}).
\end{aligned}
$$

The proximity operator of the $||C||_{1,2}$ function is the column-wise soft-thresholding operator as derived in Appendix A.1. Which is defined by $\zeta_\epsilon(C)$, such that if $||\boldsymbol{C}_{:,i}||_2 \leq \epsilon$ set $\boldsymbol{C}_{:,i} = \mathbf{0}$, otherwise set $\boldsymbol{C}_{:,i} = \boldsymbol{C}_{:,i} - \epsilon \cdot \boldsymbol{C}_{:,i}/||\boldsymbol{C}_{:,i}||_2$. Now the update for C^{k+1} becomes

$$C^{k+1} = \zeta_{\frac{\lambda}{p_1}}(M - L^{k+1} + \frac{Z_1^k}{p_1}).$$

3. **Updating** Q:

$$Q^{k+1} = \underset{Q}{\operatorname{argmin}}\, \gamma \operatorname{tr}(Q\Phi Q^T) + \langle Z_2, Q - L\rangle + \frac{p_2}{2}||Q - L||_F^2.$$

$$= \underset{Q}{\operatorname{argmin}}\, \gamma \operatorname{tr}(Q\Phi Q^T) + \frac{p_2}{2}||Q - (L^{k+1} - \frac{Z_2^k}{p_2})||_F^2.$$

This is a differentiable and convex function; thus finding the first derivative and equating it to zero finds a closed form solution for Q^{k+1}:

$$Q^{k+1} = p_2(L^{k+1} - \frac{Z_1^k}{p_2})(\alpha\Phi + p_2 I)^{-1}. \tag{4.5}$$

Algorithm 8, although it is a 3-block ADMM it is guaranteed to converge because it satisfies **Condition 1** (In section 3.6.1). When problem 4.3 is compared to the general from of multi-block ADMM 3.32, it is seen that it has coefficient matrices A_1,A_2,$A_3 = I$. This means that any two coefficient matrices are orthogonal to each other, which satisfies **Condition 1** (Section 3.6.1).

Algorithm 8 Alternating Direction Method of Multipliers **(Graph Regularized Outlier Pursuit)**

input: $M \in \mathbb{R}^{m\times n}$,λ,α, Φ, $p_1 = 1, p_2 = 1$

1. Initialise L^0, C^0, Q^0 to random matrices.
2. $Z_1^0 = M - L^0 - C^0$ and $Z_2^0 = Q^0 - L^0$.
3. **repeat following until convergence**
4. $L^{k+1} = \underset{L}{\operatorname{argmin}}\mathcal{L}(L, C^k, Q^k, Z_1^k, Z_2^k)$
5. $C^{k+1} = \underset{C}{\operatorname{argmin}}\mathcal{L}(L^{k+1}, C, Q^k, Z_1^k, Z_2^k)$
6. $Q^{k+1} = \underset{Q}{\operatorname{argmin}}\mathcal{L}(L^{k+1}, C^{k+1}, Q, Z_1^k, Z_2^k)$
7. $Z_1^{k+1} = Z_1^k + p_1(M - L^{k+1} - C^{k+1})$
8. $Z_2^{k+1} = Z_2^k + p_2(Q^{k+1} - L^{k+1})$

output: $\hat{L} = L^k$, $\hat{C} = C^k$ when k is last iteration.

4.3 Comparing Methods

4.3.1 Outlier Pursuit

Recalling that our input data matrix $M \in \mathbb{R}^{p\times n}$ has n samples arranged in columns with each sample having p features, $M = [\boldsymbol{M}_{:,1}, \boldsymbol{M}_{:,2}, ..., \boldsymbol{M}_{:,n}]$. Where $\boldsymbol{M}_{:,i} \in \mathbb{R}^p$ denotes the i^{th} column of matrix M. We consider the outliers to be fully corrupted columns. The OP objective is to decompose the data matrix M as $M = L + C$, where L is a low rank matrix and C is a column sparse matrix which has a small fraction of its columns that are non-zero. This method is modelling outlier samples as the non-zero columns in C, where they are considered the corrupted points of the data matrix. Moreover, OP models the uncorrupted column space that needs to be recovered as the column space of the low rank matrix L. OP introduced by [3] seeks to minimize the following function:

$$\min_{L,C} ||L||_* + \lambda||C||_{1,2} \quad \text{subject to: } M{=}L{+}C, \tag{4.6}$$

where $||L||_*$ is the nuclear norm of L and it is defined as the sum of its singular values. $||C||_{1,2}$ is the sum of the l_2 norm of the columns of C. λ is a regularization parameter which needs to be tuned; this will be addressed in Subsection 4.4.1. Problem 4.6 is convex, thus it is efficiently solved using first order optimization methods. The algorithm to solve this problem is given in Algorithm 9,

Algorithm 9 Accelerated Proximal Gradient **(Outlier Pursuit)**

input: $M \in \mathbb{R}^{p\times n}$, λ, $\delta = 10^{-5}$, $\eta = 0.9$, $\mu_0 = 0.99||M||_F$.

1. choose initial value of $C^0, C^{-1}, L^0, L^{-1} \in \mathbb{R}^{p\times n}$; $t^0, t^{-1} \leftarrow 1$; $\bar{\mu} \leftarrow \delta\mu_0$
2. **repeat the following until convergence**
3. $Y_L^k = L^k + \frac{t^{k-1}-1}{t^k}(L^k - L^{k-1})$; $Y_C^k = C^k + \frac{t^{k-1}-1}{t^k}(C^k - C^{k-1})$;
4. $(U, S, V) = \text{svd}\big(Y_L^k + \frac{1}{2}(Y_L^k + Y_C^k - M)\big)$;
5. $L^{k+1} = U\xi_{\frac{\mu^k}{2}}(S)V^T$;
6. $C^{k+1} = \zeta_{\frac{\mu^k\lambda}{2}}\big(Y_C^k + \frac{1}{2}(Y_L^k + Y_C^k - M)\big)$
7. $\mu^{k+1} = \max\big(\eta\mu^k, \bar{\mu}\big)$
8. $t^{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$; $\quad k++$

output: $\hat{L} = L^k$, $\hat{C} = C^k$ when k is last iteration.

where L^k stands for L at iteration k, and $||M||_F$ is the Frobenius norm of matrix M defined by $||M||_F = \sqrt{\sum_{i=1}^{n} ||\boldsymbol{M}_{:,i}||_2^2}$. In step 5, $\xi_{\frac{\mu^k}{2}}(S)$ is the soft-thresholding operator, which acts on the diagonal elements of S (as expressed in equation 3.18), with parameter $\frac{\mu^k}{2}$. Furthermore, step 6, $\zeta_{\frac{\mu^k\lambda}{2}}(C)$ is the column wise soft-thresholding operator of C (expressed in equation 3.21), with parameter $\frac{\mu^k\lambda}{2}$.

4.3.2 Outlier Pursuit Algorithm

We have already introduced OP (problem 4.6) in Section 3.4 and how it can be solved using the Accelerated Proximal gradient (APG) method (Section 3.4), which benefits from an optimal convergence rate of $O(1/k^2)$ [175] (where k is the number of iterations). APG method is the accelerated version of the more general Proximal Gradient method which has a convergence rate of $O(1/k)$. The improvement in convergence rate is achieved by the momentum steps devised by [166, 167, 175], in steps 3 and 8 of Algorithm 9 (refer to Section 3.3).

4.3.3 Detecting Outliers with Gaussian Density Estimation

As a benchmark method for outlier detection we will fit a single Gaussian density to the data. We choose a single Gaussian density instead of a mixture of Gaussian densities to detect outliers, as the latter is known to overfit a cluster of closely knit outliers, which will make them hard to detect [173]. The single Gaussian density estimation method models the dataset to be normally distributed about its means in the form of a multivariate Gaussian distribution. The probability distribution function is described as follows:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}|\Sigma|} \exp\big[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\big], \tag{4.7}$$

where $|\Sigma|$ denotes the determinant of the covariance matrix, $\boldsymbol{\mu}$ is the m-dimensional sample mean vector, and Σ is the $m \times m$ sample covariance matrix. The term in the exponential is half the squared Mahalanobis distance of the sample $\mathbf{x}$ to the mean $\boldsymbol{\mu}$. The Mahalanobis distance can be used as the outlier score of each observation $\mathbf{x}$ and it is computed as follows:

$$\mathrm{MD}(\mathbf{x}, \boldsymbol{\mu}, \Sigma) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \tag{4.8}$$

Then the Gaussian density estimation method for outlier detection consists of finding the Mahalanobis distance to the sample mean for each observation in the dataset. Then the points that have the highest distance will have the lowest likelihood $f(\mathbf{x})$ and these samples are considered to be outliers. The Mahalanobis distance to the

mean has shown promising results in intrusion and outlier detection by [176] and [75].

4.3.4 Traditional Outlier Detection Methods

Methods such as Median Absolute Deviation (MAD) and Boxplot (BP) have been applied successfully to gene expression datasets to detect outliers [35]. Such methods are non-parametric and can detect outliers in absence of any assumptions about the distribution of the data. BP method needs to find the lower quartile (25th percentile) and the upper quartile (75th percentile) of a specific sample which consists of a collection of gene expressions. Outlier genes of a specific sample are the data points that are above the upper fence or the data points below the lower fence. The upper fence is 1.5 times the inter quartile range (IQR) above the upper quartile and the lower fence is 1.5 the IQR lower than the lower quartile. The IQR is defined by the difference between the upper and lower quartile. The BP method assigns a sample as an outlier when the number of outlier genes for that sample are greater than a pre-defined threshold. The MAD outlier detection method is implemented by first finding the median of all genes of a sample then the median absolute deviation of all the genes from the median is calculated by:

$$\text{MAD}_i = \text{median}\big(\big|\boldsymbol{M}_{:,i} - \text{median}(\boldsymbol{M}_{:,i})\big|\big)$$

this is found for each sample i. The MAD can be thought of an outlier score for each sample. Therefore, samples with MAD higher than a pre-defined threshold are assigned to be outliers.

4.4 Detecting Outliers Using OP and GOP

The objective of both OP and GOP is to decompose the input data matrix into a low rank plus a column sparse matrix. In the real dataset case, we need to consider that noise will be present, and that the recovered low rank matrix $\hat{L}$ and column sparse matrix $\hat{C}$ will be corrupted by noise. This will result in a $\hat{C}$ matrix that is not strictly column sparse, but will have high l_2 norm for columns that are considered outliers [3]. Therefore, for both OP and GOP we use two methods to detect the outliers:
1) $\hat{C}$ **method**: Rank l_2 norms of columns of $\hat{C}$ in descending order and choose outliers to be the points with l_2 norm higher than a threshold.
2) $\hat{L}$ **method**: First, find the Singular Value Decomposition (SVD) of the recovered $\hat{L}$ matrix, $\hat{L} = U\Sigma V^T$. Then, find the low-dimensional embedding Z by projecting $\hat{L}$ onto its column space U, $Z = U^T\hat{L}$. Finally, perform k-means clustering onto Z to fit two clusters. The minority cluster is chosen to be the outliers.
For the first method, we can only choose a small fraction of highest points as being the

outlier. This method suffers from the drawback that choosing a fraction of outliers needs prior knowledge of the domain. For the second method, k-means clustering will decide which points correspond to which cluster by giving cluster labels, showing a cut-off between outlier and main samples without deciding an outlier fraction a priori. The F-score of two cluster k-means on the low-dimensional embedding Z is used to quantify the performance of outlier detection of the $\hat{L}$ method. The F-score is a measure of accuracy and will give a higher score if the outlier and main samples are better separated in the low-dimensional embedding. The F-score is defined by:

$$\text{F-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \ ,$$

where, Precision = True Positives/(True Positives + False Positives) and Recall = True Positives/Positives. Here the Positives class is defined as the known outliers present in the datasets that we use.

4.4.1 Parameter Setting for OP and GOP

Tuning parameters for unsupervised problems such as GOP and OP is more challenging than for supervised problems where the true labels are available. The presence of labels for supervised problems allows to measure the accuracy of detection, which can be used as a metric to choose optimal regularization parameters. In the case of outlier detection algorithms, using the knowledge of true outliers in the tuning process would not be practical, as users will not know the true outliers beforehand. In the case of GOP and OP, the two factors affected by the regularization parameters are the rank of $\hat{L}$ and the number of outliers detected. Therefore, we can only use both of them to tune the regularization parameters λ and α. The outliers during the tuning process will be detected using the $\hat{L}$ method explained in Section 4.4. The tuning process consists of solving the problem for each value of λ in a specific range, and looking for stable regions of the rank of $\hat{L}$. We then refine the λ search space to the stable region and record the number of outliers. A suitable value of λ needs to be chosen in such a way that the number of detected outliers are less than or equal to an expected fraction of outliers. From our studies we expect a fraction of outliers that is less than 25 % of the data. Moreover, we found that practically the number of outliers and the rank of $\hat{L}$ are not affected by the value of α; thus, we choose its value to be one. An illustration of this parameter setting procedure is shown in the next subsection.

4.4.2 Tuning λ for OP and GOP for the Colon Cancer Dataset

We will show how λ is tuned for the colon cancer dataset. The procedure is the same as the other datasets used in this study. To tune λ for OP we perform a parameter search to find the optimal value of λ. For each λ value we solve the OP problem, and we use the $\hat{L}$ method to detect outliers as explained in Section 4.4. A parameter search is performed on λ from 0.1 to 0.8, and the rank of $\hat{L}$ is recorded at each step. Figure 4.1(a) shows that the most stable rank for $\hat{L}$ is one and three. Therefore, we refine the range of λ from 0.2 to 0.5, and we take the number of outliers recorded using the $\hat{L}$ method, as shown in Figure 4.1(b). We need to state that outliers need to be a small fraction of the total dataset it is best chosen to be less than 25 % of the data. Therefore, we choose a suitable λ to be 0.46 as it gives the smallest number of outliers in the refined range. This gives 9 outlier points, after inspecting their labels, 4 are true outliers and 5 are false positives. To tune λ for GOP we follow the same procedure as done for OP. We solve the GOP problem for a range of λ, from 0.1 to 3, and find the number of outliers for each λ. Figure 4.1(c) shows that 4 and 2 outliers are detected over this range of λ. The most stable rank of $\hat{L}$ is 1 and it detects 4 samples, which means that they can be confidently chosen as being outliers. To be able to visualize in a two-dimensional space, we choose the suitable λ value as being equal to 1.168, which gives 4 outliers and a rank of $\hat{L}$ being equal to 2. After finding optimal λ, we notice that both the number of outliers and the rank of $\hat{L}$ are robust to the value of α. As a result, we can choose for simplicity α being equal to 1.

4.5 Datasets and Data Preparation

We demonstrate our results on three gene expression datasets, which are all publicly available. They are introduced as follows:

1. Colon cancer dataset from [4]. It consists of a normalized dataset that contains 62 samples with gene expression levels of the 2000 genes with highest minimal intensity across the samples. The 62 samples are comprised of 40 tumor samples and 22 normal samples. A total of 40 patients are considered in their study and each tumor sample is taken from a different patient. In this study we will only take into consideration the 40 tumor samples. The author of the data has shown that tumor samples of patients number: 2, 30, 33, 36 and 37 are outliers. They proved this by finding a muscle index for each of the 40 tumor samples and the 22 normal samples, taken from normal non-cancerous tissue. By taking into consideration that colon cancer samples mostly contain epithelial cells, which contain no muscle tissue, a high muscle index suggests a tissue being highly heterogeneous, thus being a misleading tumor sample. Samples of patients: 2, 30, 33, 36 and 37 have muscle index that lies in the range of the muscle index

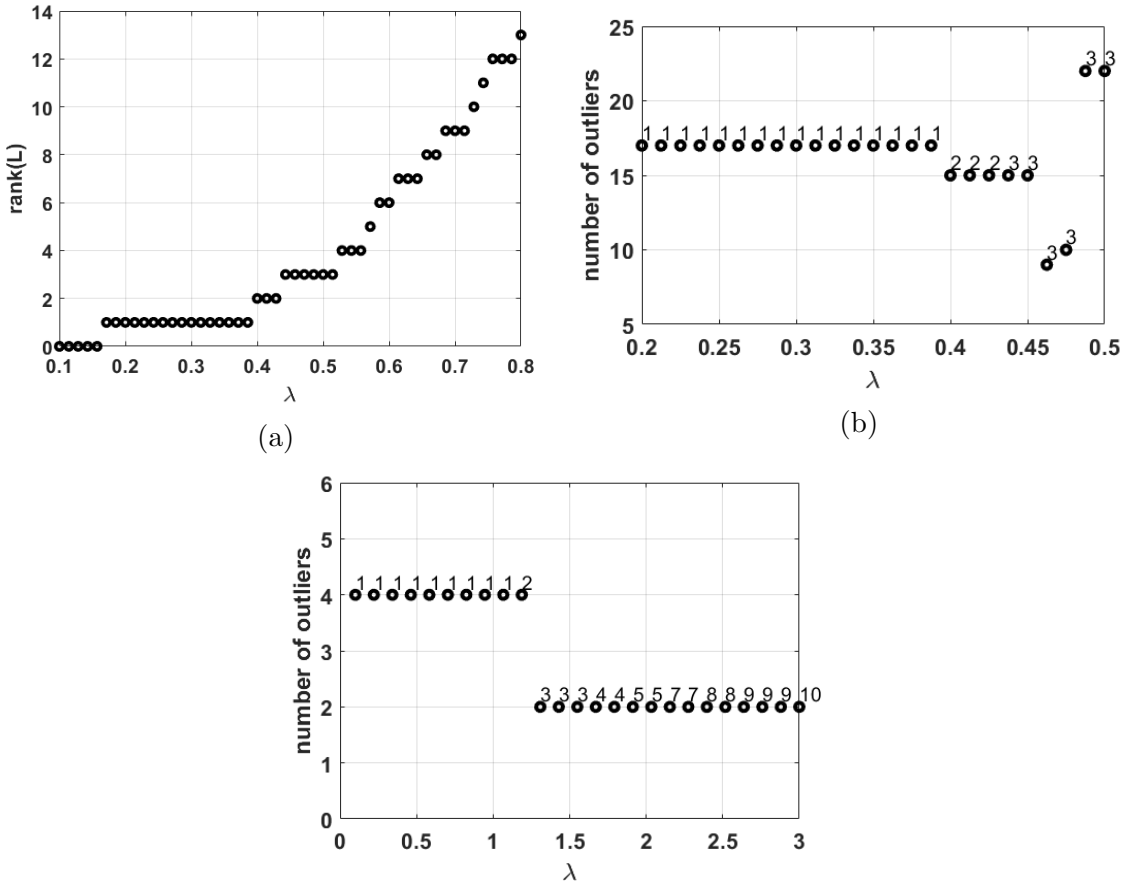


Figure 4.1: (a) Rank of recovered low rank matrix by OP versus regularization parameter λ. Figure shows that the most stable rank for $\hat{L}$ is 1 and 3. Therefore, we can refine the λ search space. (b) Refined λ search space from 0.2 to 0.5. The labels on the circles are the rank of $\hat{L}$ for a specific λ. We choose optimal λ to be 0.46 which gives the smallest number of outliers, in this case 9 outliers. (c) λ vs number of outliers detected. The rank of recovered $\hat{L}$ for each λ is shown as the number above each circle. We choose λ that gives 4 outliers and a rank of 2.

of normal samples, thus being considered as outliers. In our work, we want to retain the genes that contain most of the information. Thus, we pre-process the data by retaining only the 700 most variable genes across samples. The number of most variable genes to keep is chosen to retain more than 85 % of the total variance (sum of the variance of each gene in the dataset). The data is quantile normalized in the same way as [177], to reduce the skew of the microarray data to high expression levels, as recommended by [178].

2. TCGA breast cancer dataset, gathered from UCSC Xena browser [179]. Dataset consists of gene expression at transcription level, expressed as $\log_2(x+1)$ trans-

formed RSEM normalized RNA-sequencing counts. The TCGA dataset contains 20530 genes with 1218 samples, 600 of which are patients with Estrogen Receptor (ER) positive status and 179 with ER negative status, the remaining 439 samples do not have labels for ER, thus are discarded. We sample 100 ER positive samples from the 600 and 5 ER negative samples from the 179. We repeat the random sampling process 30 times, to have 30 datasets that will be used for outlier detection. For each of the datasets the 5 ER negative samples are considered to be the outliers. The choice of 100 ER+, 5 ER- and 30 random sampling repetitions does not affect the conclusions we get from our results. Conclusions will be consistent as long as the datasets are constructed with the reasoning that outlier samples need to be a small fraction of the overall dataset, and that we need to repeat the random sampling process, so that we test the used methods on datasets with the same structure but with potentially different samples.

Since the number of genes is considerably large (>20000), it is necessary to diminish the dimensionality of the data to reduce the time of computation of the robust subspace methods and to get more stable results. Therefore, the data is filtered to retain the most variable genes. This is a commonly used pre-processing procedure for machine learning algorithms applied to genomic datasets [180], to choose the most informative genes. The number of genes to retain is chosen to trade-off between the time of computation and the fraction of the total variance explained by the chosen genes. For each of the 30 datasets, the 2000 most variable genes are retained. Note that choosing a number of genes greater than 2000 does not change the conclusion deduced from the results; it only increases the time needed for computation.

3. Single cell dataset consisting of single cell measurements of mouse embryonic stem cells at 3 different stages of the cell cycle (G1,S,G2M) gathered from [181]. The dataset consists of log transformed normalized count values of gene expressions measured by single-cell RNA-seq for 8989 genes. There is a total of 182 cells, of which 59 in G1, 58 in S, and 65 in G2M. We build a dataset that consists of both the 59 G1 cells and 6 randomly sampled cells from the 65 G2M population. We repeat this random sampling process to gather 30 datasets that have 6 different G2M cells in each instance. For each of the datasets the 6 G2M cells are taken to be the outliers. For each of the 30 constructed datasets the 1000 most variable genes are retained following the same reasoning explained for the breast cancer dataset.

4.6 Results

4.6.1 Outlier Detection on Colon Cancer Dataset

After finding a suitable λ with the procedure shown in Subsection 4.4.2, outliers are detected by inspecting the l_2 norms of the columns of $\hat{C}$. We expect that the columns of $\hat{C}$ corresponding to outliers to have higher l_2 norm than the non-outlier samples. From Figure 4.2 we can see that the 4 highest points are actually 4 out of the 5 known outliers. Using the $\hat{L}$ method for outlier detection, we detect 9 outliers in the minority cluster, 4 of which are the same true outlier samples detected by the $\hat{C}$ method and 5 are false positives.

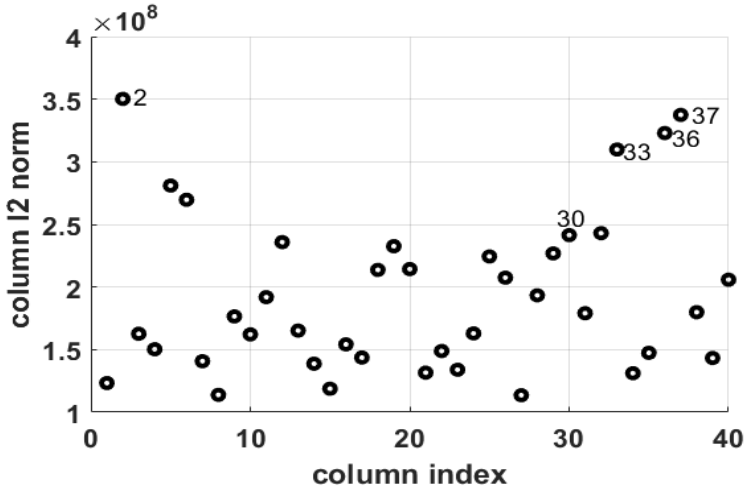


Figure 4.2: Inspecting l_2 norm of columns of $\hat{C}$. The labelled samples are the outlier samples found by the authors of the data in [4]. Figure shows that patients (2,33,36,37) are detected as outlier, except patient 30. This method differs from the $\hat{L}$ method for outlier detection, in that we need to choose the threshold by having prior knowledge of the fraction of outliers.

We apply GOP to the colon cancer dataset to detect outlier samples. Outliers are detected using the $\hat{L}$ method as explained in Section 4.4. The regularization parameters λ and α are tuned as explained in Section 4.4.1. Using α as 1 and optimal λ, 4 outliers are detected and they are part of the 5 known outliers from the author of the data [4]. This gives better outlier detection than OP, which finds the same 4 outliers but has 5 false positives. GOP performs the same as OP when the $\hat{C}$ matrix is used, but we did not have to choose a suitable fraction of outliers. It should be noted that the same 4 out of the 5 known outliers are picked up by average hierarchical clustering used in [35]. To further compare the outlier detection capability of OP and GOP, we project $\hat{L}$ (recovered from each method using optimal regularization parameters) on its first two principal directions, and we find the Mahalanobis distance (MD) between

the two cluster centres found by k-means on the projection. Furthermore, the capability of capturing the outliers visually is compared for GOP, OP, PCA and t-SNE by finding their 2-dimensional embedding. We get an MD of 2.806 for $\hat{L}$ from GOP, and 1.8973 for $\hat{L}$ from OP and an MD of 1.7839 and 1.6777 for PCA and t-SNE respectively. We can see the greater separation between outlier and main samples from GOP in Figure 4.3. In conclusion, $\hat{L}$ recovered by GOP gives better separation between main samples and outlier samples; and this gives fewer false positives when detecting outliers. Although, GOP gives less false positives than OP, it still missed the same outlier that OP missed.

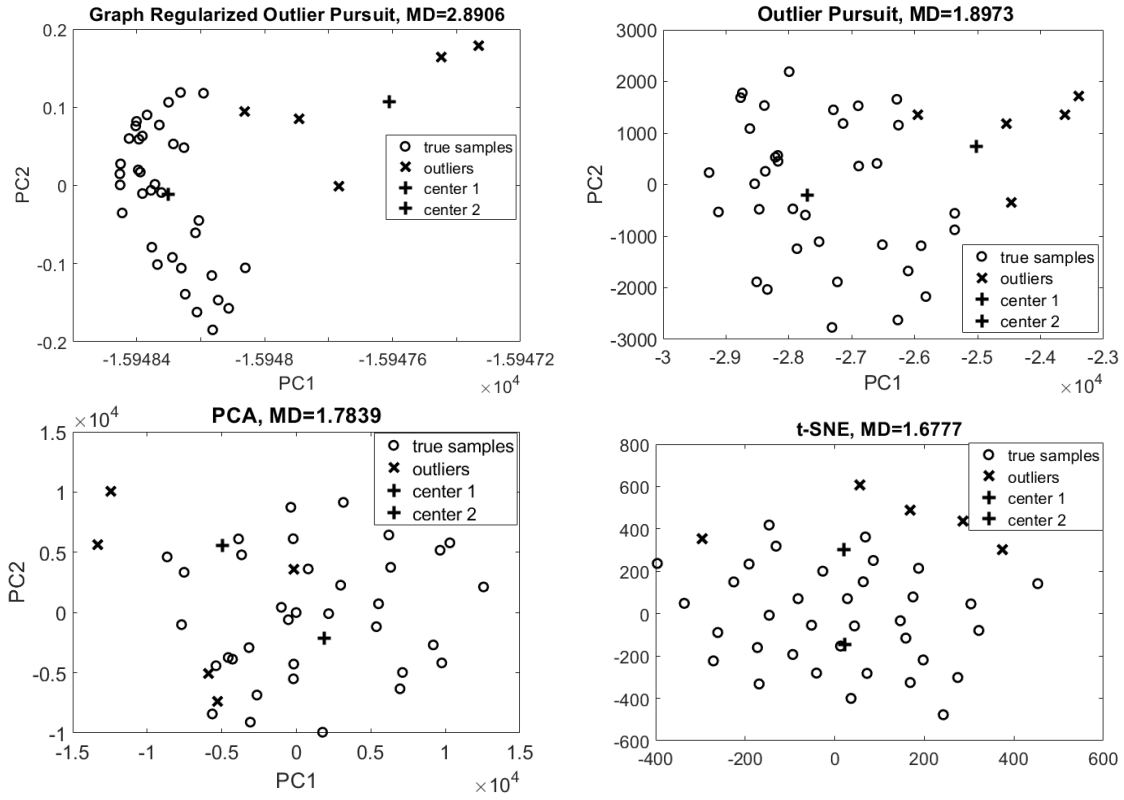


Figure 4.3: 2-dimensional visualization found by GOP, OP, PCA and t-SNE. For GOP and OP we project $\hat{L}$ onto its first two principal directions. PCA and t-SNE are applied directly to the colon cancer dataset. Figure shows that the separation between main and outlier samples is greater in the subspace found by GOP.

4.6.2 Outlier Detection Capability on Breast Cancer Dataset

Given the 30 sampled datasets with 105 samples, we need to detect the 5 ER negative as outliers using the five methods: OP, GOP, Gaussian density, MAD and BP. Each of the 30 datasets will be supplied as input for the aforementioned methods. We detect outliers for OP and GOP using the l_2 norms of the $\hat{C}$ matrix, and will record

the number of false positives encountered before finding all the 5 outlier samples. For the Gaussian density method, we will use the Mahalanobis distance to the sample mean as an outlier score for each sample and record the number of false positives needed to recover all the 5 outliers. For the MAD method the outlier score will be the MAD of each sample, and we record the number of false positives to detect all the 5 known outliers. For the BP method, we will use the number of outlier genes as an outliers score for each sample, and we record the false positives encountered to detect all the 5 known outliers. False positives from the five methods will be recorded for all the 30 randomly sampled datasets. This experiment will be repeated by changing the dimensionality of the 30 datasets. The dimensionality will be changed by using the most variable genes across samples. The false positives for the 30 datasets will be recorded for GOP, OP and the Gaussian density method at 25, 50, 80, 95, 100 and 200 dimensions. At 200 dimensions MAD and BP are added to validate the performance of GOP and OP. The results are shown in Figure 4.4. We can see that for 25 dimensions the number of false positives is high for all three methods, and it decreases when the number of dimensions increases to 50 and 80. This is due to the fact that there is more useful information injected by the added dimensions. At 95, 100 and 200 dimensions the performance of the Gaussian density method is degraded as there are not enough samples compared to the number of dimensions. However, we can see that for both OP and GOP the outlier detection keeps improving by increasing the dimensions of the input dataset. Furthermore, we note that GOP has a smaller median of false positives encountered to detect all 5 outliers on each chosen dimension when compared to OP. Finally, we can see at 200 dimensions that MAD and BP record a much higher median percentage of false positives compared to GOP and OP.

Low-Dimensional Embedding Outlier Detection and Visualization

In the previous subsection we showed that the graph regularizer can enhance the outlier detection performance using the $\hat{C}$ matrix. In this subsection we demonstrate that the same can be achieved for the separation of outlier and main samples in the recovered low rank matrix $\hat{L}$, and how this better separation can be visualized in two dimensions. We recover $\hat{L}$ for each of the 30 randomly sampled datasets. Next, we perform k-means clustering with two clusters on the projection of $\hat{L}$ and find the F-score. This is performed on all the low-dimensional embeddings of GOP, OP, PCA and t-SNE. We supply the same input to all the dimensionality reduction methods which is the 105 samples after filtering its genes to the 2000 most variable genes across samples. From Figure 4.5(a) it is observed that the F-score found on the GOP low dimensional embedding is greater than all other methods. In this case the F-score of GOP is highest because it detects more true positives and less false positives than

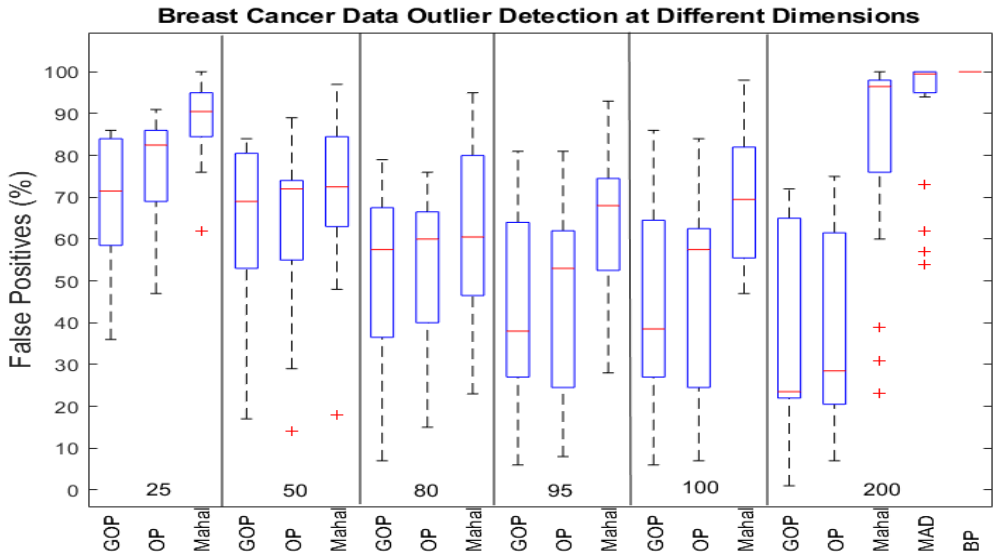


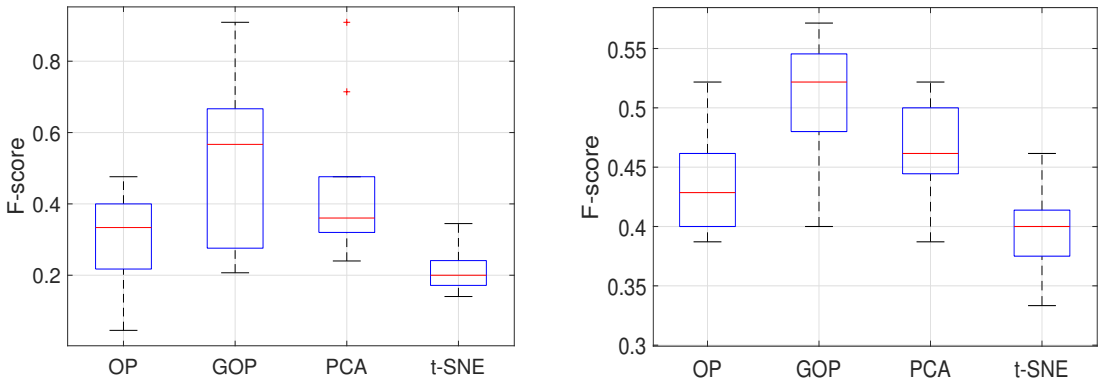
Figure 4.4: Boxplots comparing the number of false positives encountered to detect all 5 outliers in the 30 instances of the breast cancer dataset. Each of the 6 subdivisions of the figure represent running GOP, OP, and the Gaussian density method for all 30 datasets at a specific dimension. The dimension used is indicated at the bottom of each subdivision. The horizontal line in each boxplot corresponds to the median of false positives. We find that the Gaussian density method finds on average more false positives than both OP and GOP. Moreover, we can see that the Gaussian density method suffers from the curse of dimensionality, whereas the subspace methods are robust to high-dimensional datasets. Furthermore, we note that GOP detects less false positives on average than both methods, showing that the outlier detection has benefited from the graph regularization.

all other methods. This greater capability to detect outliers in the low-dimensional embedding can be further seen visually by projection $\hat{L}$ onto its first two principal directions.

The visualization in two dimensions is shown in Figure 4.6. We can observe that GOP gives better separation of the 5 ER- samples and the 100 ER+ samples in its two-dimensional projection. In the t-SNE two-dimensional embedding the separation is also seen clearly. However, we note from the F-score that two cluster k-means on this space fails to find the outliers and main samples accurately. In the case of GOP we can visually observe the separation and quantitatively measure this separation using a standard clustering technique such as k-means.

4.6.3 Outlier Detection Capability on Single Cell Dataset

In this subsection we compare the outlier detection performance of GOP, OP, Gaussian density, MAD and BP methods on the 30 randomly sampled single cell datasets



(a) (b)

Figure 4.5: (a) **(Breast Cancer Dataset)** F-score of k-means clustering for all dimensionality reduction methods, found on the 30 instances of the breast cancer dataset. Each boxplot shows the F-score for all 30 randomly sampled datasets by the corresponding dimensionality reduction method. We can see that GOP has a considerably higher median F-score compared to all other methods. (b) **(Single Cell Dataset)** F-score of k-means for all dimensionality reduction methods applied to the 30 instances of the single cell dataset. We can see that GOP gives the best F-score in its low-dimensional embedding compared to all other methods.

constructed as discussed in Section 4.5. The task consists of finding the 6 G2M cells in the population of 59 G1 cells as outliers in each of the 30 datasets. As in the breast cancer dataset, we record the number of false positives to detect all the known 6 outliers using the aforementioned methods. This is repeated for 6 different dimensions by retaining the most variable genes across samples. The different dimensions are: 2, 20, 30, 50, 60, 70. Figure 4.7 shows the outlier detection results for the single cell dataset. We can see that the outlier detection performance of the Gaussian density method improves when increasing the dimensionality from 2 to 20. Also, the number of false positives start to increase monotonically by increasing the dimensions from 20 to 70. Moreover, we note that the performance of the robust subspace methods improves with the increase in dimensionality, showing that they are effective in filtering out the noise and extracting useful information from high-dimensional datasets. They avoid falling into the curse of dimensionality because the data matrix is modelled to be a low rank matrix that is corrupted by a column sparse matrix modelling the outliers. Thus, the robust subspace methods work in a reduced dimensional space, which help to circumvent the curse of dimensionality. Furthermore, we note that GOP outperforms OP on all dimensions, showing that the graph regularizer is also beneficial on the single cell dataset. Finally, we can see that at 70 dimensions MAD and BP record a considerably higher percentage of false positives when compared to GOP and OP.

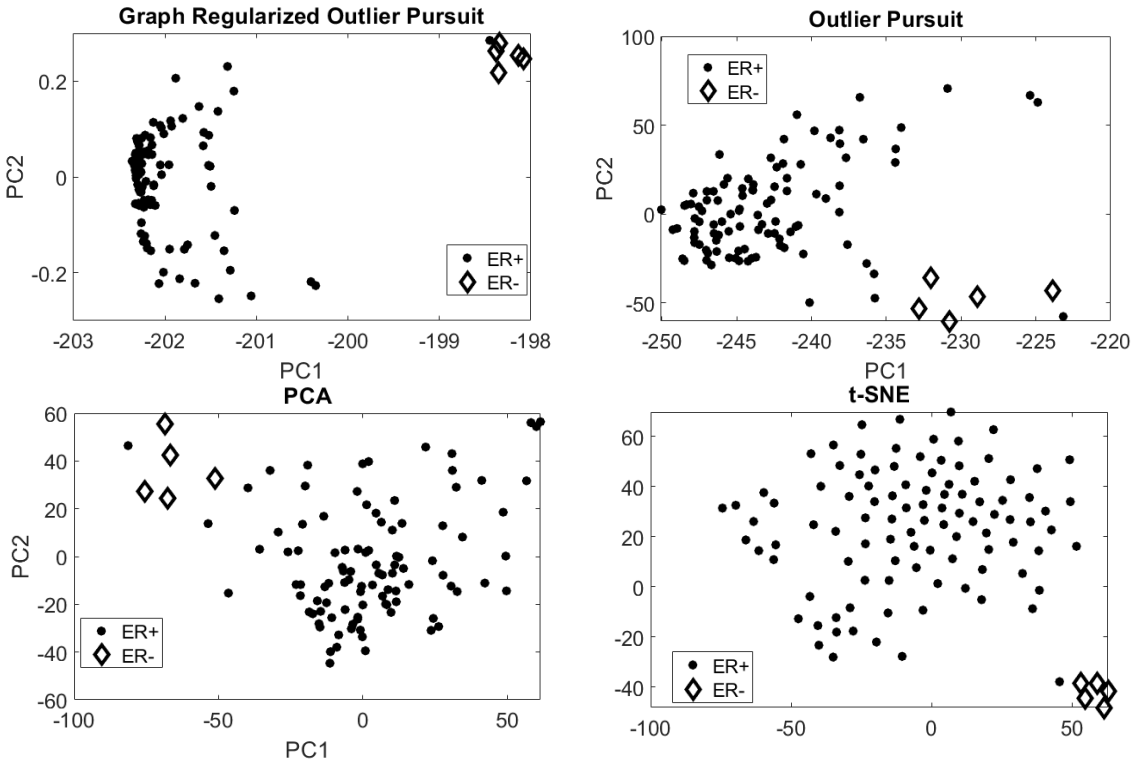


Figure 4.6: Visualization of 2-dimensional embedding for each dimensionality reduction method on a chosen instance of the breast cancer dataset. Figure shows the enhanced separation of main and outlier samples in the GOP embedding compared to OP, PCA and t-SNE

Low-Dimensional Embedding Outlier Detection and Visualization

Here, we show the F-score of k-means with two cluster centres on the low-dimensional projections of GOP, OP, PCA and t-SNE. We give as input to each dimensionality reduction technique, an instance of the single cell data after filtering it to its 1000 most variable genes. We find the F-score for all the 30 different instances of the single cell data. As seen from Figure 4.5(b), the greater F-score of GOP indicates that the outlier samples are better separated from the main samples in the lower dimensional embedding of GOP. In this case, the F-score of GOP is highest because it detects less false positives than all other methods. All the methods generally detect all the known 6 outliers. From Figure 4.8 we can see that GOP separates the outlier and main samples better than the other dimensionality reduction methods. This gives GOP an enhanced visualization property compared to other methods.

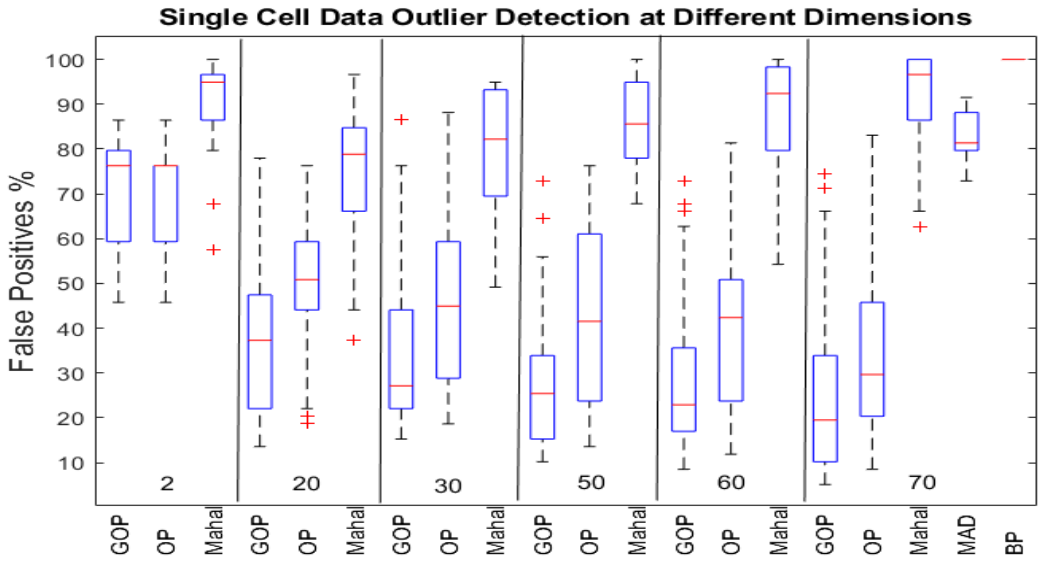


Figure 4.7: Boxplots comparing the number of false positives encountered to detect all 6 outliers in the 30 instances of the single cell dataset. We inspect the number of false positives at 6 different dimensions. We note that the performance of the Gaussian density method improves by increasing the dimensions from 2 to 20, as this adds more useful information to the dataset. However, it starts to degrade when increasing further. Moreover, we can see that the outlier detection performance of the robust subspace methods improves with the increase in dimensionality. Furthermore, we can see that GOP detects less median of false positives than OP at every dimension chosen.

4.7 Comparing Outlier Detection Performance of GOP and RPCAG

We should emphasize that there is a method closely related to GOP, which is Robust PCA on graphs (RPCAG) by [58]. The main difference between these two methods is the model of sparsity of the reconstruction matrix C. In [58] the reconstruction matrix is modelled by a l_1 norm which induces overall sparsity of the whole matrix. This model is more suitable to images which is what they demonstrate their results on. In the case of gene expression data a model of column sparsity fits more efficiently the model of outliers. The main algorithmic difference between GOP and Shahid's robust PCA model is in the update of the C matrix, step 5 of Algorithm 8. We show here the enhanced outlier detection performance of Graph regualrized Outlier Pursuit (GOP) and the Robust PCA on Graphs (RPCAG) formulated by [58]. We use the 30 randomly sampled instances of the breast cancer data and the single cell dataset. We compare the outlier detection performance using the $\hat{C}$ method, by sorting the l_2 norms of the columns of $\hat{C}$ and record the number of false positives before all known

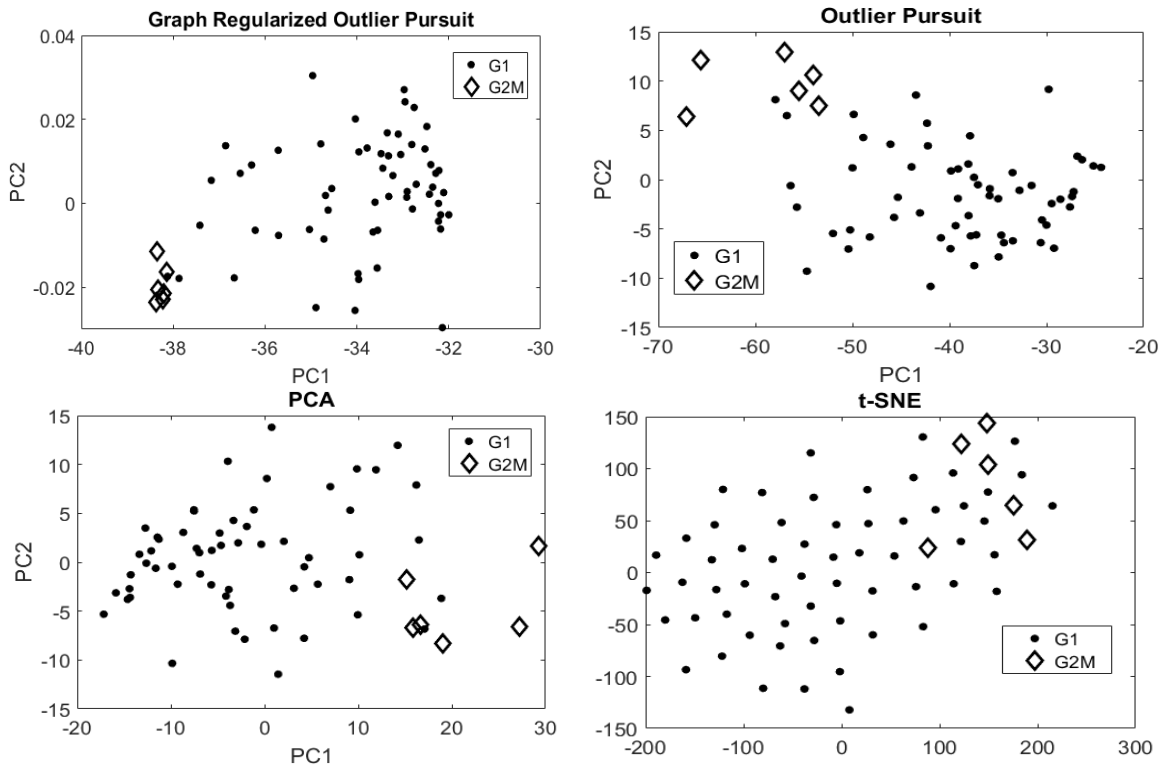


Figure 4.8: 2-dimensional visualization of the dimensionality reduction methods for a specific instance of the single cell dataset. Figure shows the enhanced visualization property of GOP compared to OP, PCA and t-SNE.

outliers are found. For both datasets the genes are filtered to retain the 200 most variable genes across samples. Figure 4.9 shows that GOP finds less false positives compared to RPCAG on both datasets.

4.8 GOP and OP Convergence

For both GOP and OP algorithms we inspect the convergence of their objective functions. Figure 4.10 and Figure 4.11 show the convergence of GOP and OP on the colon cancer, breast cancer, and single cell datasets. In Figure 4.10 it is evident that the ADMM algorithm (Algorithm 8) formulated to minimize the GOP objective function takes steps in the direction that reduces the objective function value. In Figure 4.11 it is also seen that the APG algorithm (Algorithm 9) formulated to minimize the OP objective function succeeds in reducing the objective function value at each iteration of the algorithm. Both figures prove experimentally that the chosen algorithms converge.

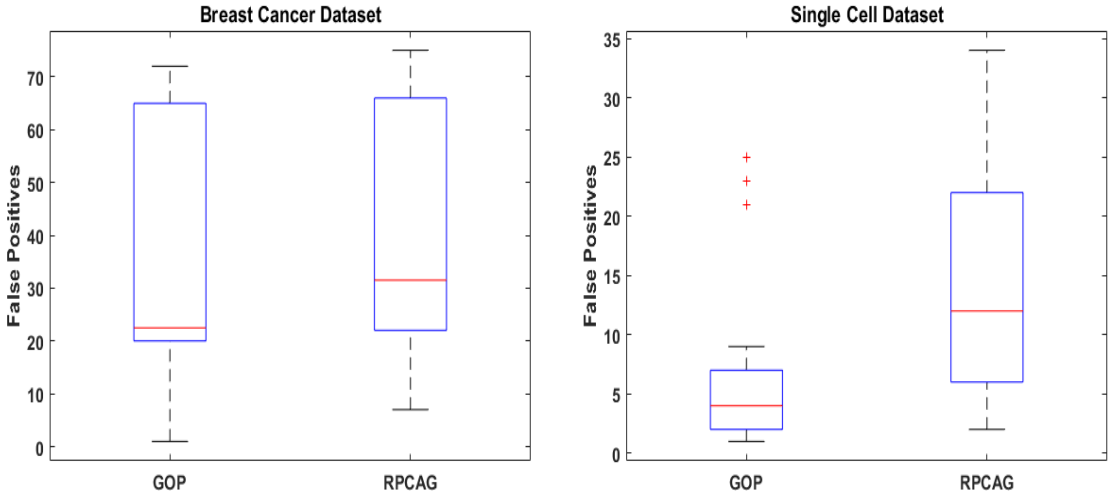


Figure 4.9: Comparison of outlier detection performance of GOP against RPCAG. Boxplots comparing the number of detected false positives recorded to detect all outliers in the 30 instances of both the breast cancer and single cell dataset. We can see that the GOP boxplot has less median false positives and a much narrower range when applied to the single cell dataset.

4.9 Discussion

The graph regularized method introduced in this work, to detect outlying samples, has more features compared to similar work found in the literature. With GOP we can detect outliers in two distinct ways: outlier ranking through the $\hat{C}$ matrix and clustering through the $\hat{L}$ matrix. Moreover, GOP is also a dimensionality reduction technique which makes it visualizable in a 2-dimensional space. Other methods such as [182] and [183] only devised an outlier ranking procedure by taking measures of global similarity between samples. This makes their method only capture outliers, but makes it harder to identify different subgroups. The identification of subgroups is leveraged by clustering techniques more than outlier ranking techniques. There are previous papers that used clustering techniques to identify similarity of samples in gene expression data, which are reviewed in [184]. However, they do not give the capability to visualize the data, and do not give an outlyingness ranking of the samples. To the best of our knowledge, GOP in the only method that combines both outlier ranking of samples with clustering and visualization.

4.10 Conclusion

In this chapter, we have developed an outlier detection framework for functional genomics data using structured low rank matrix approximation methods. We have

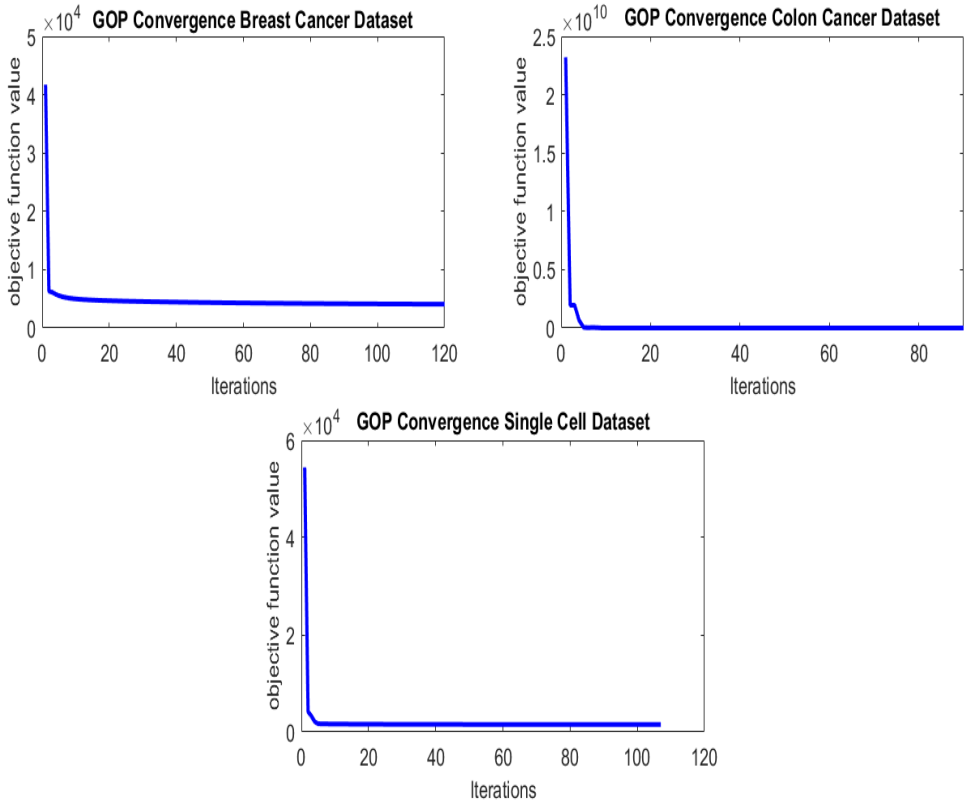


Figure 4.10: GOP objective function value with respect to number of iterations. The figure shows that the ADMM algorithm formulation for GOP is able to minimize the objective function.

explored two ways of extracting outliers from the decomposition of the data matrix into a low rank and column sparse matrix. Furthermore, we have shown that a better outlier detection is gained by including a regularizer based on the graph Laplacian of the data. Using transcriptomic data from bulk and single-cell measurements, we show that GOP reliably detects injected outliers, particularly when the graph regularizer is used. Most importantly, when compared to a density-based method of thresholding the Mahalanobis distance, and to traditional methods of measuring location and scatter (such as MAD and BP), the proposed method GOP does not fail with increasing dimensions. Thus finding the low rank subspace, in this case it has shown to circumvent the curse of dimensionality.

The graph regularizer used in this study is based on affinity (or neighbourhood) of the samples. However, this can be a convenient handle to inject prior knowledge into the problem domain. Thus, future work in this topic can be focused on the use of archived prior knowledge (interaction networks of the resulting proteins, for example) as regularizers.

In this chapter we only looked at genomic datasets with a single-view. In the next chapter we will expand this work by extending the GOP model to be able to take

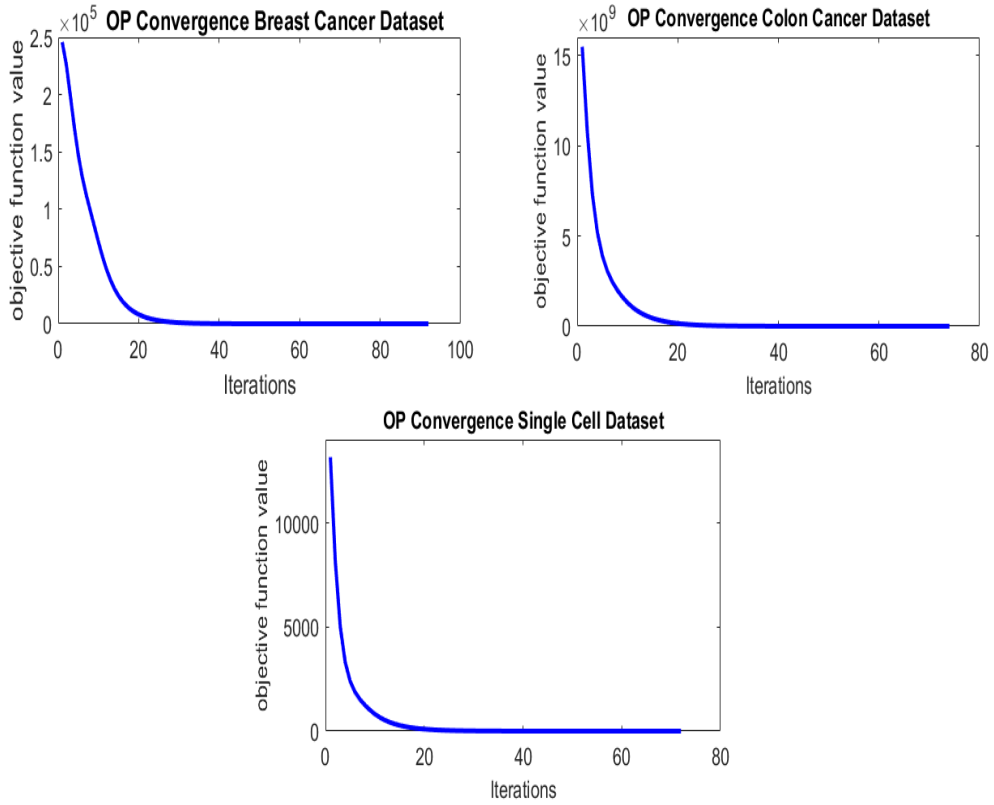


Figure 4.11: OP objective function value versus number of iterations. We can see from the figure that the OP objective function is minimized by the APG algorithm.

into account genomic datasets with multiple views.

Chapter 5

Convex Multi-View Clustering via Robust Low Rank Approximation with Application to Multi-*Omic* Data

Recent advances in high throughput technologies have made large amounts of biomedical *omics* data accessible to the scientific community. Single-*omic* data clustering has proved its impact in the biomedical and biological research fields. Multi-*omic* data clustering and multi-*omic* data integration techniques have shown improved clustering performance and biological insight. Cancer subtype clustering is an important task in the medical field to be able to identify a suitable treatment procedure and prognosis for cancer patients. State of the art multi-view clustering methods are based on non-convex objectives, which only guarantee non-global solutions that are high in computational complexity. Only a few convex multi-view methods are present. However, their models do not take into account the intrinsic manifold structure of the data. In this chapter, we introduce a convex graph regularized multi-view clustering method that is robust to outliers. We compare our algorithm to state of the art convex and non-convex multi-view and single-view clustering methods, and show its superiority in clustering cancer subtypes on publicly available cancer datasets from the TCGA repository. We also show our method's better ability to potentially discover cancer subtypes compared to other state of the art multi-view methods.

5.1 Introduction

Recent advances in high throughput sequencing technologies have made available large amounts of biomedical data consisting of measurement of genomic features across mul-

tiple *omic* scales. The different measurements across *omic* features, when combined together, form multi-*omic* datasets. Multi-*omic* data has been recently used to efficiently visualize and cluster cancer subtypes [48]. Clustering for biomedical data is a useful pattern discovery technique, which is the initial step taken in data exploration. Clustering is especially of great use in the emerging field of precision medicine in discovering cancer subtypes [47]. Separately clustering each *omic* has the capability of finding patterns in the data. However, using several *omics* for integrative clustering on the same group of samples, has the prospect to expose more detailed structures, that are not revealed by examining only a single-*omic* measurement. For example, it has been shown that cancer subtypes can be better defined when integrating both DNA methylation and gene expression [46, 185]. Cancer is a group of diseases caused by DNA alterations that change cell behaviour, which causes malignancy and uncontrolled growth. General treatments are challenging to develop due to the high genetic heterogeneity of this disease [186]. The field of cancer multi-*omics* has the aim to discover potential subtypes and their affiliated molecular biomarkers, that can be used for more individualised treatment and prognosis. Cancer multi-*omic* datasets consist of measuring different molecular parameters which include RNA expression, microRNA expression, DNA methylation and Protein expression, etc. These are examples of expression datasets taken at different *omic* levels, such as: (transcriptomics, epigenomics, and proteomics). While inference of cellular function or state from any one of these *omics* is easy to carry out, and dominates much of research reported in literature, cellular regulation is complex and combined analysis can reveal more information. For example, genes that are transcribed (DNA to mRNA) are not always translated into protein. The mRNA is held (for example, in structures like P-bodies) and is translated only when it is needed. Similarly, proteins may be synthesized at different rates from the corresponding mRNA by different numbers of ribosomes binding to them. Where disruptions to such regulation is the cause of disease, analysis at any one level can lead to misleading pictures.

In the machine learning community the problem of integrating information from different data types to achieve a joint clustering solution is called multi-view clustering. Multi-view clustering acts on multi-view data, where multi-*omic* datasets are a specific type of this general category of datasets. The problem of unsupervised multi-view clustering has gained much interest in the machine learning research community. Multi-view clustering methods found in the literature encompass: canonical correlation analysis (CCA) [55], Co-Training Expectation Maximization (co-EM) [49], multi-view normalized cut [187], co-regularized multi-view spectral clustering [56], multi-view neighbouring preserving projections [188], CCA regularized with common source graph [50] and Multi-view Non-Negative Matrix Factorization (Multi-NMF) [51]. CCA method in [55] and multi-view spectral clustering of [56], showed that finding a common latent representation between different views can enhance the

clustering performance. Moreover, Multi-NMF showed that learning a latent representation for each view, by constraining these representations to be similar to a 'consensus' representation, results in an improved clustering performance. The problem with these multi-view clustering methods is that they either work only with data that has two views [50, 55], or they optimize non-convex objective functions, that can only be solved by alternating optimization methods that converge to arbitrary local minima [49, 51, 56, 188].

In contrast, convex methods are found in the literature for the single-view case, where methods for subspace learning make use of convex loss functions [2, 189, 190]. These papers exploit a convex regularizer that reduces rank in place of constraining the dimension of the latent representation with a hard lower bound. Moreover, some authors [54, 162, 163] have approached the problem of multi-view subspace learning by using convex loss function formulations that look to find a common latent representation, which is then subsequently used for clustering. [54] finds a shared latent representation by minimizing a low rank regularized likelihood of a probabilistic model, which assumes a Gaussian distribution for real valued data. [163] finds a common latent representation by minimizing a regularized l_2 norm squared reconstruction error over the multiple views. Similarly, [162] minimizes a regularized reconstruction loss over the data views. Their reconstruction loss function is generic and can be any convex loss function; however it can only take into account two views. Both [54] and [163] are sensitive to outliers, as their loss functions minimize the l_2 norm squared and the Gaussian density function respectively, which are known to be fragile to even one outlier [2, 32]. All the previously mentioned convex multi-view methods do not take into account the local geometric structure of the data; a shortcoming that has been recently addressed by methods involving graph regularizers.

Graph regularizers have recently emerged in both the dimensionality reduction and data clustering areas of applied machine learning. That encode the geometric structure of the data in the form of a graph to be exploited by the learning models as an injection of structural knowledge [5, 50, 52, 53, 57, 58, 191, 192]. More specifically, multi-view subspace learning methods of Graph regularized Multiset Canonical Correlation Analysis (GrMCCA) [52] and Graph Multi-view Canonical Correlation Analysis (GMCCA) [53] are able to take into account more than two views. Both are formulated as optimizing a non-convex objective function and have closed form solutions that are computed by eigendecompositions. In the case of GMCCA [53] the graph regularization consists of a common source graph, meaning that it can not model the graph of each view separately; this is a limitation, as it can only be used for applications where common source graphs are available. Moreover, in GrMCCA [52] each view uses a separate graph regularizer. Furthermore, all previously mentioned multi-view subspace learning methods are fragile to even a small number of outliers. This is mainly because they minimize loss functions that have a quadratic term which will

amplify the errors produced by the outliers in the data [32].
In this chapter we address the above limitations by introducing Convex Graph regularized Robust Multi-view Subspace Learning (CGRMSL) for the problem of multi-view clustering. It is formulated with a convex objective function, that separately takes into account the manifold structure of each view of the data, is robust to outliers, and finds a shared latent representation of the data. We show that our method has superior clustering performance and is better able to visualize the data than other convex and non-convex multi and single-view subspace learning methods. We also demonstrated the ability of our model to detect potential subtypes more significantly than other state of the art multi-view methods. This is shown on genomic cancer datasets from the Cancer Genome Atlas (TCGA) repository [10].

5.2 Material and Methods

5.2.1 Convex Graph Regularized Robust Multi-View Subspace Learning

The algorithm we introduce in this chapter is called Convex Graph regularized Robust Multi-view Subspace Learning (CGRMSL). It utilizes more than one view to find a common latent representation, and takes into account complementary information from the different views to find a shared low-dimensional latent representation that will enhance clustering. The dataset to be considered has V views, with each view being represented by a matrix $M_v \in \mathbb{R}^{p_v \times n}$, with n being the number of samples that are common to the different views, and p_v is the number of features present in each view v. M_v consists of n samples arranged in columns with each sample having p_v features, expressed as: $M_v = [\boldsymbol{M}_v^{:,1}, \boldsymbol{M}_v^{:,2}, ..., \boldsymbol{M}_v^{:,n}]$. The objective of this method is to decompose each view M_v into a low rank matrix L_v, that gives a low-dimensional representation for the given view, and a column sparse matrix C_v that has non-zero columns in the samples that have high reconstruction errors, thus outliers. By modelling the reconstruction matrix C_v to be column sparse, our method detects (thus is robust to) outlier samples. Because, in the case of *omic* data, samples are more likely to be corrupt than a particular genomic feature across all data samples. The common latent representation is found by constraining the low rank matrices L_v to be similar to a shared matrix between all views, L^*. The graph which has nodes corresponding to samples, is constructed by first finding the K nearest neighbours of each sample, measured in Euclidean distance. Then for each sample we weight the edges to its K neighbours through the Gaussian kernel function $W_v^{i,j} = \exp(-\frac{||\boldsymbol{M}_v^{:,i} - \boldsymbol{M}_v^{:,j}||_2^2}{2\sigma^2})$. All other points that are not in the K nearest neighbours of the sample are weighted as zero. The matrix that incorporates the neighbouring and similarity information

for each view is the affinity matrix $W_v \in \mathbb{R}^{n \times n}$. Then, the graph Laplacian matrix $\Phi_v \in \mathbb{R}^{n \times n}$ is defined by $\Phi_v = D_v - W_v$. D_v is a diagonal matrix where each entry on its diagonal is the row sum of the corresponding row in W_v, $D_v^{i,i} = \sum_j W_v^{i,j}$. The CGRMSL optimization problem is as follows:

$$\min_{L_v, L*, C_v} \sum_{v=1}^{V} \Big(||L_v||_* + \lambda_v ||C_v||_{1,2} + \gamma_v ||L_v - L^*||_F^2 + \alpha \operatorname{tr}(L_v \Phi_v L_v^T) \Big). \quad \text{s.t: } M_v = L_v + C_v. \tag{5.1}$$

Where λ_v, γ_v, and α are real valued regularization parameters. The first term in the objective function, $||L_v||_*$ is the nuclear norm of L_v, which is the sum of its singular values. It induces low rankness in the matrix L_v. Minimizing the nuclear norm of a matrix is the closest convex surrogate of the (intractable and combinatorial) rank minimization problem [3,174]. The second term $||C_v||_{1,2}$ is the sum of the l_2 norms of the columns of C_v. It will induce column sparseness in the matrix C_v. The $l_{1,2}$ norm is the nearest convex surrogate to the number of non-zero columns in a matrix [3]. From the constraint of Problem 5.1, $M_v = L_v + C_v$, we can note that $C_v = M_v - L_v$ is the reconstruction error matrix for view v. Therefore, CGRMSL aims to model the outliers by inducing a column sparse structure to the reconstruction error matrix C_v, so that they are filtered out from the low rank matrix L_v. Both the nuclear norm and the $l_{1,2}$ norm have been used in the literature to induce low rankness and column sparseness respectively [2, 3, 58]. Both these norms have been used in our precursor work [5] to induce the structures of the low rankness and column sparseness of the single-view matrix decomposition ($M = L + C$), with an additional graph regularizer (same as the fourth term of Problem 5.1), to detect outliers and improve clustering quality of the recovered subspace. It has been demonstrated on single-view data of single cell genomics and cancer genomic data. Here, CGRMSL builds on and goes beyond our previous work [5] in being able to model multiple data views to find a shared latent space and is robust to outliers in each view. The third term constrains the low rank matrices of each view to be similar to a shared matrix L^*. The third term, for a specific view v can be rewritten as: $\sum_{i=1}^{n} ||\boldsymbol{L}_v^{:,i} - \boldsymbol{L}_{:,i}^*||_2^2$; this constraints each of the column vectors of the low rank matrix of a view, L_v, to be as close as possible in Euclidean distance to each corresponding column vector of L^*. Summing this over all views (as in Problem 5.1) will integrate the complementary information for all the available views to extract the common latent representation. To extract this we first compute the truncated Singular Value Decomposition (SVD) of L^*, $L^* = U\Sigma V^T$. Then, the common low-dimensional latent representation is the projection of L^* onto its truncated column space U, i.e. $Z = U^T L^*$. The fourth term is a graph regularizer on the low rank matrices. It preserves the intrinsic manifold information of the input data in the form of a graph. To best interpret the function of the graph regularization

term for a specific view v, $\text{tr}(L_v \Phi_v L_v^T)$, we can rewrite it in the following way:

$$\text{tr}(L_v \Phi_v L_v^T) = \frac{1}{2} \sum_{i,j=1}^{n} ||\boldsymbol{L}_v^{:,i} - \boldsymbol{L}_v^{:,j}||_2^2 W_{i,j},$$

The graph regularization term can be better interpreted now as $\frac{1}{2} \sum_{i,j=1}^{n} ||\boldsymbol{L}_v^{:,i} - \boldsymbol{L}_v^{:,j}||_2^2 W_{i,j}$. This function will impose structure in the recovered low rank matrix L_v, in the sense that if two points have high affinity in the original input space, the distance of the corresponding columns in L_v needs to be small. Problem 5.1 is a convex problem; it can be solved to find a stable global solution using the Alternating Direction Method of Multipliers (ADMM) optimization method [169].

5.2.2 CGRMSL Algorithm

Here we use ADMM to optimize the objective function in Problem 5.1. ADMM has been used to optimize problems in similar contexts of low rank and sparse matrix decompositions with an additional graph regularizer; in [5, 58]. The main difference between CGRMSL and our previous work [5] is the summation of the graph regularized decomposition of the input matrix ($M = L + C$) over all the available views, and the third term of Problem 5.1, that integrates the subspaces recovered from the different views. To solve CGRMSL using ADMM, we need to introduce an auxiliary variable, so that we can divide the objective function into four separate blocks. We rewrite the objective function of problem 5.1 as follows:

$$\begin{aligned} \min_{L_v, L*, C_v} \sum_{v=1}^{V} \Big(& ||L_v||_* + \lambda_v ||C_v||_{1,2} + \gamma_v ||Q_v - L^*||_F^2 \\ & + \alpha \, \text{tr}(Q_v \Phi_v Q_v^T) \Big). \\ & \text{s.t: } M_v = L_v + C_v \, , \; L_v = Q_v. \end{aligned} \tag{5.2}$$

Where Q_v with v from 1 to V are the auxiliary variables. Now we can define the augmented Lagrangian function of 5.2:

$$\begin{aligned} & \mathcal{L}(L_v, L^*, C_v, Q_v, Z_{1v}, Z_{2v}) = \sum_{v=1}^{V} \Big(||L_v||_* + \lambda_v ||C_v||_{1,2} \\ & + \gamma_v ||Q_v - L^*||_F^2 + \alpha \, \text{tr}(Q_v \Phi_v Q_v^T) + \langle Z_{1v}, M_v - L_v - C_v \rangle \\ & + \frac{p_1}{2} ||M_v - L_v - C_v||_F^2 + \langle Z_{2v}, Q_v - L_v \rangle \\ & + \frac{p_2}{2} ||Q_v - L_v||_F^2 \Big). \end{aligned}$$

Where the Frobenius inner product between two matrices $\langle X, Y \rangle$ is defined as $\text{tr}(X^T Y)$. To minimize the augmented Lagrangian with respect to each of the six variables, we use ADMM. The general form of the ADMM algorithm to solve CGRMSL is shown in Algorithm 10,

Algorithm 10 ADMM **Convex Graph Regularized Robust Multi-View Subspace Learning (CGRMSL)**

input: $M_v \in \mathbb{R}^{p_v \times n}$, λ_v, α, γ_v, $\Phi_v \ \forall \ v$)

1. initialize $L_v^0, L^{*,0}, C_v^0, Q_v^0 \ \forall \ v$ to random matrices.
2. $Z_{1v}^0 = M_v - L_v^0 - C_v^0$, $Z_{2v}^0 = Q_v^0 - L_v^0$. $p_1 = 1$ and $p_2 = 1$.
3. **repeat following until convergence**
4. **for v=1 to v=V**
5. $L_v^{k+1} = \underset{L_v}{\text{argmin}} \ \mathcal{L}(L_v, L^{*k}, C_v^k, Q_v^k, Z_{1v}^k, Z_{2v}^k)$
6. $C_v^{k+1} = \underset{C_v}{\text{argmin}} \ \mathcal{L}(L_v^{k+1}, L^{*k}, C_v, Q_v^k, Z_{1v}^k, Z_{2v}^k)$
7. $Q_v^{k+1} = \underset{Q_v}{\text{argmin}} \ \mathcal{L}(L_v^{k+1}, L^{*k}, C_v^{k+1}, Q_v, Z_{1v}^k, Z_{2v}^k)$
8. $L^{*k+1} = \underset{L^*}{\text{argmin}} \mathcal{L}(L_v^{k+1}, L^*, C_v^{k+1}, Q_v^{k+1}, Z_{1v}^k, Z_{2v}^k)$
9. $Z_{1v}^{k+1} = Z_{1v}^k + p_1(M_v - L_v^{k+1} - C_v^{k+1})$
10. $Z_{2v}^{k+1} = Z_{2v}^k + p_2(Q_v^{k+1} - L_v^{k+1})$

output: $\hat{L}_v = L_v^{k+1}$, $\hat{C}_v = C_v^{k+1}$, $\hat{L}^* = L^{*,k+1}$ when k is last iteration.

where Z_{1v}^k and Z_{2v}^k are the Lagrange multiplier matrices corresponding to the v^{th} view, and k is the iteration index. Steps 5 to 8 in Algorithm 10 have closed form solutions, derivations of which are shown below. The ADMM algorithm has been proven to converge to a global solution for convex objective functions [169].

CGRMSL ADMM Algorithm Derivation

Algorithm 10 requires to solve four sub-problems sequentially for each of the views v of the data from 1 to V.

1. $L_v^{k+1} = \underset{L_v}{\text{argmin}} \ \mathcal{L}(L_v, L^{*k}, C_v^k, Q_v^k, Z_{1v}^k, Z_{2v}^k)$.
2. $C_v^{k+1} = \underset{C_v}{\text{argmin}} \ \mathcal{L}(L_v^{k+1}, L^{*k}, C_v, Q_v^k, Z_{1v}^k, Z_{2v}^k)$.

3. $Q_v^{k+1} = \underset{Q_v}{\operatorname{argmin}}\ \mathcal{L}(L_v^{k+1}, L^{*k}, C_v^{k+1}, Q_v, Z_{1v}^k, Z_{2v}^k)$.

4. $L^{*k+1} = \underset{L^*}{\operatorname{argmin}}\ \mathcal{L}(L_v^{k+1}, L^*, C_v^{k+1}, Q_v^{k+1}, Z_{1v}^k, Z_{2v}^k)$.

The four sub-problems have closed form solutions. Their derivation is shown below.

Updating L_v (finding L_v^{k+1}):
$L_v^{k+1} = \underset{L_v}{\operatorname{argmin}}\ \mathcal{L}(L_v, L^{*k}, C_v^k, Q_v^k, Z_{1v}^k, Z_{2v}^k)$. Terms that are not related to L_v are constants and thus are discarded. This gives us:

$$\begin{aligned}
L_v^{k+1} &= \underset{L_v}{\operatorname{argmin}}\ ||L_v||_* + \frac{p_1}{2}\Big|\Big|L_v - (M_v - C_v^k + \frac{Z_{1v}^k}{p_1})\Big|\Big|_F^2 \\
&+ \frac{p_2}{2}\Big|\Big|L_v - (Q_v^k + \frac{Z_{2v}^k}{p_2})\Big|\Big|_F^2. \\
&= \underset{L_v}{\operatorname{argmin}}\ \frac{||L_v||_*}{p_1+p_2} + \frac{1}{2}\Big|\Big|L_v - \frac{p_1 R_{1v}^k + p_2 R_{2v}^k}{p_1+p_2}\Big|\Big|. \\
&= \operatorname{prox}_{\frac{||L_v||_*}{p_1+p_2}}(\frac{p_1 R_{1v}^k + p_2 R_{2v}^k}{p_1+p_2}).
\end{aligned}$$

Where $R_{1v}^k = M_v - C_v^k + \frac{Z_{1v}^k}{p_1}$ and $R_{2v}^k = Q_v^k + \frac{Z_{2v}^k}{p_2}$. The proximity operator of the nuclear norm function is the singular value soft-thresholding operator (this is derived in the Appendix A.2), which is defined as $\mathcal{D}_\epsilon(X) = U\xi_\epsilon(\Sigma)V^T$, where $X = U\Sigma V^T$ is the singular value decomposition (SVD) of X, and $\xi_\epsilon(\Sigma)$ is the soft-thresholding operator on the diagonal elements of Σ (as expressed in 3.18), with parameter ϵ. Now let $H_v = \frac{p_1 R_{1v}^k + p_2 R_{2v}^k}{p_1+p_2}$ and $p = \frac{p_1+p_2}{2}$. The update for L_v^{k+1} becomes: $L_v^{k+1} = \mathcal{D}_{\frac{1}{p}}(H_v)$.

Updating C_v:

$$\begin{aligned}
C_v^{k+1} &= \underset{C_v}{\operatorname{argmin}}\ \frac{\lambda_v}{p_1}||C_v||_{1,2} + ||C_v - (M_v - L_v + \frac{Z_{1v}^k}{p_1})||_F^2. \\
&= \operatorname{prox}_{\frac{\lambda_v}{p_1}||C_v||_{1,2}}(M_v - L_v^{k+1} + \frac{Z_{1v}^k}{p_1}).
\end{aligned}$$

The proximity operator of the $||C_v||_{1,2}$ function is the column-wise soft-thresholding operator (defined in equation 3.21). Now the update for C_v^{k+1} becomes:

$$C_v^{k+1} = \zeta_{\frac{\lambda_v}{p_1}}(M_v - L_v^{k+1} + \frac{Z_{1v}^k}{p_1}).$$

Updating Q_v:

$$
\begin{aligned}
Q_v^{k+1} &= \underset{Q_v}{\operatorname{argmin}} \alpha \operatorname{tr}(Q_v \Phi_v Q_v^T) + \langle Z_{2v}, Q_v - L_v \rangle \\
&+ \frac{p_2}{2} ||Q_v - L_v||_F^2 + \gamma_v ||Q_v - L^*||_F^2. \\
&= \underset{Q_v}{\operatorname{argmin}} \alpha \operatorname{tr}(Q_v \Phi_v Q_v^T) + \frac{p_2}{2} ||Q_v - (L_v^{k+1} - \frac{Z_{2v}^k}{p_2})||_F^2 \\
&+ \gamma_v ||Q_v - L^*||_F^2.
\end{aligned}
$$

Find first derivative and set to zero to find closed form solution for Q_v^{k+1}.

$$
Q_v^{k+1} = \Big(p_2(L_v^{k+1} - \frac{Z_{2v}^k}{p_2}) + \gamma_v L^*\Big)\Big(\alpha \Phi_v + (p_2 + \gamma_v) I\Big)^{-1}.
$$

Updating L^*:

$$
L^{*k+1} = \underset{L^*}{\operatorname{argmin}} \sum_{v=1}^{V} \gamma_v ||Q_v - L^*||_F^2.
$$

Find derivative and set to zero. $L^{*k+1} = \sum_{v=1}^{V} \frac{\gamma_v}{\theta} Q_v^{k+1}$,

where $\theta = \sum_{v=1}^{V} \gamma_v$.

Algorithm 10, although it is a 4-block ADMM it is guaranteed to converge because it satisfies **Condition 1** (In section 3.6.1). When problem 5.2 is compared to the general from of multi-block ADMM 3.32, it is seen that it has coefficient matrices A_1,A_2,A_3, $A_4 = I$. This means that any two coefficient matrices are orthogonal to each other, which satisfies **Condition 1** (Section 3.6.1).

5.2.3 Non-Robust version of CGRMSL

Here a version of CGRMSL which is not robust to outliers is introduced to evaluate the contribution of such robustness to the clustering task. Thus we introduce another multi-view subspace learning algorithm, Convex Graph regularized Multi-view Subspace Learning (CGMSL). In CGMSL the $l_{1,2}$ norm for computing the reconstruction errors is replaced by the standard Frobenius norm squared : $||M_v - L_v||_F^2$. The squared term present in this reconstruction error amplifies the outlier samples giving them much larger weight than non-outlier samples. This in turn skews the low-dimensional subspace towards the outliers making the CGMSL model not robust

to outliers. The optimization problem of CGMSL is as follows:

$$\min_{L_v, L*, C_v} \sum_{v=1}^{V} \Big(||L_v||_* + \lambda_v ||M_v - L_v||_F^2 + \gamma_v ||L_v - L^*||_F^2 + \alpha \operatorname{tr}(L_v \Phi_v L_v^T) \Big). \tag{5.3}$$

This objective function too is convex, thus a global solution can be found using ADMM. To optimize 5.3 with ADMM, we need to separate the objective function into three separate blocks by introducing auxiliary variables:

$$\min_{L_v, L*, Q_v} \sum_{v=1}^{V} \Big(||L_v||_* + \lambda_v ||M_v - Q_v||_F^2 + \gamma_v ||Q_v - L^*||_F^2 + \alpha \operatorname{tr}(Q_v \Phi_v Q_v^T) \Big). \quad \text{s.t: } L_v = Q_v. \tag{5.4}$$

Where Q_v for v from 1 to V are the auxiliary variables. Now we can define the augmented Lagrangian function of 5.4:

$$\mathcal{L}(L_v, L^*, Q_v, Z_{1v}) = \sum_{v=1}^{V} \Big(||L_v||_* + \lambda_v ||M_v - Q_v||_F^2 + \gamma_v ||Q_v - L^*||_F^2 + \alpha \operatorname{tr}(Q_v \Phi_v Q_v^T) + \langle Z_{1v}, Q_v - L_v \rangle + \frac{p_1}{2} ||Q_v - L_v||_F^2 \Big).$$

We then minimize the augmented Lagrangian with respect to the three variables separately. The ADMM algorithm for CGMSL is shown in Algorithm 11.

Algorithm 11 ADMM **Convex Graph Regularized Multi-View Subspace Learning (CGMSL)**

input: $M_v \in \mathbb{R}^{p_v \times n}$, λ_v, α , γ_v, $\Phi_v \; \forall \, v)$

1. initialize $L_v^0, L^{*,0}, Q_v^0 \; \forall \, v$ to random matrices.
2. $Z_{1v}^0 = Q_v^0 - L_v^0$. $p_1 = 1$.
3. **repeat following until convergence**
4. **for v=1 to v=V**
5. $L_v^{k+1} = \underset{L_v}{\operatorname{argmin}} \, \mathcal{L}(L_v, L^{*k}, Q_v^k, Z_{1v}^k)$
6. $Q_v^{k+1} = \underset{Q_v}{\operatorname{argmin}} \, \mathcal{L}(L_v^{k+1}, L^{*k}, Q_v, Z_{1v}^k)$
7. $L^{*k+1} = \underset{L^*}{\operatorname{argmin}} \, \mathcal{L}(L_v^{k+1}, L^*, Q_v^{k+1}, Z_{1v}^k)$
8. $Z_{1v}^{k+1} = Z_{1v}^k + p_1(Q_v^{k+1} - L_v^{k+1})$

output: $\hat{L}_v = L_v^{k+1}$, $\hat{L}^* = L^{*,k+1}$ when k is last iteration.

Steps 5 to 7 in Algorithm 11 have closed form solutions. Step 7 (Updating L^*) has the same closed form solution as CGRMSL (algorithm 10), steps 5 and 6 are different and their derivations are shown below.

CGMSL ADMM Derivation

Algorithm 2 requires to solve three sub-problems sequentially for each of the views v of the data from 1 to V.

1. $L_v^{k+1} = \underset{L_v}{\operatorname{argmin}} \, \mathcal{L}(L_v, L^{*k}, Q_v^k, Z_{1v}^k)$.
2. $Q_v^{k+1} = \underset{Q_v}{\operatorname{argmin}} \, \mathcal{L}(L_v^{k+1}, L^{*k}, Q_v, Z_{1v}^k)$.
3. $L^{*k+1} = \underset{L^*}{\operatorname{argmin}} \, \mathcal{L}(L_v^{k+1}, L^*, Q_v^{k+1}, Z_{1v}^k)$.

The 3rd sub-problem, updating L^*, has the same closed form solution as CGRMSL. The derivation of the first two sub-problems are shown below.

Updating L_v (finding L_v^{k+1}):
$L_v^{k+1} = \underset{L_v}{\operatorname{argmin}} \, \mathcal{L}(L_v, L^{*k}, Q_v^k, Z_{1v}^k)$. Terms that are not related to L_v are constants

and thus are discarded. This gives us:

$$
\begin{aligned}
L_v^{k+1} &= \underset{L_v}{\operatorname{argmin}} ||L_v||_* + \frac{p_1}{2}\Big|\Big|L_v - (Q_v^k + \frac{Z_{1,v}^k}{p_1})\Big|\Big|_F^2. \\
&= \underset{L_v}{\operatorname{argmin}} \frac{||L_v||_*}{p_1} + \frac{1}{2}\Big|\Big|L_v - (Q_v^k + \frac{Z_{1v}^k}{p_1})\Big|\Big|_F^2. \\
&= \operatorname{prox}_{\frac{||L_v||_*}{p_1}}(Q_v^k + \frac{Z_{1v}^k}{p_1}).
\end{aligned}
$$

Now let $H_v = Q_v^k + \frac{Z_{1v}^k}{p_1}$ and $p = \frac{1}{p_1}$. The update for L_v^{k+1} becomes: $L_v^{k+1} = \mathcal{D}_{\frac{1}{p}}(H_v)$.

Updating Q_v:

$$
\begin{aligned}
Q_v^{k+1} = \underset{Q_v}{\operatorname{argmin}}\, &\alpha \operatorname{tr}(Q_v \Phi_v Q_v^T) + \frac{p_1}{2}||Q_v - (L_v^{k+1} - \frac{Z_{1v}^k}{p_1})||_F^2 \\
&+ \gamma_v||Q_v - L_v^*||_F^2 + \lambda_v||M_v - Q_v||_F^2.
\end{aligned}
$$

Find first derivative and set to zero to find closed form solution for Q_v^{k+1}.

$$
Q_v^{k+1} = \Big(p_1(L_v^{k+1} - \frac{Z_{1v}^k}{p_1}) + \gamma_v L^* + \lambda_v M_v\Big)\Big(\alpha\Phi_v + (p_1 + \gamma_v + \lambda_v)I\Big)^{-1}.
$$

5.3 Simulation Study

5.3.1 Data Simulation

In this subsection we evaluate our model CGRMSL on two synthetic datasets by comapring against GrMCCA [52] and CGMSL. The first synthetic datasets is generated by a mixture of Gaussians (convex shapes). The second synthetic dataset comprises of a mixture of non-convex shapes, namely a mixture of 'moons'. We will show that our model is capable of finding a shared latent space that takes into account all complementary information from the different data views. Furthermore, we will show that our model is robust to outliers by finding a shared latent space that is not affected by their presence.

The structure of the first synthetic dataset is comprised of two 3-dimensional views with each view containing three different classes, each view is generated by a mixture of three Gaussian densities. Both views are generated by $p(M_v)=\sum_{i=1}^{3}\frac{1}{3}\mathcal{N}(\boldsymbol{\mu}_v^i, \Sigma_v^i)$, $v = 1, 2$, where $\boldsymbol{\mu^i}$ and Σ^i are the mean vector and Covariance matrix of the i^{th} Gaussian. Each Gaussian generates 500 samples for one class. For the 1^{st} view the three classes: C1, C2, and C3 have Gaussian density parameters set as follows: $\boldsymbol{\mu}_1^1 = (1\ 2)^T$,

$\boldsymbol{\mu}_1^2 = (1\ 4)^T$ and $\boldsymbol{\mu}_1^3 = (6\ 6)^T$. For the 2nd view the three classes: C1, C2, and C3 are parametrized by: $\boldsymbol{\mu}_2^1 = (1\ 2)$, $\boldsymbol{\mu}_2^2 = (6\ 6)^T$ and $\boldsymbol{\mu}_2^3 = (1\ 4)^T$. For both views all covariance matrices are set to the identity matrix, and the third dimension is generated by concatenating to the samples from the 2-dimensional Gaussians a standard uniform random variable in the interval (0,0.5). Furthermore, for both views we inject two outliers deeper in the third dimension with coordinate vectors: $(2\ \ 4\ \ 1.5)^T$ and $(3\ \ 4\ \ -1.5)^T$. Figure 5.1 shows the input dataset structure of both views generated from a mixture of bivariate Gaussian densities.

The second dataset is also comprised of two 3-dimensional views with each view having a mixture of three classes with each class containing 500 samples. Each view is generated as follows. First, Three 2-dimensional 'moons' are generated. Then, the third dimension is formed by concatenating to the samples from the 2-dimensional 'moons' a standard uniform random variable in the interval (0,0.5). The two views are constructed to have complementary information to separate all the three classes, as done for the first synthetic dataset. For this dataset, fractions of the whole 1500 samples of the dataset are corrupted to generate the outliers. The fraction of outliers generated are : 0.1 %, 1%, 3 %, 5%, 7%, 10%, 12 %, 15%. The outlier samples are generated by following a "salt and pepper" corruption model.

For both datasets we demonstrate the synthetic example in a 3-dimensional setting. This is to illustrate visually that or method is succeeding in being robust to outliers and is able to extract the complementary information between views. However, our method in a realistic scenario is used on high-dimensional datasets. Such as cancer genomic datasets which will be investigated in this chapter.

5.3.2 Experimental setting and results

We first construct a K Nearest Neighbour graph for each view. Then, we compute the Gaussian kernel function W_v for each view by setting σ as the squared mean of the euclidean distances between all samples. Finally, the graph Laplacian matrices Φ_1 and Φ_2 are constructed from their corresponding W_v as described in Subsection 2.2.1. The shared latent space for both CGRMSL and CGMSL is found by computing Z as explained in Subsection 2.2.1. For GrMCCA the shared latent space is computed as described in [52], where Y_{shared} is computed as $Y_{\text{shared}} = Y_1 + Y_2$, and Y_v is the projection of M_v onto the eigenvectors solving the eigendecomposition problem formulated in [52]; $Y_v = P_v^T M_v$. The reconstruction error of each sample is computed to show how the outliers affect each method. For CGRMSL and CGMSL, the reconstruction error for each sample is computed by the l_2 norm of the error between the i^{th} sample $\boldsymbol{M}_v^{:,i}$ of the v^{th} view and its corresponding reconstruction from the shared latent space $\hat{\boldsymbol{L}}_{:,i}^*$: $e_v^i = ||\boldsymbol{M}_v^{:,i} - \hat{\boldsymbol{L}}_{:,i}^*||_2$ for $i = 1, 2..., n$. However, for GrMCCA finding the reconstructions of the shared latent space in the original data space is not feasible.

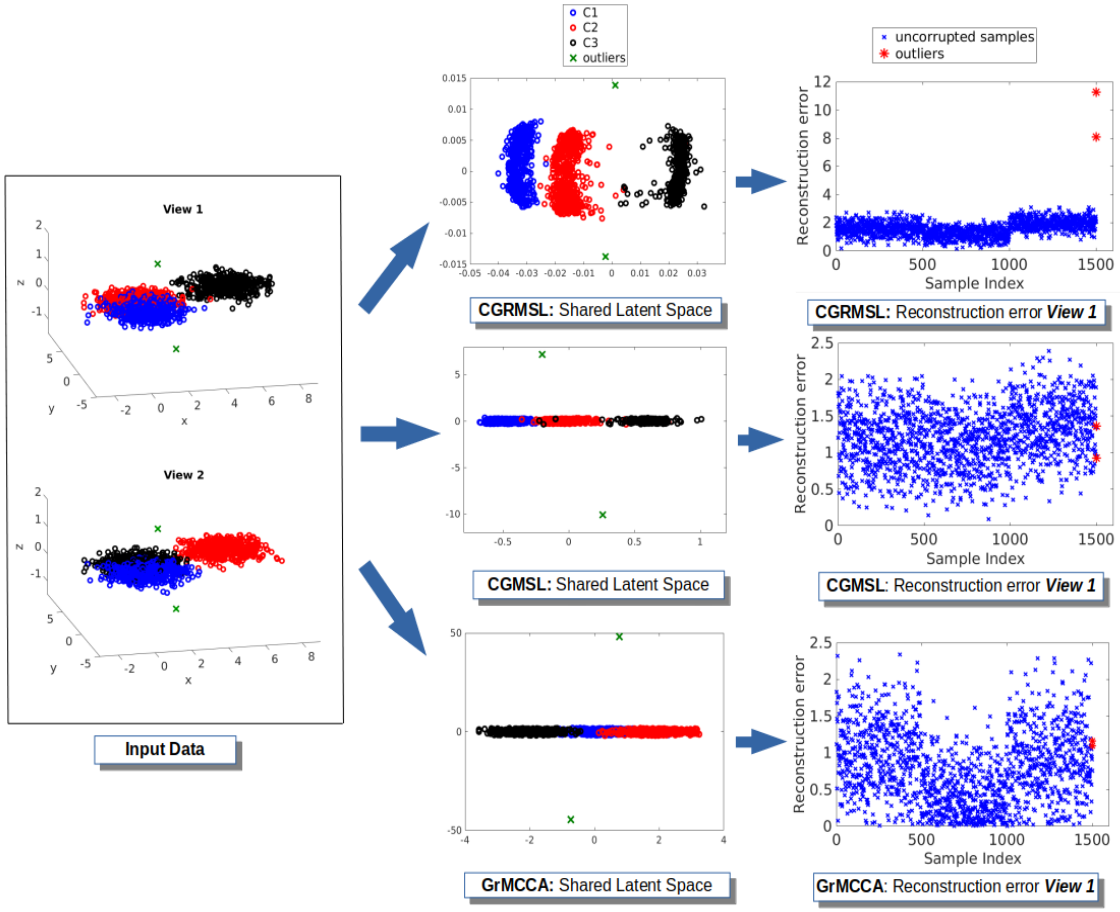


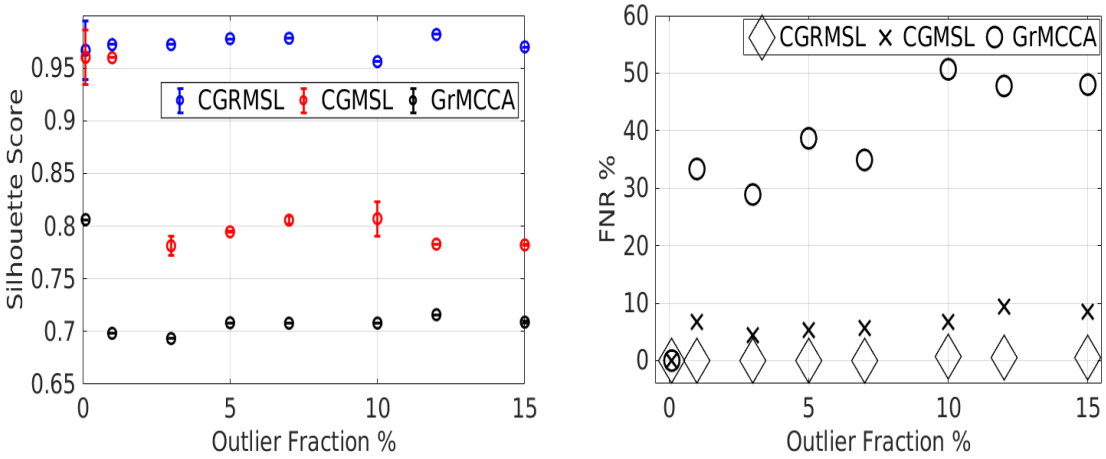
Figure 5.1: Synthetic example to compare the three algorithms: CGRMSL, CGMSL and GrMCCA. For each method the shared latent space and the reconstruction error for each sample are shown. We can see that CGRMSL shows robustness to outliers as expected. Whereas, CGMSL and GrMCCA are skewed to accommodate the outliers.

This is because it does not solve directly for a shared latent representation; instead it first solves for the projection vectors P_v of each view, then sums the projections of each view to create a shared latent representation. Hence, finding the reconstruction errors to investigate outliers can only be achieved by investigating reconstruction errors of each projected view. For GrMCCA the reconstruction for the v^{th} view is computed by $R_v = P_v Y_v$ and the reconstruction error for the v^{th} view is expressed by $e_v^i = ||\boldsymbol{M}_v^{:,i} - \boldsymbol{R}_v^{:,i}||_2$. Both synthetic datasets have been constructed to have three classes with information in both views, to be able to separate all three classes. However, each view alone has two out of the three classes with significant overlap and the third class being separate from the first two, as shown in Figure 5.1 (Input Data). Therefore, if the method used is capable of integrating the complementary information in both views, then the three classes should be all separated from each other in the shared latent space.

For the Gaussian mixture dataset, Figure 5.1 shows the shared lower dimensional latent space of the synthetic data and the reconstruction error of each sample on the 1^{st} view for CGRMSL, CGMSL and GrMCCA (2^{nd} view draws the same conclusion, only one is shown for simplicity). From Figure 5.1 we can see that the shared latent space of CGRMSL effectively separates the three different classes present in the two views. Whereas the shared latent space from GrMCCA shows less separability. We can also see from the reconstruction error of CGRMSL that the outliers have considerably higher reconstruction errors compared to all other samples. This indicates that the subspace of the shared latent space is not skewed to accommodate the outliers, thus proving the robustness of our method to outliers. On the other hand, for GrMCCA and CGMSL, the reconstruction error of the outliers are in the range of the other samples, showing that the outliers have skewed the shared latent subspace to accommodate them.

For the mixture of 'moons' dataset, we evaluate each method's ability to separate clusters in the recovered latent space, and the ability to detect the generated outliers by inspecting the reconstruction errors. This is done for the different outlier fractions mentioned in Subsection 6.3.1. The first step to evaluate the ability of a method to separate the three different classes is to compute cluster assignments on the extracted shared latent representation by using k-means clustering. Then, to evaluate the obtained clusters the silhouette score is computed, which is the mean of the silhouette values of each sample. The silhouette value of each sample is a measure of how similar a sample is to its own cluster compared to the other clusters. For the i^{th} sample, the smallest average distance of the i^{th} sample to all points in any other cluster is denoted as a_i, and the average dissimilarity between the i^{th} sample to all other data points in the same cluster is denoted as b_i. The silhouette value for the i^{th} sample is defined as $s_i = (a_i - b_i)/(\max(a_i, b_i))$. The silhouette ranges from -1 to 1. A silhouette score close to 1 indicates that clusters are well separated. Figure 5.2 (a) shows the errorbar of the silhouette scores of 50 runs of k-means computed on the shared latent spaces extracted from CGRMSL, CGMSL and GrMCCA. It is seen in the Figure that CGRMSL has the highest silhouette scores for all fraction of outliers compared to the other two non-robust methods.

The outlier detection performance is computed by the False Negative Rate (FNR). This computes the amount of outliers that have reconstruction errors overlapping with the reconstruction errors of the uncorrupted samples. Therefore, the reconstruction error threshold that is chosen to compute the FNR is the maximum of the reconstruction errors of the uncorrupted samples. Figure 5.2 (b) shows the FNR for all the three methods for the different outlier fractions. It is seen from the Figure that CGRMSL has the best outlier detection performance with and FNR starting at zero and remaining close to zero. Moreover, CGMSL and GrMCCA have an increasing overall trend of FNR when greater outlier fractions being generated.



(a) (b)

Figure 5.2: Performance of the three different Algorithms on the mixture of 'moons' dataset. (a) Displays the silhouette score of clusters computed on the shared latent representation of each method. (b) Shows the ability of detecting all the injected outliers by inspecting the reconstruction errors.

5.4 Comparisons

We compare CGRMSL to other convex methods, both single and multi-view. Another set of methods that we compare against are single and multi-view non-convex methods that have analytical solutions.

Single-view Subspace Learning (SSL). The aim of single-view learning is to find a low-dimensional latent representation of the input dataset, by taking into account only a single-view. We will compare against [189] which finds a sparse low-dimensional latent representation by minimizing a convex objective function.

Single-view non-convex. These methods act on a single-view and their objective functions are non-convex, but have closed form solutions based on eigendecompositions. These are Principal Component Analysis (PCA) [193] and Graph-Laplacian PCA (GPCA) [57].

Multi-view Subspace Learning. The aim of these methods is to find a common low-dimensional latent representation by using information from multiple views. The methods we compare against are the following convex methods: LRA Cluster from [54], Convex multi-view subspace learning (CMSL) from [162].

Multi-view non-convex. These methods are non-convex multi-view methods but have closed form solutions; they are: multi-view clustering via canonical correlation

analysis (CCA) [55] and GrMCCA [52].

Convex Graph regularized Robust Single-view Subspace Learning (CGRSSL). This method is a single-view subspace learning counterpart of our proposed method CGRMSL. It uses the $\hat{L_v}$ found from Algorithm 1 and from there finds the latent representation of the vth view by projecting $\hat{L_v}$ onto its truncated column space: $Z_v = U_v^T \hat{L}_v$. With the truncated SVD of $\hat{L}_v$ being, $\hat{L}_v = U_v \Sigma_v V_v^T$.

Convex Graph regularized Multi-view Subspace Learning (CGMSL). This is the non-robust version of CGRMSL described in Subsection 6.2.3. It replaces the robust $l_{2,1}$ norm of CGRMSL with the standard Frobenius norm squared for the reconstruction error.

5.5 Experimental Results Relevant to Cancer

In this section we validate our method against the other state-of-the-art multi-view and single-view methods described in Section 6.4. To evaluate our method we conduct experiments on five different TCGA cancer data types [10]: breast cancer (BRCA), esophageal cancer (ESCA), endometrioid cancer (UCEC), kidney renal clear cell carcinoma (KRCCC) and lung squamous cell carcinoma (LSCC). For BRCA, ESCA and UCEC pre-processed data is gathered from the UCSC Xena browser [179]. For KRCCC and LSCC the pre-processed data is provided by Wang et al. [154].
We first validate the clustering performance by finding a clustering assignment on the projection of the samples on the obtained subspace of all benchmark multi-view and single-view methods. Subsequently, the clustering assignments are compared to the given subtype labels from the TCGA clinical data for three of the five cancer types due to availability of subtype labels (Subsection 5.5.2). The three different cancer types are: BRCA, ESCA and UCEC. For the remaining LSCC and KRCCC cancer subtype labels are not present; therefore the objective is to find clusters that can be potential subtypes. Potential subtypes are discovered by performing a survival analysis and comparing how significantly survival times differ between samples in each cluster (Subsection 5.5.3). In Subsection 5.5.3 we compare only against the benchmark multi-view methods. Table 5.1 summarizes the different datasets used in this study.
In Table 5.1 the column 'features per view' describes the number of features retained per view for a specific cancer type. The number of features per view is chosen to be the smallest between the number of features of all views, because our method needs to have the same number of features for all the views. The features with the highest variability across samples are retained for each view. Features in the case of the five TCGA datasets can be: mRNAs for gene expression, DNA methylation

	Patients	features per view	views	subtype labels	subtypes
BRCA	292	250	2	YES	3
ESCA	194	300	2	YES	2
UCEC	112	1000	2	YES	2
KRCCC	122	329	3	NO	to be found
LSCC	106	352	3	NO	to be found

Table 5.1: Summary of the five TCGA cancer datasets used in this chapter.

sites for DNA methylation, and miRNAs for miRNA expression. BRCA, ESCA and UCEC have two different views that consist of measurements at two *omic* scales: gene expression (transcriptome) and DNA methylation (epigenome). KRCCC and LSCC have three views spanning two different *omic* scales: gene expression (transcriptome), DNA methylation (epigenome), and miRNA expression (transcriptome).

5.5.1 Parameter settings

We tuned parameters for all methods by conducting a parameter search. Afterwards, in the following subsections, the values with the best performance are recorded. For all methods with graph regularizers the value of K is chosen to be the one that gives the best cluster purity or p-value in the range [1, number of samples]. For GPCA α is chosen in the range [0.01,2]. η for GrMCCA is chosen in the range [0.5e-4,1e-2]. α for SSL and CMSL is chosen in the range [1, 100]. For CGRMSL α is chosen in the range of [0.1,100], $\gamma_v \, \forall v$ are in the range of [0.1,8] and $\lambda_v \, \forall v = \lambda$ with λ chosen in the range of [0.1,10]. For CGMSL the optimal α, $\gamma_v \, \forall v$ of CGRMSL are used and then λ is tuned in the range [0.1,10].

5.5.2 Clustering

Here we compare the proposed CGRMSL method against the benchmark single and multi-view methods described in Section 6.4. We compute the clustering performance on the learned representations found from each method. The clustering performance is evaluated on the three TCGA cancer types described previously. The problem that is investigated is cancer subtype clustering. For BRCA the three most common breast cancer subtypes are: Luminal, Basal, and Her2-enriched. For ESCA the subtypes are: Adenocarcinoma and squamous cell carcinoma. For UCEC the subtypes are: Serous and Endometrioid. For each cancer type we only retain the most variable genes across samples for each view as described in Table 5.1 third column; and after finding the common samples between both views the resulting datasets comprise of n =292, 194, 112 patients for BRCA, ESCA and UCEC respectively.
For our method and all of the benchmark methods described above, we evaluate

the clustering performance by measuring cluster purity. This has been used before in [194] to measure the performance of their multi-view clustering method for cancer subtype clustering. Cluster purity is a measure of how much the clusters contain a single class. It is calculated by first counting the number of data points from the most common class in each cluster. Then, the average is taken over all clusters. It is defined mathematically as:

$$\text{Purity} = \frac{1}{N}\sum_{i=1}^{C} \max_j |c_i \cap t_j|,$$

where c_i is the i^{th} cluster, t_j is the j^{th} class, C is the number of clusters and N is the number of data points. Cluster purity takes a value from 0 to 1, a value of 1 means that all the different classes present in the data have been perfectly identified as separate clusters. The higher the cluster purity the better the clustering has identified the different classes. Clustering is performed by k-means clustering which is run 50 times on the latent representation of each method; the average clustering purity of all 50 runs is reported in Tables 5.2 and 5.3 (clustering purity is multiplied by a 100 to give values from 0 to 100). The clustering purity of our method (CGRMSL) against all the benchmark single-view methods is shown in Table 5.2. We can see from Table 5.2 that CGRMSL gives higher clustering purity compared to all benchmark single-view methods applied to each view separately. It is also evident from Table 5.3 that CGRMSL gives better cluster purity compared to all other benchmark multi-view methods.

Another result worth highlighting is the capability of our method to visualize the three cancer types. We can see from Figure 5.3 that CGRMSL tightly places the different subtypes in distinct regions of the two-dimensional latent space. Moreover, Figure 5.3 also shows the misclassified samples when clustering on the CGRMSL subspace, and misclassified samples by k-means on the original space before dimensionality reduction.

	k-means v_1	k-means v_2	PCA v_1	PCA v_2	GPCA v_1	GPCA v_2	SSL v_1	SSL v_2	CGRSSL v_1	CGRSSL v_2	CGRMSL
BRCA	89.14±1.62	84.49 ±5.84	88.36	79.18±0.52	88.03±0.28	73.47 ±0.43	88.60 ±1.22	80.84 ±0.07	96.92	91.79±3.78	**97.26**
ESCA	93.68 ±0.22	91.18 ±0.28	93.75 ±0.17	91.25 ±0.26	93.50±0.25	90.72	93.30	90.21	96.90	93.81	**97.94**
UCEC	86.89±1.6	85.71	86.53±1.48	85.71	87.32±2.17	85.71	88.39	85.71	91.96	87.21 ±0.026	**96.43**

Table 5.2: cluster purity (average ±std) for single-view subspace learning methods, k-means on original space and CGRMSL. Readings with absent error bars have a std of zero for all 50 k-means runs. v_1 **is the gene expression view** and v_2 **is the DNA methylation view.**

	CMSL	LRA Cluster	CCA	GrMCCA	CGMSL	CGRMSL
BRCA	88.36	88.36	88.35	96.92	97.05 ±0.17	**97.26**
ESCA	95.36	95.88	94.85	95.36	95.57 ±0.24	**97.94**
UCEC	85.94	85.71	90.58±0.44	92.36 ±3.65	95.39 ±0.85	**96.43**

Table 5.3: cluster purity for multi-view subspace learning methods and our method (CGRMSL).

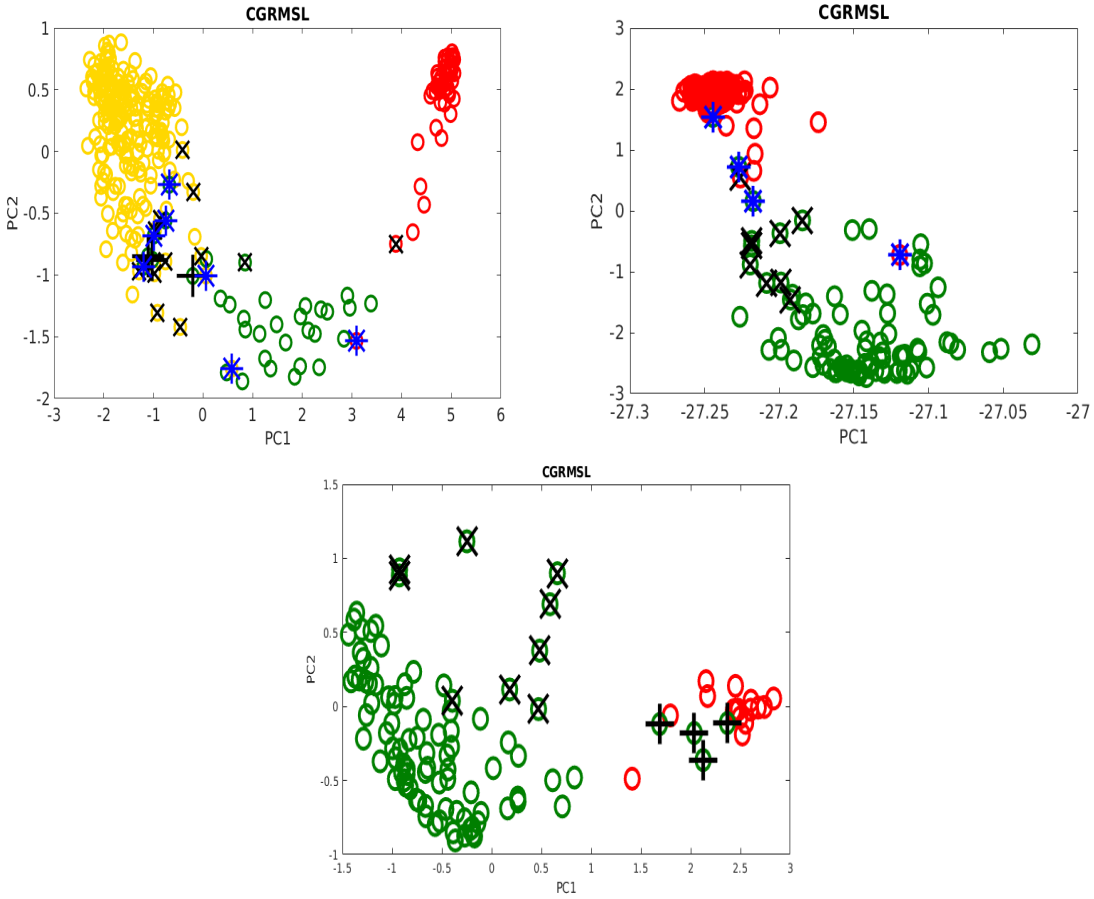
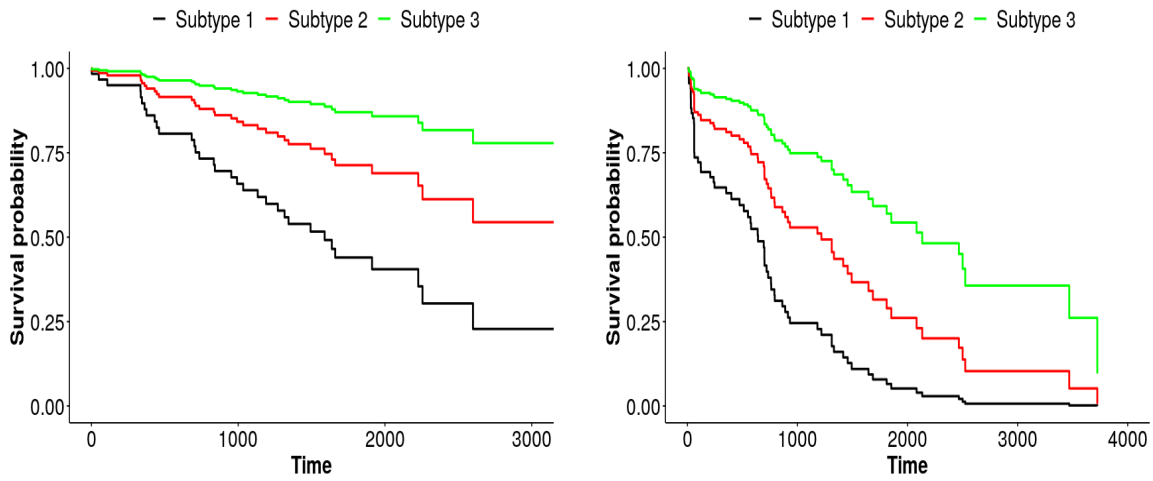


Figure 5.3: Visualization of CGRMSL for BRCA, ESCA and UCEC. Different subtypes are labelled by: green, red, and yellow 'o'. Misclassified samples by k-means on the CGRMSL subspace are labelled by a **black '+'**. Misclassified samples by k-means on the original space is labelled by a **black 'x'**. Samples that are both misclassified by k-means on the original space and the CGRMSL subspace are labelled by a **blue** '$\star$'.

5.5.3 Survival Analysis and Subtype Identification

Different cancer subtypes are expected to have significantly different survival times [48]. Here we apply our model to identify potential cancer subtypes by performing a survival analysis on the obtained clusters. This is performed on kidney renal clear cell



(a) KRCCC, **p-value** = 3.13e-4

(b) LSCC, **p-value** = 3.27e-5

Figure 5.4: Kaplan-Meier survival curves for KRCCC and LSCC. Shows distinct survival times of identified subtypes.

carcinoma (KRCCC) and lung squamous cell carcinoma (LSCC), described in Section 6.5. To measure how significantly the methods have identified different subtypes, the Cox survival p-value is used; it is computed using the Cox Wald test to measure whether the subtypes have significantly different survival times. A lower Cox p-value indicates that survival profiles among subtypes are more significantly different, and consequently potential subtypes might be discovered.

After projecting the samples onto the subspace given by CGRMSL we perform k-means clustering 50 times and report the lowest Cox Wald test p-value. The lowest p-value over the parameters of each method is reported. Here we cluster into three clusters as it gives the lowest p-value when compared to clustering into two and four clusters. We compare our method to other state of the art multi-view methods that can take into account more than two views; these results are shown in Table 5.4. It is seen from Table 5.4 that our method, CGRMSL, scores a more significant p-value compared to the other multi-view methods and the single-view version of our algorithm CGRSSL (for each view). Moreover, the table shows that the outlier fragile version of our algorithm, CGMSL, performs better than the other multi-view clustering methods. In addition, to show the distinct survival curves between identified subtypes, we display in Figure 5.4 the Kaplan-Meier survival curves for both cancer types using the subtypes identified by our method. From Figure 5.4 (a) and (b) it is evident that for both cancer types the three identified subtypes have significantly different survival profiles, a property that was not labelled.

	LRA Cluster	GrMCCA	CGMSL	CGRSSL v_1	CGRSSL v_2	CGRSSL v_3	CGRMSL
KRCCC	1.47e-2	1.2e-3	9.48e-04	4.30e-4	6.30e-2	2.36e-2	**3.13e-4**
LSCC	8.21e-4	2.71e-4	4.46e-5	5.1e-3	4.62e-2	1.1e-3	**3.27e-5**

Table 5.4: Cox Wald test p-value for all different multi-view methods. Parameters for each method are tuned and the best p-value is reported.

5.6 Conclusion

In this chapter we proposed an efficient convex multi-view clustering method that learns a common latent representation which takes into account the complementary information found in the separate views of the data. It is robust to outliers in the data and takes into account the intrinsic manifold structure of the data. We have shown that our method, CGRMSL, is superior to other convex and non-convex multi-view methods found in the literature, and also to single-view methods applied to each view separately. We have also shown that CGRMSL takes advantage of learning a shared matrix L^* as compared to only the single-view version of our method. We have demonstrated better clustering performance on an important biomedical problem: cancer subtype clustering, and the ability of our method to potentially discover new subtypes.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis presents two low rank matrix decomposition frameworks that are both robust to outliers, and take into account the intrinsic non-linear structure present in high-dimensional data. The matrix decomposition models are based on the decomposition of the high-dimensional data matrix into a low rank and column sparse matrix, also know as Outlier Pursuit (OP) by Xu et al. in [3]. In this thesis, starting from Chapter 1 Subsection 1.4.2, we show that the column sparse corruption model of OP is optimal for high-dimensional genomic data structures, as compared to the sparse corruption model (RPCA) of Candes et al. in [2], which is designed to efficiently model image data corruptions. The non-linear geometric structure of the data is modelled in the low rank approximation of the data, by adding a regularization to the decomposition model that smooths the low rank matrix onto the graph of data similarity. Thus, if the data lies on a low-dimensional manifold the continuity in the manifold would be preserved in the recovered $\hat{L}$ (low rank) matrix.
The two novel robust low rank decomposition methods proposed in this work are: 1) Graph Regularized Outlier Pursuit (GOP), and 2) Convex Graph regularized Multi-view Subspace Learning (CGRMSL).
Both of these methods are constructed as convex optimization problems, which gives guarantees of obtaining a global solution and computationally efficient algorithms. We summarize the conclusion of both methods as follows:

1. ***GOP conclusions***: GOP acts on single-view genomic datasets. We have tested GOP and OP on three high-dimensional gene expression datasets: colon cancer dataset, breast cancer dataset, and single-cell cell-cycle dataset. We have seen from our results that adding a graph regularizer enhances the separation in term of clustering performance between the outlier and normal samples in the recovered low rank matrix $\hat{L}$ compared to the recovered $\hat{L}$ from OP, and

traditional dimensionality reduction methods, such as PCA and t-SNE. This better clustering ability in the low rank matrix L justifies the better visualization property of GOP compared to other dimensionality reduction methods, as shown in Chapter 5. Moreover, with the addition to the graph regularizer, we get on average less number of false positives recorded before finding all outliers, compared to OP and traditional outlier detection framework based on density-based method; the latter are shown to fail with increasing dimensions.
We can see from the results in Chapter 5, that the similarity matrix can find the intrinsic structure in the data that would separate points belonging to different classes in the recovered low rank matrix $\hat{L}$, even when the number of points in each class is heavily imbalanced (in our case the minority class are set to be the outliers).
We can conclude from this report the following points:

- GOP and OP are suitable methods for high-dimensional datasets, as shown on the gene expression datasets.
- There are fast and efficient computational algorithms that can solve both problems in an efficient way.
- GOP has shown its effectiveness in outlier detection compared to the traditional density based methods that are affected by the high-dimensional data setting.
- GOP gives better separability in the recovered matrix $\hat{L}$, thus also better clustering and visualization capability when compared to: OP, standard PCA, and t-SNE.

2. ***CGRMSL conclusions***: CGRMSL is optimal for high-dimensional multi-view datasets. It has been applied on cancer genomic datasets gathered from TCGA to learn a common latent representation which takes into the account the complementary information found in separate views of the multi-*omic* data. It is robust to outliers in each view of the data, and models the manifold structure of each view using a graph regularization similar to GOP. We have shown in Chapter 6 that the proposed CGRMSL is superior to other convex and non-convex multi-view methods found in the literature and also to single-view methods applied to each view separately. We have also shown that CGRMSL takes advantage of learning a shared latent representation as compared to the single-view counterpart of the proposed CGRMSL method. Moreover, we have shown better clustering performance on cancer multi-*omic* datasets compared to state-of-the-art multi and single-view clustering methods. Finally, we demonstrate that our proposed method can more significantly discover new cancer subtypes compared to other state-of-the-art multi-view clustering methods.

6.2 Future Work

Possible directions of future work:

1. A direct improvement that can be done to the graph regulaized Outlier Pursuit model is in the graph construction itself. We can construct the affinity graph W from Protein-Protein Interaction (PPI) networks, and build a graph of gene-gene similarity. This will take the advantage of the similarity matrix to inject *apriori* biological knowledge which would find a better low-dimensional subspace of the gene expression data. Where clustering data, visualization, and outlier detection would take place. The expected results would be that injecting biological knowledge in the form of PPI networks would give a better representation of the underlying manifold that the data would live in. Some initial inspiration on how to build the gene-gene similarity matrix from the PPI data, is found in the following paper [195].

2. Another direction would be to use multi-*omic* manifold learning techniques such as CGRMSL proposed in this thesis to find a low-dimensional manifold of single cell multi-*omic* data that can capture cell differentiation. The manifold assumption tends to hold to a large extent with single-cell data, since cellular state spaces typically consists of smooth transitions which can be captured in the low-dimensional manifold [196]. Single cell RNA sequencing captures a snapshot of single-cells during this smooth transitions. This would motivate the use of manifold like algorithms other than algorithms that have linear assumptions of low dimensions.

3. Another extension of this work is inspired by Robust Deep Autoencoders (RDA) [197] and Graph regualrized Deep Autoencoders (GDA) [198]. RDA introduced by Zhou in [197] changes Robust PCA with column sparse corruptions (OP) in such a way that the nuclear norm of L is replaced by a deep autoencoder objective function. This augmentation would learn a non-linear manifold structre for the low rank matrix. In [198] the traditional autoencoder function is augmented by adding a graph regularizer which is used in image representation learning. A possible improvement is to combine the RDA framework with adding a graph regularizer to the autoencoder objective function. This can be applied to bulk genomic and single-cell data.

4. Another direction is to devise an on-line version of the robust PCA method with column sparse corruptions (OP). As we have shown in this thesis the column sparse corruption is optimal in detecting outlier samples in high-dimensional spaces. An on-line algorithm for Non-Negative Matrix Factorization (NMF), sparse coding, and general dictionary learning has been developed by Mairal et

al. in [199]. Thus following from their work, an on-line version of OP can be implemented.

5. Another promising direction of our work would be in precision medicine, or more specifically, precision oncology. An increasing amount of evidence in recent studies suggests that ageing diseases, such as cancer, Parkinsons, and many more can be better understood through mutated or dysregulated gene pathways or networks rather than individual mutations. Thus, publicly available biological networks can be integrated with the patient's genomic profiles to personalize treatments and to potentially discover new drug targets. Many machine learning methods that integrate biological networks as graphs within their model have emerged recently; they are extensivley reviewed in [200]. However, these methods can only model one genomic profile and can not integrate multiple views.

   A promising and obvious extensions from our work, is to extend CGRMSL (proposed in this thesis in Chapter 6) by injecting biological knowledge extracted from a biological network in the form of a graph between genes. Then, a graph regularizer based on the biological network can be added to the CGRMSL model. In this case, not only is information in different genomic views being integrated but also prior gene regulatory knowledge is being injected in the model. This method could potentially aid in drug target discovery for complex diseases.

Bibliography

[1] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems*, pages 2080–2088, 2009.

[2] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[3] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[4] Uri Alon, Naama Barkai, Daniel A. Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[5] Omar Shetta and Mahesan Niranjan. Robust subspace methods for outlier detection in genomic data circumvents the curse of dimensionality. *Royal Society Open Science*, 7(2), 2020.

[6] Todd R. Golub, Donna K. Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P. Mesirov, Hilary Coller, Mignon L. Loh, James R. Downing, Mark A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[7] Francois Bertucci, Sebastien Salas, Severine Eysteries, Valery Nasser, Pascal Finetti, Christophe Ginestier, Emmanuelle Charafe-Jauffret, Béatrice Loriod, Loic Bachelart, Jérôme Montfort, et al. Gene expression profiling of colon cancer by DNA microarrays and correlation with histoclinical parameters. *Oncogene*, 23(7):1377–1391, 2004.

[8] Karin Birkenkamp-Demtroder, Lotte L. Christensen, Harder S. Olesen, Casper M. Frederiksen, Päivi Laiho, Lauri A. Aaltonen, Søren Laurberg, Flemming B. Sørensen, Rikke Hagemann, and Torben F. Ørntoft. Gene expression in colorectal cancer. *Cancer Research*, 62(15):4352–4363, 2002.

[9] Tanya Barrett, Stephen E. Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F. Kim, Maxim Tomashevsky, Kimberly A. Marshall, Katherine H. Phillippy, Patti M. Sherman, Michelle Holko, et al. NCBI GEO: archive for functional genomics data setsupdate. *Nucleic Acids Research*, 41(D1):D991–D995, 2013.

[10] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Contemporary Oncology*, 19(1A):A68, 2015.

[11] Yawwani Gunawardana and Mahesan Niranjan. Bridging the gap between transcriptome and proteome measurements identifies post-translationally regulated genes. *Bioinformatics*, 29(23):3060–3066, 2013.

[12] Yawwani Gunawardana, Shuhei Fujiwara, Akiko Takeda, Jeongmin Woo, Christopher Woelk, and Mahesan Niranjan. Outlier detection at the transcriptome-proteome interface. *Bioinformatics*, 31(15):2530–2536, 2015.

[13] Shuangge Ma and Ying Dai. Principal component analysis based methods in bioinformatics studies. *Briefings in Bioinformatics*, 12(6):714–722, 2011.

[14] Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, 9(5):392–403, 2008.

[15] Ian Jolliffe. *Principal component analysis*. Springer-Verlag, 1989.

[16] Dov Greenbaum, Nicholas M. Luscombe, Ronald Jansen, Jiang Qian, and Mark Gerstein. Interrelating different types of genomic data, from proteome to secretome:'oming in on function. *Genome Research*, 11(9):1463–1468, 2001.

[17] Fábio CP Navarro, Hussein Mohsen, Chengfei Yan, Shantao Li, Mengting Gu, William Meyerson, and Mark Gerstein. Genomics and data science: an application within an umbrella. *Genome Biology*, 20(109), 2019.

[18] Mark W. Feinberg and Kathryn J. Moore. MicroRNA regulation of atherosclerosis. *Circulation Research*, 118(4):703–720, 2016.

[19] Wengong Si, Jiaying Shen, Huilin Zheng, and Weimin Fan. The role and mechanisms of action of microrna s in cancer drug resistance. *Clinical Epigenetics*, 11(25), 2019.

[20] Claire Mulvey, Bettina Thur, Mark Crawford, and Jasminka Godovac-Zimmermann. How many proteins are missed in quantitative proteomics based on ms/ms sequencing methods? *Proteomics Insights*, 3(61), 2010.

[21] Liang Liu, Yuanyuan Li, and Trygve O Tollefsbol. Gene-environment interactions and epigenetic basis of human diseases. *Current Issues in Molecular Biology*, 10(1-2):25–36, 2008.

[22] Stephen B. Baylin, Manel Esteller, Michael R. Rountree, Kurtis E. Bachman, Kornel Schuebel, and James G. Herman. Aberrant patterns of dna methylation, chromatin formation and gene expression in cancer. *Human Molecular Genetics*, 10(7):687–692, 2001.

[23] James Eberwine, Jai-Yoon Sul, Tamas Bartfai, and Junhyong Kim. The promise of single-cell sequencing. *Nature Methods*, 11(1):25–27, 2014.

[24] Xianwen Ren, Boxi Kang, and Zemin Zhang. Understanding tumor ecosystems by single-cell sequencing: promises and limitations. *Genome Biology*, 19(1):1–14, 2018.

[25] Mario L. Suvà and Itay Tirosh. Single-cell rna sequencing in cancer: lessons learned and emerging challenges. *Molecular Cell*, 75(1):7–12, 2019.

[26] Steen Knudsen. *Cancer diagnostics with DNA microarrays*. John Wiley & Sons, 2006.

[27] Geoffrey McLachlan, Kim-Anh Do, and Christophe Ambroise. *Analyzing microarray gene expression data*, volume 422. John Wiley & Sons, 2005.

[28] Alexei A. Sharov, Dawood B. Dudekula, and Minoru SH. Ko. A web-based tool for principal component and significance analysis of microarray data. *Bioinformatics*, 21(10):2548–2549, 2005.

[29] Matthew A. Hibbs, Nathaniel C. Dirksen, Kai Li, and Olga G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6(1):1–10, 2005.

[30] Ka Yee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[31] Shuangge Ma and Michael R. Kosorok. Identification of differential gene pathways with principal component analysis. *Bioinformatics*, 25(7):882–889, 2009.

[32] Peter J. Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[33] Lei Xu and Alan L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, 1995.

[34] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[35] Ahmad Barghash, Taner Arslan, and Volkhard Helms. Robust detection of outlier samples and genes in expression datasets. *Journal of Proteomics and Bioinformatics*, 9(02):38–48, 2016.

[36] Ricardo A. Maronna and Víctor J. Yohai. Robust estimation of multivariate location and scatter. *Wiley StatsRef: Statistics Reference Online*, pages 1–12, 2014.

[37] Norm A Campbell. Robust procedures in multivariate analysis i: Robust covariance estimation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29(3):231–237, 1980.

[38] Christophe Croux and Gentiane Haesbroeck. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87(3):603–618, 2000.

[39] Mia Hubert, Peter J. Rousseeuw, and Karlien Vanden Branden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

[40] Guoying Li and Zhonglian Chen. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 80(391):759–766, 1985.

[41] Héctor Climente-González, Chloé-Agathe Azencott, Samuel Kaski, and Makoto Yamada. Block hsic lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics*, 35(14):i427–i435, 2019.

[42] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[43] Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[44] Yunli Wang and Youlian Pan. Semi-supervised consensus clustering for gene expression data analysis. *BioData Mining*, 7(7), 2014.

[45] Sriparna Saha, Kuldeep Kaushik, Abhay Kumar Alok, and Sudipta Acharya. Multi-objective semi-supervised clustering of tissue samples for cancer diagnosis. *Soft Computing*, 20(9):3381–3392, 2016.

[46] Nimrod Rappoport and Ron Shamir. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research*, 46(20):10546–10562, 2018.

[47] Vinay Prasad, Tito Fojo, and Michael Brada. Precision oncology: origins, optimism, and potential. *The Lancet Oncology*, 17(2):e81–e86, 2016.

[48] Menglan Cai and Limin Li. Subtype identification from heterogeneous tcga datasets on a genomic scale by multi-view clustering with enhanced consensus. *BMC Medical Genomics*, 10(75), 2017.

[49] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

[50] Jia Chen, Gang Wang, Yanning Shen, and Georgios B Giannakis. Canonical correlation analysis of datasets with a common source graph. *IEEE Transactions on Signal Processing*, 66(16):4398–4408, 2018.

[51] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

[52] Yun-Hao Yuan and Quan-Sen Sun. Graph regularized multiset canonical correlations with applications to joint feature extraction. *Pattern Recognition*, 47(12):3907–3919, 2014.

[53] Jia Chen, Gang Wang, and Georgios B Giannakis. Graph multiview canonical correlation analysis. *IEEE Transactions on Signal Processing*, 67(11):2826–2838, 2019.

[54] Dingming Wu, Dongfang Wang, Michael Q Zhang, and Jin Gu. Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification. *BMC Genomics*, 16(1022), 2015.

[55] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 129–136, 2009.

[56] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2011.

[57] Bo Jiang, Chris Ding, Bio Luo, and Jin Tang. Graph-laplacian PCA: closed-form solution and robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3498, 2013.

[58] Nauman Shahid, Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Robust principal component analysis on graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2812–2820, 2015.

[59] Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(717), 2009.

[60] Emmanuel J. Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[61] Francis Ysidro Edgeworth. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375, 1887.

[62] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[63] Volker Roth. Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems*, pages 1169–1176, 2005.

[64] Volker Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006.

[65] Anup K. Ghosh, James Wanken, and Frank Charron. Detecting anomalous and unknown intrusions against programs. In *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98Ex217)*, pages 259–267, 1998.

[66] Anup K. Ghosh, Aaron Schwartzbard, and Michael Schatz. Learning program behavior profiles for intrusion detection. In *Workshop on Intrusion Detection and Network Monitoring*, volume 51462, pages 1–13, 1999.

[67] P. Barson, S. Field, N. Davey, G. McAskie, and R. Frank. The detection of fraud in mobile phone networks. *Neural Network World*, 6(4):477–484, 1996.

[68] Alexandre Nairac, Neil Townsend, Roy Carr, Steve King, Peter Cowley, and Lionel Tarassenko. A system for the analysis of jet engine vibration data. *Integrated Computer-Aided Engineering*, 6(1):53–66, 1999.

[69] Mahesh V. Joshi, Ramesh C. Agarwal, and Vipin Kumar. Mining needle in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 91–102, 2001.

[70] Mahesh V. Joshi, Ramesh C. Agarwal, and Vipin Kumar. Predicting rare classes: can boosting make any weak learner strong? In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 297–306, 2002.

[71] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[72] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004.

[73] Gary M. Weiss and Haym Hirsh. Learning to predict rare events in event sequences. In *KDD*, volume 98, pages 359–363, 1998.

[74] Ricardo Vilalta and Sheng Ma. Predicting rare events in temporal domains. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 474–481. IEEE, 2002.

[75] Jorma Laurikkala, Martti Juhola, and Erna Kentala. Informal identification of outliers in medical data. In *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, volume 1, pages 20–24, 2000.

[76] Jen-Pei Liu and Chung-Sing Weng. Detection of outlying data in bioavailability/bioequivalence studies. *Statistics in Medicine*, 10(9):1375–1389, 1991.

[77] Nong Ye and Qiang Chen. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International*, 17(2):105–112, 2001.

[78] Deepak Agarwal. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowledge and Information Systems*, 11(1):29–44, 2007.

[79] Garry Hollier and Jim Austin. Novelty detection for strain-gauge degradation using maximally correlated components. In *ESANN*, pages 257–262, 2002.

[80] Simon J. Hickinbotham and James Austin. Novelty detection in airframe strain data. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, pages 536–539. IEEE, 2000.

[81] Clay Spence, Lucas Parra, and Paul Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*, pages 3–10. IEEE, 2001.

[82] Lionel Tarassenko, Paul Hayton, Nicholas Cerneaz, and Michael Brady. Novelty detection for the identification of masses in mammograms. In *Fourth International Conference on Artificial Neural Networks*, pages 442–447, 1995.

[83] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[84] Kenji Yamanishi and Jun-ichi Takeuchi. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 389–394, 2001.

[85] Stephen Roberts and Lionel Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994.

[86] Simon Byers and Adrian E. Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584, 1998.

[87] G. Manson. Identifying damage sensitive, environment insensitive features for damage detection. In *Proceedings of the Third International Conference on Identification in Engineering Systems*, pages 187–197, 2002.

[88] Graeme Manson, Gareth S. Pierce, Keith Worden, Thomas Monnier, Philippe Guy, and Kathryn Atherton. Long-term stability of normal condition data for novelty detection. In *Smart Structures and Materials 2000: Smart Structures and Integrated Systems*, volume 3985, pages 323–334. International Society for Optics and Photonics, 2000.

[89] Graeme Manson, Gareth Pierce, and Keith Worden. On the long-term stability of normal condition for damage detection in a composite panel. In *Key Engineering Materials*, volume 204, pages 359–370. Trans Tech Publ, 2001.

[90] James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

[91] Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 251–261, 2003.

[92] Christopher Krügel, Thomas Toth, and Engin Kirda. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, pages 201–208, 2002.

[93] Lawrence L. Ho, Christopher J. Macey, and Ronald Hiller. A distributed and reliable platform for adaptive anomaly detection in IP networks. In *International Workshop on Distributed Systems: Operations and Management*, pages 33–46. Springer, 1999.

[94] Tom Fawcett and Foster Provost. Activity monitoring: noticing interesting changes in behavior. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, 1999.

[95] MJ. Desforges, PJ. Jacob, and JE. Cooper. Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 212(8):687–703, 1998.

[96] Dit-Yan Yeung and Calvin Chow. Parzen-window network intrusion detectors. In *Object Recognition Supported by User Interaction for Service Robots*, volume 4, pages 385–388. IEEE, 2002.

[97] Christopher M. Bishop. Novelty detection and neural network validation. *IEEE Proceedings Vision, Image and Signal processing*, 141(4):217–222, 1994.

[98] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to data mining*. Addison-Wesley, 2 edition, 2005. Chapter 2.

[99] Sigurour E. Guttormsson, RJ. Marks, MA. El-Sharkawi, and I. Kerszenbaum. Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion*, 14(1):16–22, 1999.

[100] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438, 2000.

[101] A. Kumar Jain and Dubes. *algorithms for clustering data*. Prentice-Hall Inc., 1988.

[102] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[103] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: a robust clustering algorithm for categorical attributes. In *Proceedings 15th International Conference on Data Engineering*, pages 512,529, 1999.

[104] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding topics in collections of documents: a shared nearest neighbor approach. In *Clustering and Information Retrieval*, pages 83–103. Springer, 2004.

[105] Rasheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski. Clustering approaches for anomaly based intrusion detection. 9, 2002.

[106] David J. Marchette. A statistical method for profiling network traffic. In *Workshop on Intrusion Detection and Network Monitoring*, pages 119–128, 1999.

[107] Adam Vinueza and G. Grudic. Unsupervised outlier detection and semi-supervised learning. *Technical Report CU-CS-976-04*, 2004.

[108] Ana M. Pires and Carla Santos-Pereira. Using clustering and robust estimators to detect outliers in multivariate data. In *Proceedings of the International Conference on Robust Statistics*, 2005.

[109] Mon-Fong Jiang, Shian-Shyong Tseng, and Chih-Ming Su. Two-phase clustering process for outliers detection. *Pattern recognition letters*, 22(6-7):691–700, 2001.

[110] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

[111] Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[112] Peter J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, 8(37):283–297, 1985.

[113] Laurie P. Davies et al. Asymptotic behaviour of S-estimates of multivariate location parameters and dispersion matrices. *The Annals of Statistics*, 15(3):1269–1292, 1987.

[114] DM. Titterington. Estimation of correlation coefficients by ellipsoidal trimming. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3):227–234, 1978.

[115] A.P. Dempster, M. Gasko-Green, et al. New tools for residual analysis. *The Annals of Statistics*, 9(5):945–959, 1981.

[116] Amrudin Agovic, Arindam Banerjee, Auroop Ganguly, and Vladimir Protopopescu. Anomaly detection using manifold embedding and its applications in transportation corridors. *Intelligent Data Analysis*, 13(3):435–455, 2009.

[117] Lucas Parra, Gustavo Deco, and Stefan Miesbach. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269, 1996.

[118] Haimonti Dutta, Chris Giannella, Kirk Borne, and Hillol Kargupta. Distributed top-k outlier detection from astronomy catalogs using the demac system. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 473–478. SIAM, 2007.

[119] Wei Wang, Xiaohong Guan, and Xiangliang Zhang. A novel intrusion detection method based on principle component analysis in computer security. In *International Symposium on Neural Networks*, pages 657–662. Springer, 2004.

[120] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[121] Olga Lyudchik. Outlier detection using autoencoders. Technical report, 2016.

[122] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015.

[123] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[124] Nikolaevich Tikhonov, Vasili Arsenin, and Fritz John. *Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.

[125] Gene H. Golub, Per Christian Hansen, and Dianne P. O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[126] Per Christian Hansen and Dianne Prost OLeary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, 1993.

[127] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[128] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Springer Science & Business Media, 2013.

[129] Jacob Abernethy, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. Low-rank matrix factorization with attributes. *arXiv preprint cs/0611124*, 2006.

[130] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th International Conference on Machine Learning*, pages 17–24. ACM, 2007.

[131] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[132] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(15):615–640, 2010.

[133] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[134] Zhenyue Zhang and Keke Zhao. Low-rank matrix approximation with manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1717–1729, 2012.

[135] Liang Tao, Horace HS. Ip, Yinglin Wang, and Xin Shu. Low rank approximation with sparse integration of multiple manifolds for data representation. *Applied Intelligence*, 42(3):430–446, 2015.

[136] Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Matrix completion on graphs. *arXiv preprint arXiv:1408.1717*, 2014.

[137] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[138] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[139] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[140] Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141), 2018.

[141] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, 2011.

[142] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015.

[143] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013.

[144] Kumardeep Chaudhary, Olivier B. Poirion, Liangqun Lu, and Lana X. Garmire. Deep learning–based multi-omics integration robustly predicts survival in liver cancer. *Clinical Cancer Research*, 24(6):1248–1259, 2018.

[145] Muxuan Liang, Zhizhong Li, Ting Chen, and Jianyang Zeng. Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(4):928–937, 2014.

[146] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[147] Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *International Conference on Machine Learning*, pages 352–360, 2013.

[148] Katherine A. Hoadley, Christina Yau, Denise M. Wolf, Andrew D. Cherniack, David Tamborero, Sam Ng, Max DM. Leiserson, Beifang Niu, Michael D. McLellan, Vladislav Uzunangelov, et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944, 2014.

[149] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

[150] Eric Bruno and Stephane Marchand-Maillet. Multiview clustering: a late fusion approach using latent models. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 736–737, 2009.

[151] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[152] Tin Nguyen, Rebecca Tagett, Diana Diaz, and Sorin Draghici. A novel approach for data integration and disease subtyping. *Genome Research*, 27(12):2025–2039, 2017.

[153] Ronglai Shen, Adam B. Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[154] Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11(3):333, 2014.

[155] Nora K. Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275, 2015.

[156] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[157] Nacim Fateh Chikhi. Multi-view clustering via spectral partitioning and local refinement. *Information Processing and Management*, 52(4):618–627, 2016.

[158] Bo Long, Philip S. Yu, and Zhongfei Zhang. A general model for multiple view unsupervised learning. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 822–833. SIAM, 2008.

[159] Moody T. Chu and J. Loren Watterson. On a multivariate eigenvalue problem, part I: algebraic theory and a power method. *SIAM Journal on Scientific Computing*, 14(5):1089–1106, 1993.

[160] Lei-Hong Zhang and Moody T. Chu. On a multivariate eigenvalue problem: II. global solutions and the gauss-seidel method. *Preprint*, 69, 2009.

[161] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[162] Martha White, Xinhua Zhang, Dale Schuurmans, and Yao-liang Yu. Convex multi-view subspace learning. In *Advances in Neural Information Processing Systems*, pages 1673–1681, 2012.

[163] Yuhong Guo. Convex subspace representation learning from multi-view data. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 387–393, 2013.

[164] Ryan Tibshirani. Gradient descent. University Lecture, 2016.

[165] L Vandenberghe. Proximal gradient method. University Lecture, 2010.

[166] Yurii Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24(3):509–517, 1988.

[167] 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.

[168] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix.

[169] 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*, 3(1):1–122, 2010.

[170] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1):57–79, 2016.

[171] Lan Jiang, Huidong Chen, Luca Pinello, and Guo-Cheng Yuan. Giniclust: detecting rare cell types from single-cell gene expression data with gini index. *Genome Biology*, 17(144), 2016.

[172] Lionel Tarassenko, Alexandre Nairac, N Townsend, I Buxton, and Peter Cowley. Novelty detection for the identification of abnormalities. *International Journal of Systems Science*, 31(11):1427–1439, 2000.

[173] Charu C. Aggarwal. Outlier analysis. In *Data Mining*, pages 237–263. Springer, 2015.

[174] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[175] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o($1/k^2$). *Sov. Math. Dokl*, 27(3).

[176] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proc. ICDM Foundation and New Direction of Data Mining workshop, 2003*, pages 172–179, 2003.

[177] Albert D. Shieh, Yeung Sam Hung, et al. Detecting outlier samples in microarray data. *Statistical Applications in Genetics and Molecular Biology*, 8(13).

[178] Benjamin M. Bolstad, Rafael A. Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[179] M. Goldman, B. Craft, M. Hastie, K. Repeka, A. Kamath, F. McDade, D. Rogers, A. Brooks, J. Zhu, and D. Haussler. The UCSC Xena platform for public and private cancer genomics data visualization and interpretation. Available at `https://xenabrowser.net/`(2020/01/09).

[180] Miriam Piles, Carlos Fernandez-Lozano, María Velasco-Galilea, Olga González-Rodríguez, Juan Pablo Sánchez, David Torrallardona, Maria Ballester, and Raquel Quintanilla. Machine learning applied to transcriptomic data to identify genes associated with feed efficiency in pigs. *Genetics Selection Evolution*, 51(10), 2019.

[181] Florian Buettner, Kedar N. Natarajan, F. Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J. Theis, Sarah A. Teichmann, John C. Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, 2015.

[182] Michael C. Oldham, Peter Langfelder, and Steve Horvath. Network methods for describing sample relationships in genomic datasets: application to huntingtons disease. *BMC Systems Biology*, 6(63), 2012.

[183] Concepcion Arenas, Claudio Toma, Bru Cormand, and Itziar Irigoien. Identifying extreme observations, outliers and noise in clinical and genetic data. *Current Bioinformatics*, 12(2):101–117, 2017.

[184] Rebecca Nugent and Marina Meila. An overview of clustering applied to molecular biology. In *Statistical Methods in Molecular Biology*, pages 369–404. Springer, 2010.

[185] Je-Keun Rhee, Kwangsoo Kim, Heejoon Chae, Jared Evans, Pearlly Yan, Byoung-Tak Zhang, Joe Gray, Paul Spellman, Tim HM. Huang, Kenneth P. Nephew, et al. Integrated analysis of genome-wide DNA methylation and gene expression profiles in molecular subtypes of breast cancer. *Nucleic Acids Research*, 41(18):8464–8474, 2013.

[186] Philippe L. Bedard, Aaron R. Hansen, Mark J. Ratain, and Lillian L. Siu. Tumour heterogeneity in the clinic. *Nature*, 501(7467):355–364, 2013.

[187] Dengyong Zhou and Christopher JC Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1159–1166, 2007.

[188] Novi Quadrianto and Christoph H. Lampert. Learning multi-view neighborhood preserving projections. In *International Conference on Machine Learning*, 2011.

[189] Xinhua Zhang, Yaoliang Yu, Martha White, Ruitong Huang, and Dale Schuurmans. Convex sparse coding, subspace learning, and semi-supervised extensions. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[190] Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

[191] Nauman Shahid, Nathanael Perraudin, Vassilis Kalofolias, Gilles Puy, and Pierre Vandergheynst. Fast robust PCA on graphs. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):740–756, 2016.

[192] Fanhua Shang, LC. Jiao, and Fei Wang. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, 45(6):2237–2250, 2012.

[193] DF. Frey and RA. Pimentel. Principal component analysis and factor analysis. 1978.

[194] Prabhakar Chalise and Brooke L. Fridley. Integrative clustering of multi-level omic data based on non-negative matrix factorization algorithm. *PloS One*, 12(5), 2017.

[195] Zena M. Hira, George Trigeorgis, and Duncan F. Gillies. An algorithm for finding biologically significant features in microarray data based on a priori manifold learning. *PloS One*, 9(3), 2014.

[196] Kevin R. Moon, Jay S. Stanley, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy. Manifold learning-based methods for analyzing single-cell RNA-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

[197] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674. ACM, 2017.

[198] Shijie Yang, Liang Li, Shuhui Wang, Weigang Zhang, and Qingming Huang. A graph regularized deep neural network for unsupervised image representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1203–1211, 2017.

[199] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1):19–60, 2010.

[200] Wei Zhang, Jeremy Chien, Jeongsik Yong, and Rui Kuang. Network-based machine learning and graph theory algorithms for precision oncology. *NPJ Precision Oncology*, 1(1):1–15, 2017.

[201] G. Alistair Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Appendix A

A.1 Column-Wise Soft-Thresholding Operator Derivation

(Column-Wise Soft-Thresholding Operator derivation): Given $X \in \mathbb{R}^{m\times n}$, we derive here the proximal operator $\text{prox}_t^h(X)$, when the function $h(X) : \mathbb{R}^{m\times n} \to \mathbb{R}$ is the $l_{1,2}$ norm; $h(X) = ||X||_{1,2} = \sum_{i=1}^{n} ||\boldsymbol{X}_{:,i}||_2$.

Proof

The separable sum property of proximal operators states that: if a function $h(\mathbf{x}_1, \mathbf{x}_2)$ is separable as, $h(\mathbf{x}_1, \mathbf{x}_2) = h_1(\mathbf{x}_1) + h_2(\mathbf{x}_2)$ then

$$\text{prox}_t^h(\mathbf{x}_1, \mathbf{x}_2) = \big(\text{prox}_t^{h_1}(\mathbf{x}_1), \text{prox}_t^{h_2}(\mathbf{x}_2)\big).$$

In the case of $h(X) = ||X||_{1,2} = \sum_{i=1}^{n} ||\boldsymbol{X}_{:,i}||_2$, $h_1, h_2, ...h_n$ are all the l_2 norm function $||.||_2$, hence we rewrite the proximity operator of the $l_{1,2}$ norm as

$$\text{prox}_t^{||.||_{1,2}}(X) = \big(\text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,1}), \text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,2}), ...\text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,n})\big)$$

then we can write this as,

$$\big[\text{prox}_t^{||.||_{1,2}}(X)\big]_i = \text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,i})$$

The proximity operator of the l_2 norm function, $\text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,i})$, is given as:

$$\text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,i}) = \begin{bmatrix} 0 & if \quad ||\boldsymbol{X}_{:,i}||_2 \leq t \\ \boldsymbol{X}_{:,i} - t\frac{\boldsymbol{X}_{:,i}}{||\mathbf{X}_{:,i}||_2} & if \quad ||\boldsymbol{X}_{:,i}||_2 > t \end{bmatrix}, \qquad (1)$$

this is the column-wise thresholding operator. The proximity operator $\text{prox}_t^{||.||_2}(\boldsymbol{X}_{:,i})$ of the l_2 norm can be derived from the **Moreau Decomposition**, which states that

for any vector $\mathbf{x}$:

$$\mathbf{x} = \text{prox}_t^h(\mathbf{x}) + t\text{prox}_{t^{-1}}^{h^*}(\mathbf{x}/t)$$

were h^* is the conjugate function of h. in the case where h is a norm its conjugate function is the indicator function of the dual norm-ball:

$$h(\mathbf{x} = ||\mathbf{x}||), \qquad h^*(\mathbf{x}) = \delta_B(\mathbf{x}) \quad \text{with } B = \{x|||x||_* \leq 1\}$$

($||x||_*$in this case is the dual norm) the indicator function is defined as:

$$\delta_B(\mathbf{x}) = \begin{bmatrix} 0 & if & \mathbf{x} \in B \\ \infty & if & \mathbf{x}_2 \notin B \end{bmatrix}.$$

In the case were h is the l_2 norm its dual norm is also the l_2 norm. Therefore, from the Moreau Decomposition we can find the proximity operator of the l_2 norm

$$\text{prox}_t^{||\cdot||_2}(\mathbf{x}) = \mathbf{x} - t\text{prox}_{t^{-1}}^{\delta_B}(\mathbf{x}/t) \tag{A.1}$$

Now we need to find the proximity operator of the indicator function $\delta_B(\mathbf{x})$, it is defined as

$$\begin{aligned} \text{prox}_{t^{-1}}^{\delta_B}(\mathbf{x}/t) &= \underset{\mathbf{u}}{\text{argmin}} \left(\delta_B(\mathbf{u}) + \frac{t}{2}||\mathbf{u} - \mathbf{x}/t||_2^2\right) \\ &= \underset{\mathbf{u}\in B}{\text{argmin}} \left(\frac{t}{2}||\mathbf{u} - \mathbf{x}/t||_2^2\right), \end{aligned}$$

getting rid of constant $\frac{t}{2}$ does not change minimization problem giving

$$\underset{\mathbf{u}\in B}{\text{argmin}} \left(||\mathbf{u} - \mathbf{x}/t||_2^2\right) = P_B(\mathbf{x}/t),$$

where $P_B(\mathbf{x}/t)$ is the projection of $\mathbf{x}/t$ onto closed convex set B. In the case where B is the unit norm ball, $\{\mathbf{x}|||\mathbf{x}||_2 \leq 1\}$, $P_B(\mathbf{x})$ is defined as

$$P_B(\mathbf{x}) = \begin{bmatrix} \mathbf{x} & if & ||\mathbf{x}||_2 \leq 1 \\ \frac{\mathbf{x}}{||\mathbf{x}||_2} & if & ||\mathbf{x}||_2 > 1 \end{bmatrix}.$$

Note that $P_B(\mathbf{x}/t)$ is the projection onto $B = \{\mathbf{x} : ||\mathbf{x}/t||_2 \leq 1\}$. Set B can be rewritten as the t-norm ball $B = \{\mathbf{x} : ||\mathbf{x}|_2 \leq t\}$, therefore we can say $P_B(\mathbf{x}/t) = P_{tB}(\mathbf{x})$. Where P_{tB} is the projection onto the t-norm ball whcih is defined as,

$$P_{tB}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} & if & ||\mathbf{x}||_2 \leq t \\ \frac{\mathbf{x}}{||\mathbf{x}||_2} & if & ||\mathbf{x}||_2 > t \end{bmatrix}.$$

Now substitution $\text{prox}_{t^{-1}}^{\delta_B}(\mathbf{x}/t) = P_{tB}(\mathbf{x})$ into A.1 we get,

$$\text{prox}_t^{||\cdot||_2}(\mathbf{x}) = \begin{bmatrix} 0 & if & ||\mathbf{x}||_2 \leq t \\ \mathbf{x} - t\frac{\mathbf{x}}{||\mathbf{x}||_2} & if & ||\mathbf{x}||_2 > t \end{bmatrix}$$

by recalling that we want to find the proximity operator for all the columns of X which are $\boldsymbol{X}_{:,i}$, the above definition is applied to all $\boldsymbol{X}_{:,i}$, giving the expression for column-wise thresholding operator (1).

A.2 Singular Value Soft-Thresholding operator Proof

(Singular Value Soft-Thresholding operator proof): Given a data matrix $X \in \mathbb{R}^{m\times n}$ and parameters $\mu, t > 0$, we prove here that the singular value soft-thresholding operator $\mathcal{D}_{\mu t}$ is the minimizer of

$$\mathcal{D}_{\mu t} = \underset{Y}{\text{argmin}} \frac{1}{2t}||X - Y||_F^2 + \mu||Y||_*. \tag{A.2}$$

Proof
Multiplying A.2 by a constant does not change the minimizer, we can rewrite it as

$$\mathcal{D}_{\mu t} = \underset{Y}{\text{argmin}} \frac{1}{2}||X - Y||_F^2 + \mu t||Y||_* \tag{A.3}$$

A.3 is strictly convex we can state that it has an existing unique minimizer , thus we need to prove that it is equal to $\mathcal{D}_{\mu t}(X)$. To do this we need to recall the subgradient optimality condition which states that, $\hat{Y}$ is a minimizer to A.3 if $\mathbf{0}$ is a subgradient of the subdifferential of A.3

$$\mathbf{0} = \partial_Y\big(\frac{1}{2}||X - \hat{Y}||_F^2 + \mu t||\hat{Y}||_*\big) \tag{A.4}$$

$$\mathbf{0} = \big(\hat{Y} - X\big) + \mu t\partial||\hat{Y}||_* \tag{A.5}$$

Where $\partial||\hat{Y}||_*$ is the set of subgradients of the nuclear norm. Now let $Y \in \mathbb{R}^{m\times n}$ be an arbitrary matrix and $U\Sigma V^T$ be its SVD. it is known [59, 201] that

$$\partial||Y||_* = \big(UV^T + W : W \in \mathbb{R}^{m\times n}, U^TW = 0, WV = 0, ||W||_2 \leq 0\big) \tag{A.6}$$

set $\hat{Y} = \mathcal{D}_{\mu t}(X)$ in order to show that $\hat{Y}$ minimizes A.3, Decompose X as $X = U_0\Sigma_0V_0^T + U_1\Sigma_1V_1^T$, where U_0, V_0 are the singular vectors associated with the singular values that are greater than μt. $U_1, V1$ are the singular vectors associated with the singular values that are smaller than or equal to μt. We have from the definition of the Singular Value Soft Thresholding operator,

$$\hat{Y} = U_0\big(\Sigma_0 - (\mu t)I\big)V^T$$

where I is the identity matrix. Therefore, rearranging A.5 as

$$X - \hat{Y} \in \mu t\partial||\hat{Y}||_* \tag{A.7}$$

and substitution X and $\hat{Y}$ gives

$$\underbrace{U_0\Sigma_0V_0^T + U_1\Sigma_1V_1^T}_{X} - \underbrace{U_0\big(\Sigma_0 - (\mu t)I\big)V_0^T}_{\hat{Y}} \in \mu t\partial||\hat{Y}||_* \tag{A.8}$$

$$\underbrace{U_0\Sigma_0V_0^T + U_1\Sigma_1V_1^T}_{X} - \underbrace{U_0\Sigma_0V_0^T + (\mu t)\,U_0V_0^T}_{\hat{Y}} \in \mu t\partial||\hat{Y}||_* \tag{A.9}$$

$$\frac{1}{\mu t}\big(U_1\Sigma_1V_1^T\big) + U_0V_0^T \in \partial||\hat{Y}||_* \tag{A.10}$$

Comparing A.10 to the definition of the subdifferential of the nuclear norm A.6. We have $W = \frac{1}{\mu t}\big(U_1\Sigma_1V_1^T\big)$. Now check if conditions inA.6 are met. by definition we can see that $U_0W = 0$ and $WV = 0$ and we know that the diagonal elements of Σ_1 have elements that are smaller than or equal to μt therefore, $||W||_2 \leq 1$. this proves that $X - \hat{Y} \in \mu t\partial||\hat{Y}||_*$ which concludes the proof.

A.3 Gaussian Noise Model for Classical PCA

Statement: *The noise model of PCA is Gaussian distributed.* In other words, we prove here that minimizing the l_2 norm squared of the reconstruction error of each sample is equivalent to maximising the log likelihood of the data, when the noise matrix $\mathcal{N}$ is sampled from a Gaussian distribution.

Proof
Classical PCA linear problem:

$$M = L + N,$$

We can think of this instance wise by taking each column separately. Then the linear problem becomes:

$$\boldsymbol{M}_{:,i} = \boldsymbol{L}_{:,i} + \boldsymbol{N}_{:,i} \quad \forall i,$$

where $i = [1, 2, ...n]$ and vector $\boldsymbol{N}_{:,i}$ is sampled from a multivariate Gaussian distribution with dimensionality p, $\mathbf{N}_{:,i} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$. Then the distribution of $\boldsymbol{M}_{:,i}$ becomes $\boldsymbol{M}_{:,i} \sim \mathcal{N}(\boldsymbol{L_i}, \sigma^2 I)$. Now we can write the log likelihood of the data M as:

$$\mathfrak{L}(M; L) = \log \prod_i p(\boldsymbol{M}_{:,i}|\boldsymbol{L}_{:,i})) = \sum_i \log(p(\boldsymbol{M}_{:,i}|\boldsymbol{L}_{:,i})).$$

Probability distribution of $\boldsymbol{M}_{:,i}$ is multivariate Gaussian with isotropic covariance:

$$p(\boldsymbol{M}_{:,i}|\boldsymbol{L}_{:,i}) = \frac{1}{\sqrt{((2\pi)^p \sigma^{2p})}} \exp\Big(-\frac{1}{2\sigma^2}(\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})^T (\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})\Big).$$

We want to maximize the log likelihood with respect to the parameter that we ant to estimate, in this case it is L. The objective function is written as:

$$\hat{L} = \underset{L}{\operatorname{argmax}} \sum_i \Big(\log(K) - \frac{1}{2\sigma^2}(\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})^T (\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})\Big),$$

where K is a constant term. By getting rid of the constants the argmax is unchanged. We can rewrite the previous expression as:

$$\hat{L} = \underset{L}{\operatorname{argmax}} - \sum_i \Big((\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})^T (\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i})\Big),$$

which is equivalent to minimizing the reconstruction error:

$$\hat{L} = \underset{L}{\operatorname{argmin}} \sum_i \Big(||\boldsymbol{M}_{:,i} - \boldsymbol{L}_{:,i}||_2^2\Big) = \underset{L}{\operatorname{argmin}} \, ||M - L||_F^2.$$

This concludes the proof.

A.4 Laplacian Noise Model for Robust PCA

Statement: *The noise model of Robust PCA is Laplacian distributed.* In other words, we prove here that minimizing the l_1 norm of the reconstruction error of each sample is equivalent to maximising the log likelihood of the data, when the noise matrix $\mathcal{N}$ is sampled from a Laplacian distribution.

Proof
Robust PCA linear model:

$$M = L + E,$$

We can think of this element-wise by taking each element separately. Then the linear problem for every element becomes:

$$M_{i,j} = L_{i,j} + E_{i,j} \quad \forall i,$$

where $i = [1, 2, ...n]$ and entries in E are sampled from a Laplace distribution, $E_{i,j} \sim \text{Laplace}(0, b)$. Then the distribution of $M_{i,j}$ becomes $M_{i,j} \sim \text{Laplace}(L_{i,j}, b)$.
Now we can write the log likelihood of the data M as:

$$\mathfrak{L}(M; L) = \log \prod_{i,j} p(M_{i,j}|L_{i,j})) = \sum_{i,j} \log(p(M_{i,j}|L_{i,j})).$$

The probability distribution of $M_{i,j}$ is a univariate Laplacian distribution:

$$p(M_{i,j}|L_{i,j}) = \frac{1}{\sqrt{2b}} \exp\Big(- \frac{1}{b}(|M_{i,j} - L_{i,j}|), \Big)$$

now we maximize the log likelihood with respect to L. The objective function is written as:

$$\hat{L} = \underset{L}{\text{argmax}} \sum_{i,j} \Big(\log(K) - \frac{1}{b}(|M_{i,j} - L_{i,j}|) \Big),$$

where K is a constant term. By getting rid of the constants the argmax is unchanged. We can rewrite the previous expression as:

$$\hat{L} = \underset{L}{\text{argmax}} - \sum_{i,j} \Big(|M_{i,j} - L_{i,j}| \Big),$$

which is equivalent to minimizing the l_1 reconstruction error:

$$\hat{L} = \underset{L}{\text{argmin}} \sum_{i,j} \Big(|M_{i,j} - L_{i,j}| \Big) = \underset{L}{\text{argmin}} \, ||M - L||_1.$$

This concludes the proof.

A.5 CCA; Deriving Eigenvalue Problem

The CCA objective function is:

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\operatorname{argmax}}\, \mathbf{w}_1 C_{12}\mathbf{w}_2 \\ &\text{s.t} \quad \mathbf{w}_1^T C_{11}\mathbf{w}_1 = 1, \quad \mathbf{w}_2^T C_{22}\mathbf{w}_2 = 1. \end{aligned}$$

The Lagrange multiplier of this problem is:

$$\mathcal{L}(\mathbf{w}_1, \mathbf{w}_2, \lambda_1, \lambda_2) = \mathbf{w}_1 C_{12}\mathbf{w}_2 - \lambda_1(\mathbf{w}_1^T C_{11}\mathbf{w}_1 - 1) - \lambda_2(\mathbf{w}_2^T C_{22}\mathbf{w}_2 - 1).$$

The Lagrange multiplier method takes the derivative of the Lagrange multiplier function with respect to each variable and equates it to 0. The derivative of the Lagrange multiplier function with respect to both $\mathbf{w}_1$ and $\mathbf{w}_2$ is:

$$\frac{\delta\mathcal{L}}{\delta\mathbf{w}_1} = 0; \quad C_{12}\mathbf{w}_2 - \lambda_1 C_{11}\mathbf{w}_1 = 0. \tag{A.11}$$

$$\frac{\delta\mathcal{L}}{\delta\mathbf{w}_2} = 0; \quad C_{21}\mathbf{w}_1 - \lambda_2 C_{22}\mathbf{w}_2 = 0. \tag{A.12}$$

From A.11 and A.12 we obtain:

$$C_{12}\mathbf{w}_2 = \lambda_1 C_{11}\mathbf{w}_1. \tag{A.13}$$

$$C_{21}\mathbf{w}_1 = \lambda_2 C_{22}\mathbf{w}_2. \tag{A.14}$$

To obtain the generalized eigenvalue problem to solve for $\mathbf{w}_1$, we rearrange equation A.14 to obtain the expression for $\mathbf{w}_2 = \frac{1}{\lambda_2}(C_{22})^{-1}C_{21}\mathbf{w}_1$. We substitute in equation A.13 to obtain:

$$C_{12}(C_{22})^{-1}C_{21}\mathbf{w}_1 = \eta C_{11}\mathbf{w}_1. \tag{A.15}$$

solving this is the generalized eigenvalue problem will give the optimal projection direction $\mathbf{w}_1$.
To obtain the generalized eigenvalue problem to solve for $\mathbf{w}_2$, we rearrange equation A.13 to obtain the expression for $\mathbf{w}_1 = \frac{1}{\lambda_1}(C_{11})^{-1}C_{12}\mathbf{w}_2$. We substitute in equation A.14 to obtain:

$$C_{21}(C_{11})^{-1}C_{12}\mathbf{w}_2 = \eta C_{22}\mathbf{w}_2. \tag{A.16}$$

solving this is the generalized eigenvalue problem will give the optimal projection direction $\mathbf{w}_2$. For both equations A.15 and A.16, $\eta = \lambda_1\lambda_2$.

A.6 MCCA; Deriving Solution

The MCCA objective function is as follows:

$$\begin{aligned}&\underset{\{\mathbf{w}_v\}_{i=1}^V}{\text{argmax}} \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk}\mathbf{w}_k \\ &\text{s.t} \quad \mathbf{w}_v^T C_{vv}\mathbf{w}_v = 1, \quad \text{for} \;\; v = \{1, 2, ..., V\}.\end{aligned}$$

The Lagrange multiplier of this problem is:

$$\mathcal{L}(\{\mathbf{w}_v\}_{i=1}^V, \{\lambda_v\}_{v=1}^V) = \sum_{v=1}^{V}\sum_{k=1}^{V} \mathbf{w}_v^T C_{vk}\mathbf{w}_k - \sum_{v=1}^{V} \lambda_v(\mathbf{w}_v^T C_{vv}\mathbf{w}_v - 1).$$

Then the derivative of the Lagrange multiplier function is taken with respect to each of the $\mathbf{w}_v \in \mathbb{R}^{p_v \times 1}$ vectors and equated to the zero vector $\mathbf{0} \in \mathbb{R}^{p_v \times 1}$. The derivative of the Lagrange multiplier function with respect to a specific $\mathbf{w}_v$ is:

$$\frac{\delta \mathcal{L}}{\delta \mathbf{w}_v} = 0; \quad 2\sum_{k=1}^{V} C_{vk}\mathbf{w}_k - 2\lambda_v C_{vv}\mathbf{w}_v = 0,$$

this gives V different equations:

$$\sum_{k=1}^{V} C_{vk}\mathbf{w}_k = \lambda_v C_{vv}\mathbf{w}_v.$$

We can collect the left and right side of the previous equation as vectors; show as:

$$\begin{bmatrix} \sum_{k=1}^{V} C_{1k}\mathbf{w}_k \\ \vdots \\ \sum_{k=1}^{V} C_{Vk}\mathbf{w}_k \end{bmatrix} = \begin{bmatrix} \lambda_1 C_{11}\mathbf{w}_1 \\ \vdots \\ \lambda_V C_{VV}\mathbf{w}_V \end{bmatrix},$$

This can be rewritten in terms of block matrices and by grouping all the w_i vectors into a single vector. This gives:

$$\begin{bmatrix} C_{11} & \dots & C_{1V} \\ \vdots & \ddots & \vdots \\ C_{V1} & & C_{VV} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_V \end{bmatrix} = \begin{bmatrix} C_{11} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & C_{VV} \end{bmatrix} \begin{bmatrix} \lambda_1 I_{p_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \lambda_V I_{p_v} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_V \end{bmatrix}. \quad \text{(A.17)}$$

A.17 is a multivariable generalized eigenvalue problem (MEP). Now the set of canonical transformations $\{\mathbf{w}_v\}_{v=1}^V$ can be found by solving this MEP.

A.7

Defining the conjugate of function $f(\boldsymbol{x})$:

$$f^*(\boldsymbol{u}) = \max_{\boldsymbol{x}} \boldsymbol{u}^T\boldsymbol{x} - f(\boldsymbol{x}). \tag{A.18}$$

Recall that the Lagrangian function of problem 3.24 is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) + \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b}), \tag{A.19}$$

now, its dual function is expressed as:

$$\begin{aligned} g(\boldsymbol{u}) &= \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = \min_{\boldsymbol{x}} f(\boldsymbol{x}) + \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b}) \\ &= -\boldsymbol{u}^T\boldsymbol{b} + \min_{\boldsymbol{x}} \big(f(\boldsymbol{x}) + \boldsymbol{u}^T A\boldsymbol{x}\big) \\ &= -\boldsymbol{u}^T\boldsymbol{b} - \max_{\boldsymbol{x}} \big((-A^T\boldsymbol{u})^T\boldsymbol{x} - f(\boldsymbol{x})\big) \\ &= \boldsymbol{u}^T\boldsymbol{b} - f^*(-A^T\boldsymbol{u}). \end{aligned} \tag{A.20}$$

A.8

Statement: If a function f is closed and convex, then $\boldsymbol{x} \in \delta f^*(\boldsymbol{u}) \iff \boldsymbol{u} \in \delta f(\boldsymbol{x}) \iff \boldsymbol{x} \in \operatorname{argmin}_{\boldsymbol{z}} f(\boldsymbol{z}) - \boldsymbol{u}^T\boldsymbol{z}$.

Proof
Recall the conjugate of function $f(\boldsymbol{x})$:

$$f^*(\boldsymbol{u}) = \max_{\boldsymbol{x}} \boldsymbol{u}^T\boldsymbol{x} - f(\boldsymbol{x}). \tag{A.21}$$

Proving: $\boldsymbol{u} \in \delta f(\boldsymbol{x}) \Rightarrow \boldsymbol{x} \in \delta f^*(\boldsymbol{u})$.
Assume that $\boldsymbol{u} \in \delta f(\boldsymbol{x})$ is known. Recall

$$-f^*(\boldsymbol{u}) = \min_{\boldsymbol{z}} f(\boldsymbol{z}) - \boldsymbol{u}^T\boldsymbol{z}, \tag{A.22}$$

then from first order optimality condition of A.22, optimal $\boldsymbol{z}$ satisfies the subgradient optimality condition:

$$\boldsymbol{0} \in \delta f(\boldsymbol{z}) - \boldsymbol{u} \implies \boldsymbol{u} \in \delta f(\boldsymbol{z}).$$

Therefore, following from the assumption that $\boldsymbol{u} \in \delta f(\boldsymbol{x})$ is true, we now know that $\boldsymbol{x}$ must minimize $f(\boldsymbol{z}) - \boldsymbol{u}^T\boldsymbol{z}$ or equivalent must maximize $\boldsymbol{u}^T\boldsymbol{z} - f(\boldsymbol{z})$.

The conjugate function with $\boldsymbol{z}$ as the variable is

$$f^*(\boldsymbol{u}) = \max_{\boldsymbol{z}} \underbrace{\boldsymbol{u}^T\boldsymbol{z} - f(\boldsymbol{z})}_{f_{\boldsymbol{z}}(\boldsymbol{u})}.$$

Then, the subgradient of $f^*(\boldsymbol{u})$ is the union of the subgraidents $f_{\boldsymbol{z}}(\boldsymbol{u})$ w.r.t $\boldsymbol{u}$ for all $\boldsymbol{z}$ that maximize $f_{\boldsymbol{z}}(\boldsymbol{u})$. It is written mathematically as the closure of the convex hull of such as set of subgradients:

$$\delta f^*(\boldsymbol{u}) = \text{cl}(\text{conv}(\cup_{\boldsymbol{z}\in M_{\boldsymbol{u}}}\{\delta f_{\boldsymbol{z}}(\boldsymbol{u})\})) = \text{cl}(\text{conv}(\cup_{\boldsymbol{z}\in M_{\boldsymbol{u}}}\{\boldsymbol{z}\})),$$

where $M_{\boldsymbol{u}}$ is the set of maximizers of $f_{\boldsymbol{z}}(\boldsymbol{u})$, and $\delta f_{\boldsymbol{z}}(\boldsymbol{u}) = \boldsymbol{z}$. Thus, this concludes that proof that $\boldsymbol{x} \in \delta f^*(\boldsymbol{u})$.

Proving: other direction $\boldsymbol{x} \in \delta f^*(\boldsymbol{u}) \Rightarrow \boldsymbol{u} \in \delta f(\boldsymbol{x})$.
Assuming that $\boldsymbol{x} \in \delta f^*(\boldsymbol{u})$ is known. Stemming from previous proof we an state that $\boldsymbol{u} \in \delta f^{**}(\boldsymbol{u})$, which equals to $\boldsymbol{u} \in \delta f(\boldsymbol{u})$, because a property of conjugates is that $f^{**} = f$.

Proving: $\boldsymbol{u} \in \delta f(\boldsymbol{x}) \iff \boldsymbol{x} \in \text{argmin}_{\boldsymbol{z}}\, f(\boldsymbol{z}) - \boldsymbol{u}^T\boldsymbol{z}$.
Referring to first proof, this stems from the first order optimality condition of A.22.