CONVERTING DATA FROM MULTI-INSTANCE TO SINGLE-INSTANCE REPRESENTATIONS USING P-ORDER LAPLACIAN PROJECTIONS

by

Saad Elbeleidy

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

Golden, Colorado

Date _____

Signed: _____
Saad Elbeleidy

Signed: _____
Dr. Hua Wang
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Tracy Camp
Professor and Head
Department of Computer Science

ABSTRACT

Fields such as Computer Vision and Natural Language Processing have a high applicability of Machine Learning algorithms. With large amounts of complex data readily available, there are two prominent approaches to handling data complexity using Machine Learning.

First, dimensionality reduction methods such as Principal Component Analysis (PCA) or Laplacian Embeddings (LE) can minimize the number of features needed to accurately represent data. This approach is often effective but has two main drawbacks. First, the input to the dimensionality reduction method is a summary of all the components that make up the data and some valuable information may be lost. Second, dimensionality reduction methods are often sensitive to outliers.

The second approach to dealing with complex data is Multi-Instance Learning (MIL). MIL introduces a new paradigm for data representation by viewing data as a grouping, called a bag, of instances. Each instance is modeled the same way the whole data would be represented but now the datum is represented as a bag of instances. Multi-Instance representation can be effective since they focus on modeling all the pieces that make up the whole datum. However, in order to use this representation in Machine Learning applications we must use MIL algorithms and cannot directly use traditional Machine Learning algorithms.

In this work, we propose a method to tackle the issues that may arise in dimensionality reduction methods and MIL methods. We do this by learning a reduced-dimension, integrated, outlier resilient single instance representation for our data. We first propose a new dimensionality reduction method of p-Order Laplacian Embeddings (pOLE) that is less sensitive to outliers than traditional LE. We then use this method to learn a projection from the instances of each bag in a Multi-Instance representation of data. This projection, combined with the Single-Instance representation of the same data can produce a reduced-dimension, integrated, outlier resilient Single-Instance representation

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF SYMBOLS

Data . . . . . **X**

Projection . . . . . **W**

Similarity matrix . . . . . **S**

Degree matrix . . . . . **D**

Laplacian matrix . . . . . **L**

LIST OF ABBREVIATIONS

Bag of Words . . . . . BoW

Term Frequency . . . . . TF

Latent Drichlect Allocation . . . . . LDA

Multi-Instance . . . . . MI

Multi-Instance Learning . . . . . MIL

Single-Instance . . . . . SI

p-Order Laplacian Embedding . . . . . pOLE

ACKNOWLEDGMENTS

I would like to thank my family and colleagues. This project would never have succeeded without their support.

CHAPTER 1

INTRODUCTION

Machine Learning (ML) applications have gained traction recently due to the value they can create to society because of their effectiveness. There are many areas of applications of Machine Learning that are changing how we live our everyday lives. Examples include applications such as speech recognition, medical diagnosis, translation, and others.

One of the reasons that machine learning models have recently become more effective is the large increase in available data. With the advances in Big Data, ML models can extract substantially more information from data as well as have more data to learn from. This is clear in certain types of data such as images when applied to Computer Vision and text documents when applied to Natural Language Processing. Due to the massive increase in images shared between people and ease of taking high resolution photos, computer vision data sets have drastically increased in both number and size. In addition to that, with news and general communication being shared over the internet, there is a large amount of text data available to researchers. With this abundance of data, the methods we use to represent it are critical to the success of our Machine Learning models.

In Computer Vision, an image can be represented as a tensor in 3 dimensions; height, width and RGB values. This approach converts an image into a numerical representation that can be used by Machine Learning models. As the resolution of images increases, the number of values stored in these tensors increases exponentially. For example, image dimensions that were used for the first LCD screens were 800 by 600 pixels, resulting in 1,440,000 integer values. HD images at 1080 by 720 pixels result in 2,332,800 integer values. Finally, 4K images at 4096 by 3072 pixels result in 37,748,736 integer values.

In Natural Language Processing, a text document can be represented in many ways. A simple representation is a vector of the number of occurrences a word appears in the text.

This specific representation is called the Bag of Words (BoW), sometimes also known as Term Frequency (TF). This method is a simple approach and can sometimes seem rudimentary due to the high sparsity that results from it. This sparsity is caused by some words that may only appear in specific documents that do not appear in the remainder of the dataset or appear infrequently. A variation of BoW is a binarization of its resulting matrix using a minimum threshold value. This means that if a word appears more than a certain number of times then the stored value is 1, otherwise it is 0. If a word does not appear beyond that minimum threshold in any documents we can then remove its column in the matrix. There are several other methods of modeling text that are more effective. Two such methods that are relevant to this work are GloVe [1] and Latent Dirichlect Allocation (LDA)[2]. Both LDA and GloVe create a representation for words based on how they are used in conjunction with other words. These methods do not extract an inherent linguistic representation and instead use a statistical approach to modeling them.

GloVe is a global method that can be trained on a large corpus of text to provide 300 numerical values for every word regardless of context. This approach models the *meaning* of each word to represents it numerically. In order to represent a sentence or document, an average of all the words in the sentence is often used as a summary of the meaning of all the words in the text.

LDA, on the other hand, is a local method as it models the topics in the target data set. It learns the context of the data set and how words are used in conjunction with others. From keeping track of how the words are used relative to each other, it then creates a pre-specified number of topics (input parameter) and can cluster documents into one or more topics. LDA is considered a topic modelling method and models the similarity between various documents of text. Similar to other clustering methods that take the number of clusters as an input, the performance of LDA can be vary significantly based on the input parameter.

Based on how large the datasets can be, the sparsity and complexity of the data, many approaches have been considered to more concisely represent the information stored within

these items. Two of these approaches are Dimensionality Reduction and Multi-Instance Learning.

The above introduces various methods of data representation for images and text. As our data increases in size of representation and number of data items to track, there is a need for methods that can handle the complexity of the data. We will discuss two such approaches to managing data complexity, dimensionality reduction and Multi-Instance Learning.

Dimensionality Reduction summarizes data by reducing the number of features to a more manageable value. This method is effective when there is redundancy in data. For example, in images with a high resolution, we may find that many pixels near each other will contain the exact same value. In text, we might find that synonyms are used frequently. With dimensionality reduction, this redundancy can be minimized and the number of features to track is reduced. We will discuss dimensionality reduction in further detail in Section 2.

Multi-Instance Learning (MIL) is an approach to perform supervised learning on data that is represented as a bag of instances instead of a single instance. This new representation of data as a bag of instances is called a Multi-Instance (MI) representation. The MI representation can be more effective in that it models all the pieces involved in the data separately instead of creating a summary for the data. For example, in applications of object detection in images a small patch of the image can be sufficient in identifying an object. In entity or topic detection in text, a single sentence can be sufficient to cause a positive labeling. MIL is very successful when a small portion of the data (eg. patch or sentence) is responsible for the label attributed to the data. One major drawback of using MI representations, however, is that we must then use MIL algorithms to perform any machine learning applications. We will discuss MIL in further detail in Section 3.

Both dimensionality reduction and MIL can be effective in solving machine learning problems, however they can suffer from several drawbacks. Dimensionality reduction can be sensitive to outliers in the data and it also uses a representation that is often itself a summary of the data. For example, as we mentioned the GloVe representation of a document

is the average of all the words' representations. This can lead to potentially differentiating information being lost in the summary. With MIL, we are forced to use MIL algorithms which lack the flexibility and diversity of traditional machine learning algorithms.

To address these issues, our work proposes a method that creates an integrated Single Instance (SI) representation that is resilient to outliers in a reduced dimension space. First, we introduce an outlier-resilient dimensionality reduction method using p-Order Laplacian Embeddings (pOLE) that will be covered in Section 4.1. Next, we introduce a novel approach to convert a MI representation to a SI representation. Our approach learns a projection from instances in a MI representation used to transform the original SI summary representation into the new space to form an integrated representation. This output representation is considered an integrated representation since it uses information from both the instances (the learned projection) as well as the whole (the original SI representation). Details of this approach will be covered in Section 4.2.

CHAPTER 2

DIMENSIONALITY REDUCTION METHODS

When we have a large number of features or dimensions in data, computation time for machine learning models can increase drastically relative to having a small number of features. There is a need to reduce the dimensionality of data to more efficiently perform computations. There are two main approach to dimensionality reduction; based on linear space and based on the manifold of the data. An example of dimensionality reduction in linear space is Principal Component Analysis which is discussed in further detail in Section 2.1. Laplacian Embeddings (LE) are an example of manifold-based dimensionality reduction and will be discussed in further detail in Section 2.2.

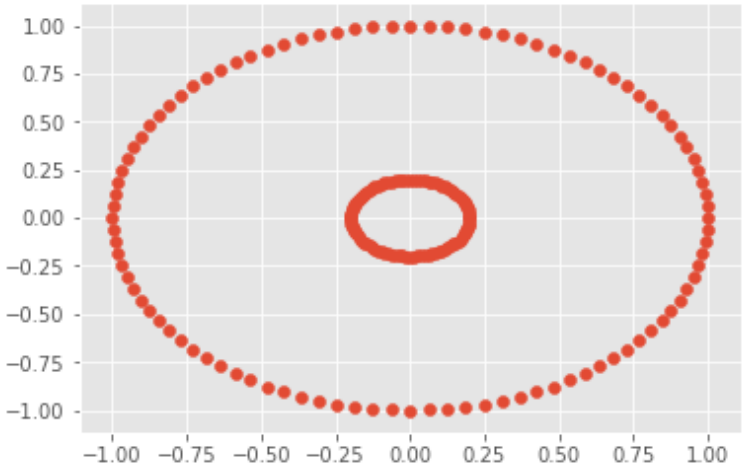
Both these approaches can perform effectively in reducing dimensionality of the data depending on the nature of the data. When there exists a relative relationship between data points such as in a data set of concentric circles, manifold-based methods can outperform linear methods. Otherwise, linear methods will often outperform manifold methods. Examples of manifold structures are shown in Figure 2.1.

2.1 Principal Component Analysis

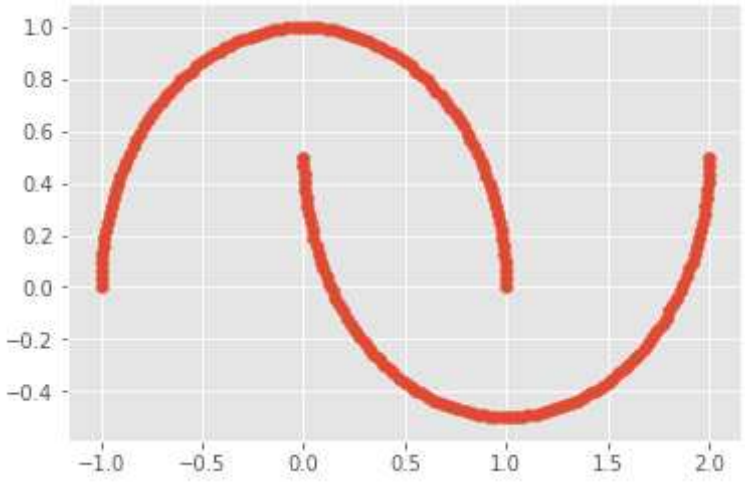
Principal Component Analysis [3] is an effective linear-based dimensionality reduction method. PCA identifies dimensions based on the data that are orthogonal to each other. These dimensions are called the principal components of the data and because they are orthogonal, they can minimize the redundancy in the data.

PCA is calculated by maximizing the following objective:

$$\begin{aligned} &\max_{W} \mathbf{tr}(\mathbf{WSW}^T), \\ &\quad s.t. \mathbf{WW}^T = \mathbf{I}, \end{aligned} \tag{2.1}$$



(a) Concentric Circles



(b) Two Moons

Figure 2.1: Examples of data possessing a manifold structure. Concentric circles (a) and two moons (b) would be easily identified as different groups using a manifold-based dimensionality reduction method's representation as compared to a linear-based method. This is because manifold-based methods model the relationships between points as opposed to their position in linear space.

where $\mathbf{W}$ is the resultant projection and $\mathbf{S}$ is a diagonal of all the eigenvalues of the initial data. Here, the trace of a matrix $\mathbf{X}$, $\mathbf{tr}(\mathbf{X})$ is the sum of its diagonal, $\sum_i^n X_{ii}$. From this maximization, we obtain the projection $\mathbf{W}$ that can be combined with the original data to produce the new representation of the data in the transformed space. This result will be in the same number of dimensions as the original and explain the variability of the data as a whole. We will find that the first few components, based on the ordering in the eigenvalue decomposition, are often sufficient in explaining a large percentage of the variance.

A common approach in applying PCA is to use a predetermined value for the new dimension space, n and use the first n dimensions from the resulting transformation. Another method is to set a desired explained variance value and selecting the minimum n components that achieve this explained variance. We can calculate the explained variance in the data with k components using the covariance matrix, $\mathbf{C}$ of the transformed data. The explained variance of each dimension, i, is calculated as,

$$\mathbf{EV_i} = \frac{\mathbf{C}_{ii}}{\mathbf{tr}(\mathbf{C})}. \tag{2.2}$$

Selecting a desired explained variance is often more effective, however the arbitrary selection of the number of components is often used due to computational or other limitations. To better understand the effectiveness of using PCA, an example of the results produced when using PCA in a simple case is shown in Figure 2.2.

2.2 Laplacian Embeddings

While PCA can be extremely effective when data exhibits linear trends, it does not always effectively model data that exhibits relational structures such as a manifold. Laplacian Embeddings [4] use the graph structure of the data to determine a lower dimensionality representation of the data.

To determing the Laplacian embeddings of data, construct a graph that models the similarity between the data points. Regardless of the method used to determine the graph structure, represent the graph using an adjacency matrix, $\mathbf{S}$ where s_{ij} represents the simi-

Figure 2.2: An example showing the original data on the left and how it can be transformed using PCA to require less dimensions to explain the variance of our data. We can see from this result that the second principal component, $pc2$, is unnecessary in describing the variance in the data. This figure is provided by http://setosa.io/ev/principal-component-analysis/.

larity between node i and node j. Afterwards, we can calculate the diagonal degree matrix, $\mathbf{D}$ using

$$\mathbf{D}_{ii} = \sum_{j}^{n} s_{i_j}. \tag{2.3}$$

After calculating both the adjacency and degree matrixes, we can calculate the graph Laplacian, $\mathbf{L}$ where,

$$\mathbf{L} = \mathbf{D} - \mathbf{S}. \tag{2.4}$$

A graph's Laplacian is a useful representation as it simplifies the calculation of various attributes and structures in a graph, including Laplacian Embeddings. To calculate the Laplacian embeddings for a data set, $\mathbf{X}$, we want to minimize the following objective,

$$\begin{aligned} &\min_{W} \sum_{i}^{n} \sum_{j}^{n} s_{ij} ||\mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j||^2, \\ &s.t. \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \tag{2.5}$$

where s_{ij} is the similarity between node i and node j. We refer to $\mathbf{W}$ here as the Laplacian Projection and $\mathbf{W}^T\mathbf{x}$ is the resulting Laplacian Embedding. What this objective denotes is that we are minimizing the squared distance between the resulting embeddings weighted by the similarity between the original nodes. If node i and node j are similar in the original dataset, then their embeddings should also be similar whereas if they were distant then their embeddings will be distant. This in effect learns a new dimensionality reduced representation of the data based on the similarities defined by the graph constructed from the the data.

CHAPTER 3

MULTI-INSTANCE LEARNING

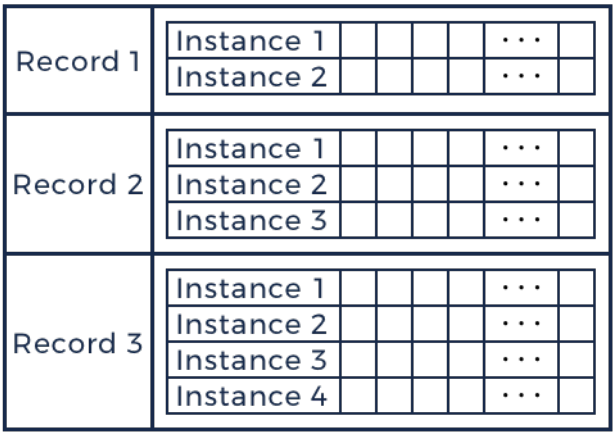
Multi-Instance Learning [5] introduces a new paradigm and approach to complex data by using a representation where each data item consists of multiple instances. Each instance in the data item is referred to as an instance. The collection of instances is referred to as a bag. This approach can appear naturally in our data or we can even structure our data using a MI representation in order to gain the benefits of MIL.

When we break up our data into smaller pieces, we can then gain information from each component of the data without its value being diminished when we look at the summarized SI representation of the whole. Some MIL algorithms can also identify the instance in the bag that triggers the labeling. This is called instance-label relation discovery. A good example of this is the Deep MIML Network [6].

To better understand how to structure MI representations, consider a dataset, $\mathbf{X}$, of n items where each item is in d dimensions. This results in our matrix $\mathbf{X}$ being in $n * d$ dimensions. So far, we are using a SI representation of our data. To convert it to an MI representation we break up each node, i, of the dataset to produce n_i instances each with a SI representation in d dimensions. Each node, i, is now represented as a matrix in $n_i * d$ dimensions instead of just d dimensions, where n_i is determined by the natural decomposition structure existing in the data. The resultant representation for the data $\mathbf{X}$ is n records each in $n_i * d$ dimensions. This data structure can be more easily visualized in Figure 3.1 which shows a comparison between SI and MI representations.

Due to the fact that the number of instances in each bag, n_i, differs between bags, we can not easily use traditional supervised learning methods. To tackle this issue, many MIL algorithms have been proposed and we will discuss two of them in Section 3.2

(a) Single Instance



(b) Multi-Instance

Figure 3.1: Visualization of MI vs SI representations

3.1 Multi-Instance Examples

The description of MI representations provides a good theoretical starting point, however this may be difficult to understand without examples in real world applications. In this section, we will cover examples in Computer Vision and Natural Language Processing which are the areas where MIL is most commonly applied.

For Computer Vision, an image can be broken up into smaller meaningful patches. In object detection and classification applications, each patch can represent an object that we wish to identify. With a labeled dataset of patches, the model can develop a better understanding of what image structure causes a specific label to occur. An example of this breakdown of images into MI representations can be seen in Figure 3.2. Each of the image patches must still be represented using the same dimensions.

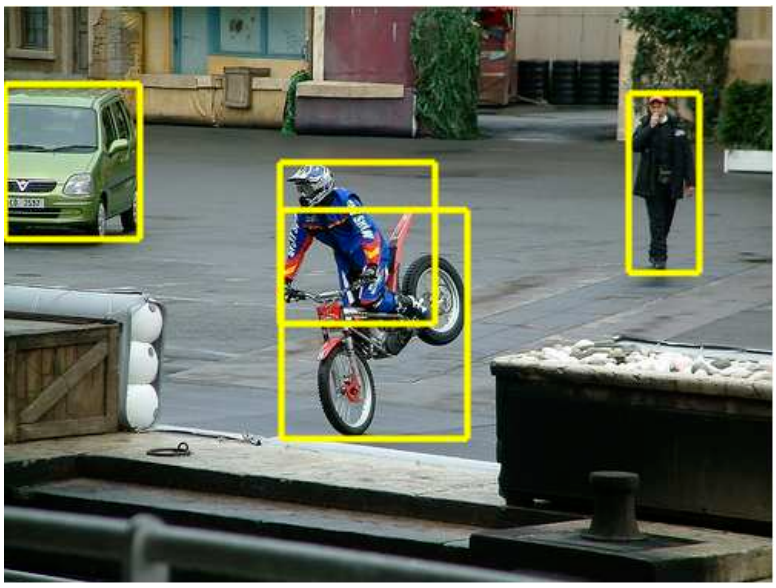


Figure 3.2: An example of an image with selected patches that make up its instances.

For Natural Language Processing, a document can be broken up into paragraphs or sentences. Each paragraph might focus on a specific topic which can then be easily identified when the model can target the passages individually instead of a summary of the document as a whole. In topic classification, this can become very effective in identifying a certain

topic due to high salience of a topic in a particular instance. Again in this representation, each passage that forms an instance must be represented in the same dimensions as each other. In NLP, this is easily done through most methods of feature extraction such as BoW or GloVe.

In both Computer Vision and Natural Language Processing, methods for determining which patches or passages become instances in the MI representation can vary. In some cases, they are provided as part of a labeled dataset, while in others they must be extracted using some inherent metric in the data. We can also use all portions of the data as instances, for example in NLP, we can use all paragraphs as instances where each paragraph is an instance of the bag / document it is a part of.

3.2 Multi-Instance Algorithms

As mentioned earlier, when we use a MI representation, we are unable to use traditional machine learning methods for modeling. In order to utilize the value available in the instances' data from the bags, many methods have been developed in order to perform effectively using MI representations. Some of these methods use traditional methods as a reference point and build on top of their approach in the new MI paradigm. Examples of these that we will briefly introduce to provide more background on the topic are MISVM [7] and MIKNN [8, 9] which are MIL algorithms analogous to Support Vector Machines (SVM) and k-Nearest Neighbors (kNN) methods in the traditional SI sense.

In general, with machine learning algorithms, having a distance metric is crucial to formulating an algorithm We need to make sure we can minimize or maximize a specific distance. With MIL algorithms, the key idea is to identify a distance between the bags of data instead of between the instances.

3.2.1 MISVM

MISVM[7] was introduced with two formulations. The first formulation approaches the problem as a way to learn to classify between the instances themselves, ignoring the bags.

This approach is referred to as the pattern margin formulation or soft-margin formulation. The second approach is to determine the margin between the bags.

3.2.2 MIKNN

There are multiple approaches in applying the k-Nearest Neighbors algorithm to MIL. The two popular approaches are MI-citation k-NN [9] and MIML-kNN[8]. Both approaches use a variation of the average Hausdorff Distance which is a measure of the distance between the bags. The average Hausdorff Distance is calculated as,

$$\mathcal{H}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{\mathbf{x}\in\mathbf{X}} \min_{\mathbf{y}\in\mathbf{Y}} ||\mathbf{x}, \mathbf{y}|| + \sum_{\mathbf{y}\in\mathbf{Y}} \min_{\mathbf{x}\in\mathbf{X}} ||\mathbf{x}, \mathbf{y}||}{|\mathbf{X}| + |\mathbf{Y}|} \tag{3.1}$$

where $|\ \cdot\ |$ denotes the set cardinality of the bag and $||\mathbf{a}, \mathbf{b}||$ denotes the Euclidean distance between a and b.

CHAPTER 4

PROPOSED METHOD

As mentioned in Section 2, many dimensionality reduction methods can be effective in reducing the complexity of data representations. However, some methods will give a disproportionate influence to outliers in the dataset. We propose a variation of Laplacian Embeddings that allows us to control the influence of outliers in Section 4.1. In Section 3, we covered the value of using Multi-Instance representations of data and applying MIL algorithms. One major drawback of MI representations is the restriction to use MIL algorithms. To mitigate this issue, in Section 4.2, we propose a method using learned projections from the instances to create a new SI representation of the data. We can also combine the two proposed methods to result in a more effective MI to SI representation transformation.

4.1 p-Order Laplacian Embeddings

In traditional Laplacian Embeddings, the objective used is a squared distance, as shown in Equation 2.5. This can be an issue in that larger distances are disproportionately weighted in the learned model. We propose a new objective function to create p-Order Laplacian Embeddings in r dimensions,

$$\begin{aligned} &\min_{W} \sum_{i}^{n} \sum_{j}^{n} s_{ij} ||\mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j||^p, \\ &s.t. \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \tag{4.1}$$

where $0 < p \leq 2$. With a power less than a square value, we can lower the effect of outliers. Traditionally, this change in power adds significant difficulty in generating a simple and elegant solution. We present an iterative solution algorithm to the objective in Equation 4.1 as Algorithm 1 and in Section 4.1.1, we rigorously prove its convergence.

Algorithm 1: The algorithm to solve the p-Order Laplacian Embedding Objective.

Input: Training data $\mathbf{X} \in \Re^{d \times n}$. The original weight matrix $\mathbf{S} \in \Re^{n \times n}$. $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element as $\sum_j s_{ij}$.

Initialize $\mathbf{W} \in \Re^{d \times r}$ such that $\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W} = \mathbf{I}$;

while *not converge* **do**

1. Calculate $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{S}}$, where $\tilde{s}_{ij} = \frac{p}{2} s_{ij} \left\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\right\|_2^{p-2}$, $\tilde{\mathbf{D}}$ is a diagonal matrix with the i-th diagonal element as $\sum_j \tilde{s}_{ij}$;
2. Update $\mathbf{W}$. The columns of the updated $\mathbf{W}$ are the first r eigenvectors of $\left(\mathbf{X}\mathbf{D}\mathbf{X}^T\right)^{-1}\mathbf{X}\tilde{\mathbf{L}}\mathbf{X}^T$ corresponding to the first r smallest eigenvalues;

Output: $\mathbf{W} \in \Re^{d \times r}$.

By using the parameter p in our Laplacian Embeddings, we gain some control over the sensitivity to outliers that we want in the model. This parameter may be data-dependent as some datasets may have more or less outliers and by changing this paraemeter we can retrieve improved representative embeddings. We conducted an empirical analysis to determine how the change in the value for p affects the resulting model and this can be seen in the Experiments in Section 5.

To derive the solution algorithm, we start by investigating the Lagrangian function of the optimization problem,

$$\begin{aligned}\mathcal{L}(\mathbf{W}) = &\sum_{i,j=1}^{n} s_{ij} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^p \\ &- \mathbf{tr}\left(\mathbf{\Lambda}(\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{X} - \mathbf{I})\right) \ .\end{aligned} \tag{4.2}$$

Here we define a Laplacian matrix $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{S}}$, where $\tilde{\mathbf{S}}$ is a re-weighted weight matrix defined by

$$\tilde{s}_{ij} = \frac{p}{2} s_{ij} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2} \tag{4.3}$$

and $\tilde{\mathbf{D}}$ is a diagonal matrix with the i-th diagonal element as $\sum_j \tilde{s}_{ij}$. Taking the derivative of $\mathcal{L}(\mathbf{W})$ with respect to $\mathbf{W}$, and setting it to zero, we have:

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{X}\tilde{\mathbf{L}}\mathbf{X}^T\mathbf{W} - \mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W}\mathbf{\Lambda} = \mathbf{0} \ , \tag{4.4}$$

which indicates that the solution $\mathbf{W}$ is the eigenvectors of $\left(\mathbf{X}\mathbf{D}\mathbf{X}^T\right)^{-1}\mathbf{X}\tilde{\mathbf{L}}\mathbf{X}^T$. Note that $\left(\mathbf{X}\mathbf{D}\mathbf{X}^T\right)^{-1}\mathbf{X}\tilde{\mathbf{L}}\mathbf{X}^T$ is dependent on $\mathbf{W}$. Thus we can derive the iterative algorithm to obtain the solution $\mathbf{W}$ such that the KKT conditions are satisfied.

In every iteration of the algorithm, $\tilde{\mathbf{L}}$ is calculated with the current solution of $\mathbf{W}$, then $\mathbf{W}$ is updated according to the currently calculated $\tilde{\mathbf{L}}$. This iteration procedure repeats until it converges. From the algorithm we can see that the original weight matrix $\mathbf{S}$ is adaptively re-weighted to minimize the objective during iterations.

4.1.1 Proof of Algorithmic Convergence

In the following, we prove the convergence of the algorithm. First, we prove the following lemmas.

Lemma 1. *For any scalar x, when $0 < p \leq 2$, we have $2|x|^p - px^2 + p - 2 \leq 0$.*

Proof: Let $f(x) = 2x^{\frac{p}{2}} - px + p - 2$, then we have

$$f'(x) = p(x^{\frac{p-2}{2}} - 1), \tag{4.5}$$

and

$$f''(x) = \frac{p(p-2)}{2}x^{\frac{p-4}{2}}. \tag{4.6}$$

Obviously, when $x > 0$ and $0 < p \leq 2$, then $f''(x) \leq 0$ and $x = 1$ is the only point that $f'(x) = 0$. Note that $f(1) = 0$, thus when $x > 0$ and $0 < p \leq 2$, then $f(x) \leq 0$. Thus $f(x^2) \leq 0$, which indicates $2|x|^p - px^2 + p - 2 \leq 0$. □

Lemma 2. *For any nonzero vectors $\mathbf{v}$ and $\mathbf{v}_0$, when $0 < p \leq 2$, the following inequality holds:*

$$\begin{aligned} &\|\mathbf{v}\|_2^p - \frac{p}{2}\|\mathbf{v}_0\|_2^{p-2}\|\mathbf{v}\|_2^2 \\ \leq &\|\mathbf{v}_0\|_2^p - \frac{p}{2}\|\mathbf{v}_0\|_2^{p-2}\|\mathbf{v}_0\|_2^2. \end{aligned} \tag{4.7}$$

Proof:

$$
\begin{aligned}
&2\Big(\tfrac{\|\mathbf{v}\|_2}{\|\mathbf{v}_0\|_2}\Big)^p - p\Big(\tfrac{\|\mathbf{v}\|_2}{\|\mathbf{v}_0\|_2}\Big)^2 + p - 2 \le 0\\
&\Rightarrow\\
&2\left\|\mathbf{v}\right\|_2^p - p\left\|\mathbf{v}_0\right\|_2^{p-2}\left\|\mathbf{v}\right\|_2^2 \le (2-p)\left\|\mathbf{v}_0\right\|_2^p\\
&\Rightarrow\\
&\left\|\mathbf{v}\right\|_2^p - \tfrac{p}{2}\left\|\mathbf{v}_0\right\|_2^{p-2}\left\|\mathbf{v}\right\|_2^2 \le \left\|\mathbf{v}_0\right\|_2^p - \tfrac{p}{2}\left\|\mathbf{v}_0\right\|_2^{p-2}\left\|\mathbf{v}_0\right\|_2^2,
\end{aligned}
$$

where the first inequality is true according to Lemma 1. □

Now we have the following theorem:

Theorem 1. *The algorithm will monotonically decrease the objective in each iteration, and converge to a local optimum of the problem.*

Proof: Suppose the updated $\mathbf{W}$ is $\tilde{\mathbf{W}}$. According to step 2 in the algorithm, we know that

$$
\begin{aligned}
&\tilde{\mathbf{W}} = \arg\min_{\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W}=\mathbf{I}} \mathbf{tr}\left(\mathbf{W}^T\mathbf{X}\tilde{\mathbf{L}}\mathbf{X}^T\mathbf{W}\right)\\
&= \arg\min_{\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} \tilde{s}_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^2.
\end{aligned}
\tag{4.8}
$$

Note that $\tilde{s}_{ij} = \frac{p}{2}s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2}$, we have

$$
\begin{aligned}
&\sum_{i,j=1}^{n} \tfrac{p}{2}s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2}\left\|\tilde{\mathbf{W}}^T\mathbf{x}_i - \tilde{\mathbf{W}}^T\mathbf{x}_j\right\|_2^2\\
&\le \sum_{i,j=1}^{n} \tfrac{p}{2}s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^2.
\end{aligned}
\tag{4.9}
$$

According to Lemma 2, we have

$$
\begin{aligned}
&\sum_{i,j=1}^{n} s_{ij}\left\|\tilde{\mathbf{W}}^T\mathbf{x}_i - \tilde{\mathbf{W}}^T\mathbf{x}_j\right\|_2^p -\\
&\sum_{i,j=1}^{n} \tfrac{p}{2}s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2}\left\|\tilde{\mathbf{W}}^T\mathbf{x}_i - \tilde{\mathbf{W}}^T\mathbf{x}_j\right\|_2^2\\
&\le \sum_{i,j=1}^{n} s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^p -\\
&\sum_{i,j=1}^{n} \tfrac{p}{2}s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^{p-2}\left\|\tilde{\mathbf{W}}^T\mathbf{x}_i - \tilde{\mathbf{W}}^T\mathbf{x}_j\right\|_2^2.
\end{aligned}
\tag{4.10}
$$

Summing Eq. (4.9) and Eq. (4.10) on both sides of the inequality, we have

$$
\begin{aligned}
&\sum_{i,j=1}^{n} s_{ij}\left\|\tilde{\mathbf{W}}^T\mathbf{x}_i - \tilde{\mathbf{W}}^T\mathbf{x}_j\right\|_2^p\\
&\le \sum_{i,j=1}^{n} s_{ij}\left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\right\|_2^p.
\end{aligned}
\tag{4.11}
$$

Thus the algorithm monotonically decreases the objective in each iteration until the algorithm converges. In the convergence, the equality in Eq. (4.11) holds, thus $\mathbf{W}$ and $\tilde{\mathbf{L}}$ will satisfy Eq. (4.4), the KKT condition of the problem. Therefore, the algorithm will converge to a local optimum. □

As this method follows from Newton's method, the convergence rate is approximately quadratic in terms of n. In terms of time complexity for the algorithm, our update of $\mathbf{S}$ is in $O(rd)$ and the eigenvalues calculation is in $O(rn^2)$ which is a reasonable time complexity.

4.2 Multi-Instance to Single-Instance Transformations

Due to the limitation of MI representations requiring MIL algorithms for applications in machine learning, we propose a method for converting MI to SI representations that creates an integrated representation using information from both the instances and the holistic representation of the data. Our approach uses a projection extracted from the instances of a data item then transforms the holistic representation of the data using this projection to determine a final integrated representation. This is termed an **integrated** representation because it utilizes information from both the holistic representation of the data as well as the instances of the data.

We begin with a representation of our data, $\mathcal{X}$ such that each data item, a, possesses two representations $\{\mathbf{x}_a, \mathbf{X}_a\}$. Here, $\mathbf{x}_a$ is a holistic SI representation of the data item in d dimensions and $\mathbf{X}_a$ is a MI representation of a bag of n_a instances where each instance is in d dimensions, resulting in a matrix of $n_a * d$ dimensions.

Next, we learn a projection $\mathbf{W}_a$ from $\mathbf{X}_a$ in $d*r$ dimensions, where r is an input parameter determining the dimensions of the final integrated representation $\mathbf{y}_a$. Finally, we calculate the integrated representation for data item a using,

$$\mathbf{y}_a = \mathbf{W}_a^T \mathbf{x}_a. \tag{4.12}$$

A visualization of this approach when used in conjunction with pOLE is shown in Figure 4.1.

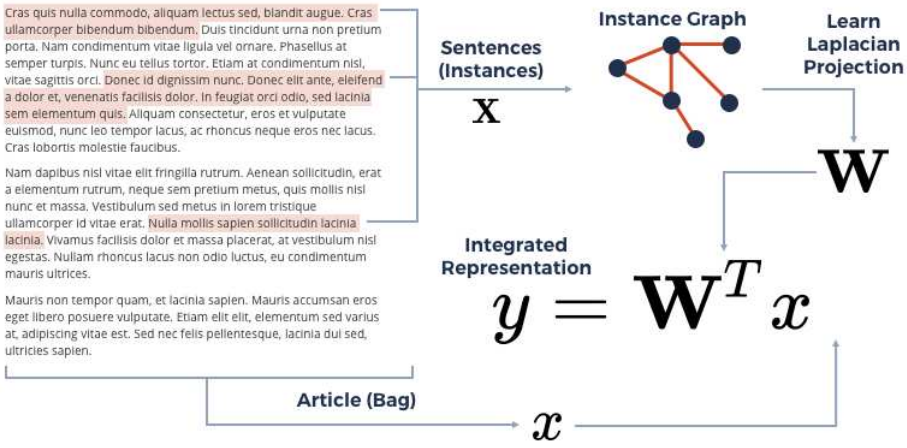


Figure 4.1: The process of our proposed method to convert a representation from Muiltiple Instances to Single Instance.

CHAPTER 5

EXPERIMENTS & RESULTS

To empirically support our proposed method we conduct experiments on natural language datasets. Preliminary experiments showed that data quality and complexity are crucial to our proposed methods. If data labeling is poor then a machine learning model will have trouble distinguishing between different labels. We also found that short documents did not possess sufficient complexity to achieve good results using a Multi-Instance representation. This is due to the data not having a sufficient number of instances to provide distinguishing features. Choosing sentences as instances also resulted in all sentences seemingly converging to the same value in a smaller document.

Based on our initial experimentation and drawbacks of other datasets, we found the BBC News Article Dataset to be a good candidate for our experiments. The dataset contained a sufficient number of articles and each article was lengthy enough for our MI representation using patagraphs. The articles are labeled by topic and so we focused on using our methods for document representation and applied these representations to a topic classification task.

First we begin by selecting articles in the dataset with at least 5 paragraphs. Then we randomly select 50 articles from each topic to obtain a stratified subset, where each label has the same number of data points. We then break up each article into two structures, one that contains all the text for the article to be used for the holistic representation and an array of paragraphs to be used for the MI representation. We assume that paragraphs are designated by two new line characters and do not apply any semantic method of paragraph splitting. This breaks up paragraphs as the author intended them and not based on some inherent meaning in the text.

From this new structure, we then apply two feature extraction methods to each string; LDA and GloVe. For LDA, we apply the feature extraction on all the holistic representations

once and on all the instances once, with the resultant number of features set to 300. For GloVe, we take the cosine average of the GloVe vectors based on a model trained on the Wikipedia dataset. From these representations, we then calculate an integrated representation for each article for each of LDA, $\mathbf{Y}_{LDA}$ and GloVe, $\mathbf{Y}_{GloVe}$.

Using these new representations, we conduct several experiments. First, we ensure that our theoretically proven converging algorithm converges and monitor the speed at which it does so. Second, we conduct a topic classification experiment where we use different projections and our MI to SI transformation method. This experiment allows us to see the effectiveness of our method in calculating an integrated representation as well as how the p-Order Laplacian Embedding-based projection performs when compared to other projections. When using the pOLE projection, we also investigate various values of p and take a look at any improvement when compared to LE as well as other values of p. In order to conduct the supervised learning experiment we apply two baseline SI machine learning algorithms of Support Vector Machines (SVM) and k-Nearest Neighbors classifiers. This shows us that we can indeed use a SI algorithm with our proposed approach.

5.1 pOLE Solution Algorithm Convergence

First we begin with an empirical study of the solution algorithm's convergence. A graph of the objective function when learning an integrated representation over several iterations for an example data item is shown in Figure 5.1. From this result we can conclude that the algorithm in fact converges and it does so at a good rate in under 30 iterations.

5.2 p Parameter Search

Next, we investigate the optimal values for p by running the supervised learning experiments over values of p from 0.1 to 1.8 with 0.1 intervals between 0.1 and 1 and 0.2 intervals between 1 and 1.8. For our supervised learning experiments, we tracked the accuracy of the labeled topic for each article. We found that in general, values between 0.5 and 1.5 for p provided the best results. This shows that we should lower the direct influence of outliers

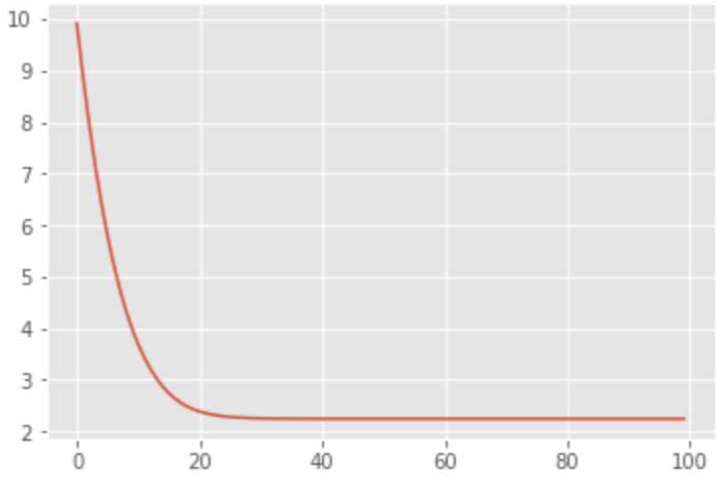


Figure 5.1: Objective function convergence for an example data item over several iterations.

in the model but that outliers can still add some value in understanding the underlying structure of the data since there may also be outliers in the test set.

Detailed results on the accuracy scores of our models across various p values are shown in the graphs in Figure 5.2.

5.3 MI to SI Transformation's Projection Comparison and Evaluation

Finally, we can compare the best results for each of PCA, LE, and pOLE as well as evaluate these results as a starting frame of reference for whether our approach of transforming data from MI to SI representations is a viable one. We can see the results for the various projections in Table 5.1. As a reference point, an accuracy of 20% would be considered random.

From the results, we can see that our model is superior to random guessing which means that our integrated representation is providing some value for our models. We can also see that the pOLE model can achieve superior results to other projections and in particular, LE.

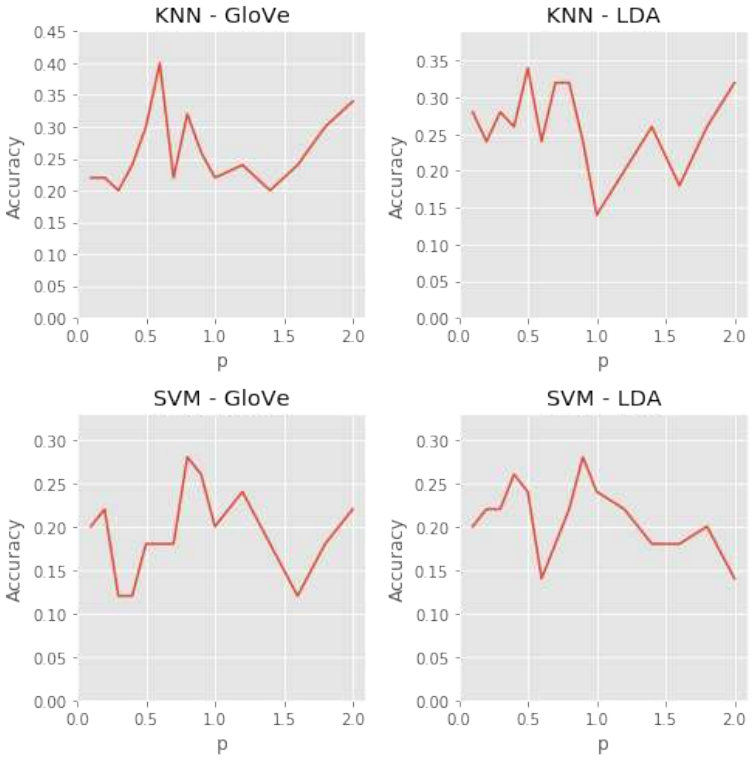


Figure 5.2: Accuracy scores across various values of p.

Table 5.1: Classification Accuracy

		Accuracy	
Representation	Features	SVM	k-NN
PCA	GloVe	12%	30%
LE	GloVe	12%	34%
Ours	GloVe	**28%**	**40%**
PCA	LDA	12%	10%
LE	LDA	18%	32%
Ours	LDA	**28%**	**34%**

CHAPTER 6

CONCLUSION

Dimensionality reduction and MIL approaches are applied to more efficiently use data in Machine Learning applications. We propose a new dimensionality reduction method based on LE that reduces the impact of outliers in the data and can provide superior results compared to LE. We also propose a method of combining dimensionality reduction and MI representations in order to learn a new integrated representation based on both MI and SI representations of our data. Our experiments show that our MI to SI representation transformation approach has value and provide a proof of concept with applications in supervised learning. We view this as a successful first milestone in this research with possible room for improvement and expanding on the research.

For the pOLE method, we see possibility for expansion by constraining the resulting representation to be non-negative. This addition would allow our method to be applied as a manifold clustering method for SI data representations. Non-negative Laplacian Embeddings have been introduced with promising results. By adding outlier resilience to that approach, we can produce even better results that can generalize the underlying structure of the data instead of being disproportionately influenced by outliers.

In addition to improvements on pOLE, the exploration and experiments applied to our MI to SI representation approach have shown a drawback when using graph based methods to learn the projection. When using a graph based approach, the largest value for the dimensions of the integrated representation, r, must be less than the number of vertices in the graph. This means that when we have MI representations with a small number of instances, we will have to reduce the number of dimensions to be low as well. This could reduce the effectiveness of our integrated representation. For future work, we will be investigating other approaches to learning the projection that do not have this limitation as

well as applying our work to data that possess a large number of instances.

In addition to that, our current approach focuses on learning a local projection of each bag and ignores any relationship with other bags. For future work, we will look into balancing between a local projection learned from each bag as well as a global projection learned from the other bags.

REFERENCES CITED

[1] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[2] Matthew D Hoffman, David M Blei, and Francis Bach. Online Learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*, 23, 2010.

[3] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, (11):559–572, 1901.

[4] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, (6):1373–1396, 2003.

[5] Oded Maron and Tomás Lozano-Pérez. A Framework for Multiple-Instance Learning. *NIPS '97 Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 570–576, 1997.

[6] Ji Feng and Zhi-hua Zhou. Deep MIML Network. *Proceedings of the 31th Conference on Artificial Intelligence (AAAI 2017)*, (2014):1884–1890, 2017.

[7] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support Vector Machines for Multiple-Instance Learning. *Journal of Chemical Information and Modeling*, (9):1689–1699, 2013.

[8] Min Ling Zhang. A k-nearest neighbor based multi-instance multi-label learning algorithm. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, pages 207–212, 2010.

[9] Dip Ghosh and Sanghamitra Bandyopadhyay. A fuzzy citation-kNN algorithm for multiple instance learning. doi: 10.1109/FUZZ-IEEE.2015.7338024.