DISCRIMINATIVE MANIFOLD EMBEDDING WITH IMPRECISE, UNCERTAIN, AND AMBIGUOUS DATA

By

CONNOR H. MCCURLEY

A ORAL QUALIFYING EXAM PROPOSAL PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AE	Autoencoder
APR	Axis-Parallel Rectangles
CCA	Canonical Component Analysis
CLFDA	Citation Local Fisher Linear Discriminant Analysis
CRF	Conditional Random Field
CT	Computed Tomography
CHL	Competitive Hebbian Learning
DD	Diverse Density
DM	Diffusion Maps
DR	Dimensionality Reduction
DSIAC	Defense Systems Information Analysis Center
EM	Expectation-Maximization
EM-DD	Expectation-Maximization Diverse Density
FA	Factor Analysis
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FPS	Frames per Second
FOV	Field of View
GAE	Graph Autoencoder
GLVM	General Latent Variable Model
GNG	Growing Neural Gas
GTM	Generative Topographic Mapping
GPS	Global Positioning System
HAC	Hierarchical Agglomerative Clustering
HS	Hyperspectral
HSI	Hyperspectral Image
ICA	Independent Component Analysis
IID	Independently and Identically Distributed
iPALM	Intertial Proximal Alternating Linearized Minimization
Isomap	Isometric Feature Mapping
KPCA	Kernel Principal Component Analysis
LDA	Fisher's Linear Discriminant Analysis
LE	Laplacian Eigenmaps
LiDAR	Light Detection and Ranging
LFW	Labeled Faces in the Wild Dataset
LLE	Locally Linear Embedding
LMNN	Large-Margin K-Nearest Neighbors
LFDA	Local Fisher Discriminant Analysis
LPP	Locality Preserving Projections
LVM	Latent Variable Model
LVQ	Learning Vector Quantization

MDS	Multi-dimensional Scaling
MI	Multiple Instance
MI-ALM	Multiple Instance Augmented Lagrangian Multiplier
MIDA	Multiple-Instance Discriminant Analysis
MidLABS	Multi-Instance Dimensionality reduction by Learning a mAximum Bag margin Subspace
MI-FEAR	Multiple-Instance Feature Ranking
MIL	Multiple Instance Learning
MIDR	Multiple Instance Dimensionality Reduction
MILES	Multiple-Instance Learning via Embedded Instance Selection
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron
MoFA	Mixture of Factor Analyzers
MoPPCA	Mixture of Probabilistic Principal Component Analysis
MWIR	Mid-wave Infrared
MVU	Maximum Variance Unfolding
NLPCA	Nonlinear Principal Component Analysis
PC	Principal Curve
PAC	Probably Approximately Correct
PCA	Principal Component Analysis
PGM	Probabilistic Graphical Model
RBF	Radial Basis Function
ROC	Receiver-Operating Characteristic
ROI	Region of Interest
S-LE	Supervised Laplacian Eigenmaps
SOA	State-of-the-Art
SOM	Self-organizing Feature Map
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
t-SNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Autoencoder
VQ	Vector Quantization

LIST OF SYMBOLS AND NOMENCLATURE

$\boldsymbol{B}$	Bag of multiple instance data instances
d	Embedding space dimensionality
D	Input space dimensionality
$\mathcal{E}$	Set of edges of an undirected, weighted graph
G	Undirected, weighted graph
h	Function that produces a class label or score
l	Binary instance-level label
L	Binary bag-level label
N	Number of samples
$\mathcal{V}$	Set of vertices in an undirected, weighted graph
$\boldsymbol{W}$	Adjacency of affinity matrix
$\boldsymbol{X}$	Data matrix of size $N \times D$
$\mathcal{X}$	Riemannian manifold
$\boldsymbol{Z}$	Latent data matrix of size $N \times d$

CHAPTER 1
INTRODUCTION

Target detection is a paramount area of research in the field of remote sensing which aims to locate an object or region of interest while suppressing unrelated objects and information. (Geng et al., 2017; Chaudhuri and Parui, 1995). Target detection can be formulated as a two-class classification problem where samples belonging to a class of interest are discriminated from a background or non-target distribution (Zare et al., 2018; Geng et al., 2017; Chaudhuri and Parui, 1995). The goal of target detection in remote sensing is to correctly identify every true positive instance (TP) in a given scene while indicating no false positives (FP) (non-target samples predicted as targets), and to identify every true negative instance (TN) without indicating any false negatives (FN) (true target samples predicted as non-target). However, there is always a trade-off in performance, and one might actually choose to miss targets to achieve a lower false-alarm rate, and vice versa (Weinberger). Traditional supervised learning approaches require extensive amounts of highly precise, sample- or pixel-level groundtruth to guide algorithmic training. However, acquiring large quantities of accurately labeled training data can be expensive both in terms of time and resources, and in some cases, may even be infeasible to obtain (Xu et al., 2014). To demonstrate these inherent labeling problems, consider the following real-world remote sensing examples, beginning with the hyperspectral target detection scenario described by Du (2017) and Bocinsky (2019):

Hyperspectral (HS) sensors collect spatial and spectral information of a scene by receiving radiance data in hundreds of contiguous wavelengths (Zare, 2008). Due to their inherent properties, HS cameras can provide a broad range of spectral information about the materials present in a scene, and are thus useful for detecting targets whose spectral signatures vary from the background. Such examples include airplanes on a tarmac or weeds in a cornfield. While HS provides nice properties for target detection, there are significant challenges when utilizing this modality. First, the spatial resolution of HS cameras can be low. As an example,

some HS cameras have spatial resolution of $30m^2$ when capturing scenes from the air. This implies that objects of interest in a scene, such as an airplane, will actually be contained in a single pixel along with other materials, such as asphalt. When performing target detection/recognition on that pixel, the HS spectra will not be distinguishable as a single, pure material, but as a sub-pixel mixture where the actual materials present as well as their corresponding proportions are unknown. Second, assuming pure target pixels are available, accurate positioning at the desired resolution may not be. For example, when analyzing a scene from an airplane or satellite, it is necessary to denote the true locations of targets on the ground using a global positioning system (GPS). It is not uncommon, however, to experience GPS error of greater magnitude than the HS pixel-level spatial resolution. This implies that a halo of uncertainty potentially surrounds every target pixel in the hyperspectral image (HSI), thus making labeling on the pixel-level difficult.

This example demonstrates inherent infeasibility to obtain accurate sample-level labels due to sensor restrictions on both resolution and accuracy. Furthermore, label imprecision and ambiguity can often be presented from subjectivity between annotators. Many applications such as medical diagnosis and wildlife identification require domain experts to provide accurate data labels (Cheplygina et al., 2019; Ruiz-Muñoz et al., 2015). However, there might not always be agreement between expert annotators and humans are prone to making mistakes. For example, when looking at computed tomography (CT) scans for malignant/benign tumors, many doctors would likely determine different pixel-level boundaries denoting a tumor, and in some cases, might even misclassify the detriment of the growth. Similarly, expert wildlife ecologists determining the identity of birds solely from their songs might be uncertain of a species due to corruptive background noise in the audio segment.

Finally, consider the scenarios shown in Figures 1-1 and 1-2. These figures show frames taken from the Defense Systems Information Analysis Center (DSIAC) MS-003-DB dataset (DSIAC, 2014) which demonstrates mid-wave infrared (MWIR) video segments of moving military vehicles taken at approximately 30 frames per second (FPS). Many computer vision

algorithms have already been developed to perform target detection using canonical bounding boxes (shown in green in Figure 1-1) (Redmon and Farhadi, 2018). However, drawing tight boxes around targets in each video frame is extremely tedious and time consuming. It would be beneficial if an annotator could provide a less-restrictive form of label, such as a relaxed bounding box (shown in blue in Figure 1-1 and bottom left in Figure 1-2) or as a small subset of target pixels such as single dot or scribble as shown in Figure 1-2. Labeling burden could be reduced even further if a single frame could be labeled at a high level as "including" or "excluding" target pixels, as shown in Figure 1-3.

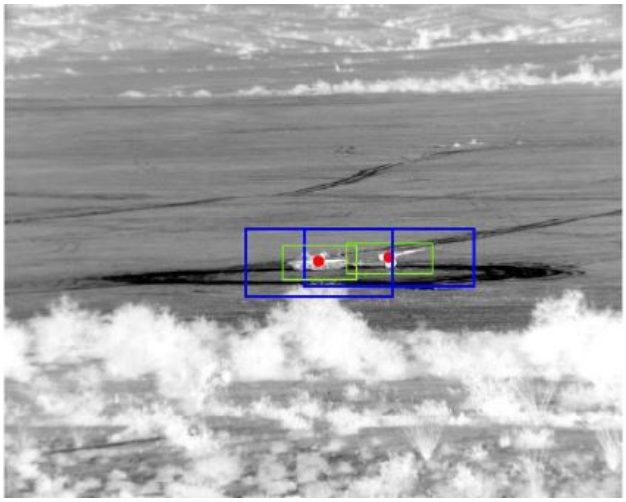


Figure 1-1: A sample frame from the DSIAC MS-003-DB MWIR dataset. Two targets are shown with canonical bounding boxes (green) and relaxed bounding boxes (blue). Red dots represent the centers of the target objects.

Techniques which can address these forms of label ambiguity while achieving comparable or better target detection than standard supervised methods can greatly ease the burdens associated with many remote sensing labeling tasks and allow for the application of pattern recognition techniques which would otherwise be infeasible.

Learning from uncertain, imprecise and ambiguous data has been an active area of research since the late 1990s and is known as *multiple instance learning* (MIL) or *weak learning*

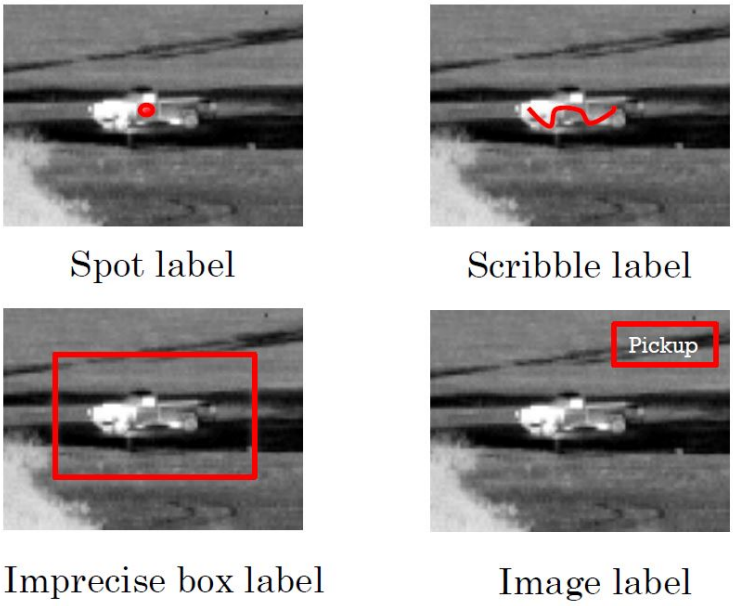


Figure 1-2: Examples of weakly-labeled infrared imagery. The images demonstrate various forms of weak groundtruth around a pickup truck taken with a mid-wave infrared camera. The images show spot, scribble, imprecise bounding box and image-level labels, respectively.

(a) (b)

Figure 1-3: Example of image-level labels for binary target detection. Image (a) is denoted to contain pixels belonging to the target class somewhere within the image, while image (b) clearly contains samples solely from the background distribution.

(Bocinsky, 2019). Supervised learning assumes that each training sample is paired with a corresponding classification label. In multiple instance learning, however, the label of each sample is not necessarily known. Instead, MIL approaches learn from groups of data points called *bags*, and each bag concept is paired with a label (Cook, 2015). Under the two-class classification scenario the bags are labeled as *negative* if all data points (or *instances*) are known to belong to the background class (not the class of interest). While the actual number of positive and negative instances may be unknown, bags are labeled *positive* if *at least one*

instance is known to belong to the target class (also called a "true positive") (Zare et al., 2018) or to contain a proportion of true target. The goal of learning under the MIL framework is to train a model which can classify a bag as positive or negative (bag-level classification) or to predict the class labels of individual instances (instance-level classification). Consider again the example shown in Figure 1-3. The figure labeled as "Target" could be considered as a positive bag because, while the image is not accompanied with pixel-level labels, it is known that a true target pixel exists somewhere within the set. Additionally, the image denoted as "Background" could be considered as a negative bag, since it does not contain any pure target pixels. Another way to formulate this problem would be to consider sets of these image patches, where a negative bag would only include samples of background patches, while a positive bag would contain at least one patch that held target or part of a target. Given data of this type, multiple instance learning could be used to discover target and/or background class representatives which could be used for class discrimination, or a classifier could be trained to label and unseen test image as containing or excluding target pixels (bag-level classification) or to label each individual pixel as belonging to the target class (instance-level classification). As with this example, the MIL problem formulation fits many remote sensing scenarios and is thus an important area of investigation (Du, 2017).

Multiple instance learning approaches in the literature can be broadly generalized into two categories: learning a generative model which effectively describes the positive and/or negative classes, or training a classifier to discriminate between target and background samples or bags (Ghaffarzadegan, 2018). The success of both learning frameworks, however, is highly dependent on the feature representations of the data and the metrics used to measure dissimilarity. Many feature learning approaches in the literature use supervised learning to obtain discriminative feature representations, or use supervised methods to fine-tune unsupervised feature extraction. *However, this cannot be done directly in MIL because of the uncertainty on the labels (Carbonneau et al., 2016).*

Another approach to obtain useful feature representations is through the application of *manifold learning*, also commonly referred to as *dimensionality reduction* (DR), *feature embedding*, *geometric machine learning* or *representation learning* in the literature. The goal of manifold learning can be posed as discovering intrinsic (often lower-dimensional) features from the data which meet an overarching objective, such as: preserving variance, finding compressed representations of data, maintaining global or local structure or promoting discriminability in the embedded space (van der Maaten et al., 2007; Bengio et al., 2012; X. Geng et al., 2005; Thorstensen, 2009). Manifold learning and dimensionality reduction have been studied extensively in the literature and have been used for visualization, classification, redundancy removal, compression and data management, improving computational tractability and efficiency, combating the curse of dimensionality and obtaining more appropriate measures of dissimilarity (Bishop et al., 1998; Nickel and Kiela, 2017; Talmon et al., 2015; Tenenbaum et al., 2000; X. Geng et al., 2005; Palomo and Lopez-Rubio, 2017; Kohonen, 1990; Kegl et al., 2008; Bengio et al., 2012).

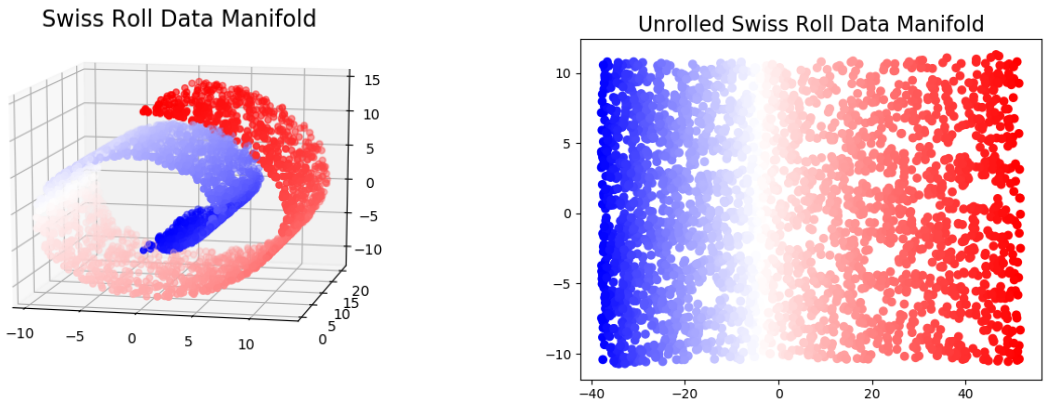
Dimensionality plays a significant role in determining class separability. Enough features should be incorporated as to adequately describe a class of interest, yet too many features may introduce redundancy, and thus be detrimental to the learning process. The curse of dimensionality states that the number of samples needed to characterize the space spanned by an entity grows exponentially with dimensionality (Murphy, 2012; Theodoridis and Koutroumbas, 2008). This fact is overwhelming in the context of remote sensing, as it is often both difficult to acquire large quantities of labeled data and there are often only few samples available to describe the target class (such as in air- or space-born HSI target detection). Additionally, dissimilarity metrics often break down in high dimensional spaces, making application of traditional classifiers difficult. While it has been argued in the literature that high-dimensionality (data, parameterization, model complexity) is more amenable for classification (Anderson et al., 2013; Breiman; Gorban and Tyukin, 2018; Huang et al., 2019), methods taking advantage of high-dimensionality are often inscrutable. However, effective application of machine learning

methods in real world applications often requires model interpretability. For example, it is important for doctors to be able to visualize models and explain to patients why an algorithm inferred an outcome. Also, simplicity allows developers to understand confusers for target detection algorithms which may be imperative in defense applications or for environmental scientists to link combinations of features to biologically-meaningful hypotheses. *Therefore, it is intuitive that approaches should be developed which can combat the curse of dimensionality and provide more appropriate similarity metrics while also addressing the problems associated with uncertain, imprecise, and ambiguously labeled training data.*

The underlying assumption in using manifold learning for discrimination is that opposing classes either reside on separate manifolds or different regions of a joint, intrinsic manifold, where samples of the same class are close and samples from disparate classes are metrically far. Essentially, the governing class distributions can be described by hyper-surfaces which follow constraints on properties such as continuity and smoothness (Belkin and Niyogi, 2004). The goal of learning in this scenario is to discover embedding functions from the input feature space to a lower-dimensional embedding space where the transformed feature representations allow for improved class discrimination.

Methods for representation learning have recently gained in popularity because they typically result in high levels of classification accuracy (Bengio et al., 2012). Some of these methods learn features in a supervised manner to obtain more discriminative representations. As mentioned previously, this learning cannot be done directly in MIL because of the uncertainty on the labels. Thus, *adaptation of discriminative feature learning methods would be beneficial to MIL* (Carbonneau et al., 2016), yet this area of research has scarcely been explored. The first true experimentation demonstrating the feasibility of dimensionality reduction for classification on MIL data was performed by (Sun et al., 2010). Sun et al. motivated their work by first showing that (as expected) Principal Component Analysis (PCA) provided poor separation between positive and negative bags because of the lack of utilized label information. Additionally, Linear Discriminant Analysis (LDA) was used to project bags into a latent space

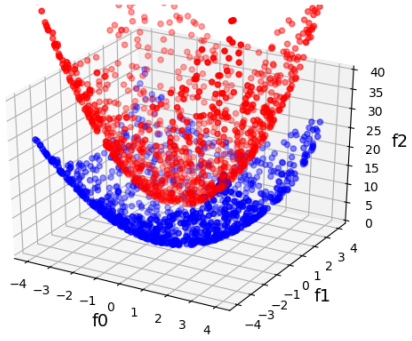
which maximized between-bag separation, while minimizing within-bag dissimilarity. However, LDA often mixed the latent bag representations due to the uncertainty of negative sample distributions in the positive bags. Thus, Sun's motivating experiments reiterated the long-known difficulties associated with applying traditional manifold learning approaches to MIL problems. To address these issues, Sun proposed Multiple Instance Dimensionality Reduction (MIDR) which optimized an objective through gradient descent to discover sparse, orthogonal projection vectors in the latent space. Their approach relied on fitting a distribution of negative instances and applying maximum likelihood estimation. This approach was later extended by Zhu et al. (2018) in attempt to improve sparsity. Additionally, the works in (Ping et al., 2010; Saehoon Kim and Seungjin Choi, 2010; Chai et al., 2014) use adaptations of LDA to project data into a low-dimensional space. All of these methods rely on finding linear projections which will adequately separate positive and negative bags in the embedding space. However, these approaches fail when the the target and background data exhibit highly-curved structure in the feature space. For example, consider the data shown in Figure 1-4. This image shows the popular Swiss Roll dataset, which is a 2-dimensional manifold folded in 3-dimensions. Assume that the data classes lie on separate ends of the manifold, and that samples are governed by a smooth labeling function. In Figure 1-4a, the red samples denote the target class while the blue represent the background class. If using Euclidean (straight line) distance to measure dissimilarity, it would appear that many red samples are (untruly) close to the blue. However, if an alternative distance metric such as geodesic distance (around the curve) is used to measure dissimilarity between samples, the true class distributions are better represented. Figure 1-4b shows the unfolding of the manifold according to geodesic distance. It can be observed that, after the unfolding, the classification problem was transformed from a nonlinear to a linear one. Additionally, a dimension of the data was deemed unnecessary. It would, therefore, be beneficial to develop multiple instance dimensionality reduction approaches which are capable of learning nonlinear manifold structure.



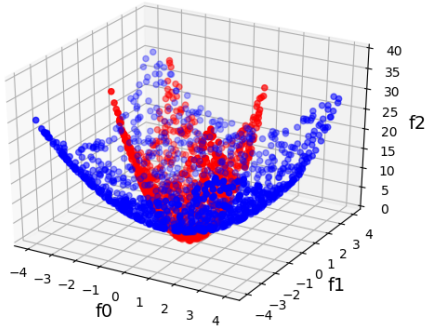
(a) (b)

Figure 1-4: (a) This dataset is known as the Swiss Roll and it depicts a 2-dimensional manifold embedded in 3 dimensions. (b) The Swiss Roll unfolded according to the geodesic path around the manifold.

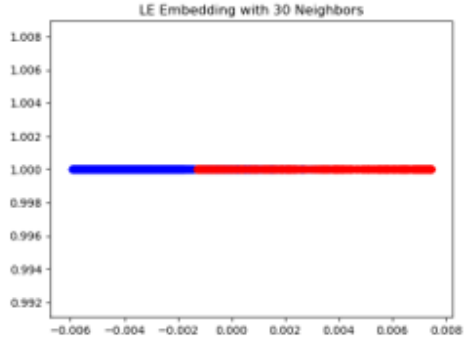
Dimensionality reduction approaches in the literature require instance-level labels or neglect class label information entirely. However, sample-level labels are not available under the MIL framework and unsupervised approaches are typically sub-optimal for discrimination. This point is demonstrated by the embeddings of 3-dimensional quadratic surfaces as shown in Figures 1-5 and 1-6. These figures demonstrate a set of separable and non-separable quadratic surfaces, respectively. The use of a traditional, unsupervised dimensionality reduction method to project the data into a 1-dimensional space managed to retain topological ordering of the samples, but failed to promote class discriminability, as shown in Figures 1-5c and 1-5d. As demonstrated in Figures 1-6c and 1-6d, however, a supervised version of the same algorithm learned a mapping which was able to separate the classes in the latent embedding space for both the separable and non-separable quadratic surfaces. This embedding used sample-level labels which, as mentioned, are not available in MIL. Therefore, the objective of the proposed work is, given only sets of instance and bag-level labels such as those shown in Figures 1-6a and 1-6b, to develop techniques for nonlinear dimensionality reduction and manifold learning which promote class separation of individual instances in the embedding



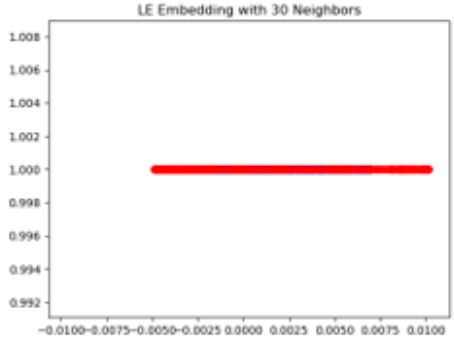
(a) Separable 3-dimensional quadratic surfaces.



(b) Non-separable 3-dimensional quadratic surfaces.



(c) 1-dimensional embedding of separable quadratic surfaces.

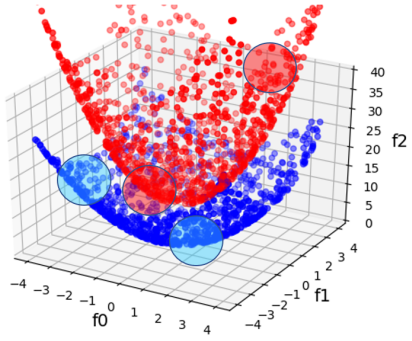


(d) 1-dimensional embedding of non-separable quadratic surfaces.

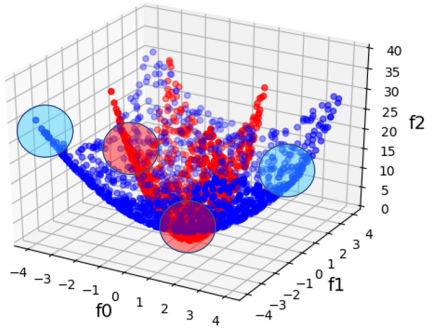
Figure 1-5: Unsupervised embedding of quadratic surfaces using Laplacian Eigenmaps with a K-nearest neighbors graph.

space. Ideally, instance-level classification accuracy in the embedding space with MIL will match the performance of strictly supervised methods, even with increased label imprecision.

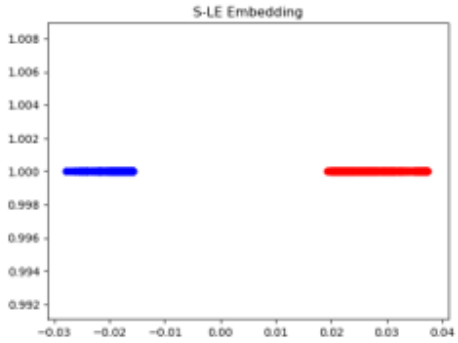
To address the points mentioned, I propose the following. During this project, techniques will be explored for use in instance-level classification given uncertain and imprecise groundtruth. These methods will be developed as universal approaches for discriminative manifold/feature representation learning and dimensionality reduction and will be evaluated on a variety of sensor modalities, including: mid-wave IR, visible, hyperspectral and multispectral imagery, LiDAR and more. *The aim of this project is to develop nonlinear dimensionality reduction methods which promote class discriminability and are simultaneously capable of*



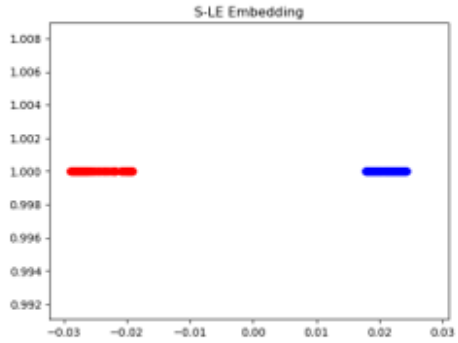
(a) Separable 3-dimensional quadratic surfaces and examples potential bag splits. Blue circles represent negative bags while red circle denote positive bags.



(b) Non-separable 3-dimensional quadratic surfaces and examples potential bag splits. Blue circles represent negative bags while red circle denote positive bags.



(c) 1-dimensional embedding of separable quadratic surfaces using Supervised Laplacian Eigenmaps.



(d) 1-dimensional embedding of non-separable quadratic surfaces using Supervised Laplacian Eigenmaps.

Figure 1-6: Embedding of quadratic surfaces using Supervised Laplacian Eigenmaps.

addressing uncertainty and imprecision in training data groundtruth. Roughly, the following research questions will be addressed during the scope of this project:

1. Supervised and semi-supervised manifold learning have proven effective at discovering low-dimensional data representations which provide adequate class separation in the latent space. However, only a handful of manifold learning procedures consider data that is weakly or ambiguously labeled. To address this gap in the literature, a method for weakly-supervised manifold learning will be developed. How does this method of manifold construction compare to state-of-the-art (SOA) manifold learning techniques as well as alternative MI dimensionality reduction methodologies for instance-level label prediction?

2. In conjunction with dimensionality, the use of metric embedding has been shown to promote class separability in the latent space. However, metric embedding typically requires knowledge of instance-level labels. Using only weak, bag-level labels, a method for metric embedding will be developed and utilized with the manifold learning approach in Objective 1 to potentially improve class separation of individual instances. Additionally, a procedure to select the most influential examples for training will be developed.

3. Do the proposed methods facilitate concept learning/selection? Using alternative state-of-the-art MIL approaches, are the selected target instances/concepts more discriminable with the proposed methods than without? How do the proposed methods compare to the alternatives in terms of representation dimensionality, computational complexity, and promotion of discriminability?

Experiments will be conducted on both synthetic data and real applications such as target detection, scene understanding and semantic segmentation in remote sensing imagery. Datasets will include the DSIAC MS-003-DB Algorithm Development Database, MUUFL Gulfport, Faces in the Wild (LFW) and benchmark MIL classification datasets (DSIAC, 2014; Gader et al., 2013; Du and Zare, 2017; Glenn et al.; Huang et al., 2007). Initial results demonstrate the aptitude of the proposed approaches and suggest further development and evaluation of these methods.

CHAPTER 2
BACKGROUND

This chapter provides a comprehensive review of literature pertinent to the work proposed in this document. First, a review is given in Section 2.1 on the multiple instance learning framework for learning from weak and ambiguous annotations. Notation is given and summaries of methods used in bag and instance-level classification and ranking are discussed. Review is then provided in Section 2.2 on manifold learning, including classic approaches as well as supervised and semi-supervised adaptations. This is followed by an entire section dedicated to reviewing manifold learning and dimensionality reduction techniques specifically tailored for use with weak labels (Section 2.3). The last part of this chapter (Section 2.4) reviews the existing literature on metric embedding, focusing heavily on the utilization of contrastive and triplet-based loss evaluation. Reviews describe basic terminology and definitions. Foundational approaches are elaborated and advances are addressed.

2.1 Multiple Instance Learning

Multiple Instance Learning (MIL) was originally proposed by Dietterich et al. (1997) as a method to handle inherent observation difficulties associated with drug activity prediction. This problem, among many others, fits well into the framework of MIL where training labels are associated with sets of data points, called *bags* instead of each individual data points, or *instances*. Under the *standard MIL assumption*, a bag is given a "positive" label if it is known that *at least one* sample in the set represents pure or partial target. Alternatively, a bag is labeled as "negative" if does not contain any positive instances (Carbonneau et al., 2016). Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ be training data where D is the dimensionality of an instance, $\boldsymbol{x}_n$, and N is the total number of training instances. The data is grouped into K *bags*, $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, with associated binary bag-level labels, $\mathcal{L} = \{L_1, \ldots, L_K\}$ where

$$L_k = \begin{cases} +1, & \exists \boldsymbol{x}_{kn} \in \boldsymbol{B}_k^+ \ni l_{kn} = +1 \\ -1, & l_{kn} = -1 \quad \forall \boldsymbol{x}_{kn} \in \boldsymbol{B}_k^- \end{cases} \tag{2-1}$$

and $\boldsymbol{x}_{kn}$ denotes the n^{th} instance in positive bag $\boldsymbol{B}_k^+$ or negative bag $\boldsymbol{B}_k^-$ (Zare et al., 2018) and $l_{kn} \in \{-1, +1\}$ denotes the instance-level label on instance $\boldsymbol{x}_{kn}$. Figure 2-1 demonstrates the concept of MIL bags. The objective of learning under MIL is, given only bag-level label information, to fit a model which can perform one of the following tasks: classification, regression, ranking or clustering (Carbonneau et al., 2016).

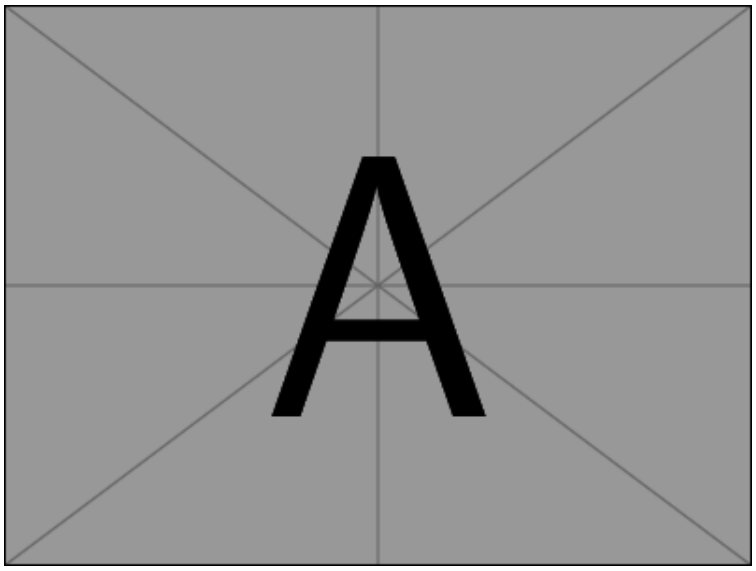


Figure 2-1: Placeholder for examples of positive and negative bag concepts

2.1.1 Multiple Instance Learning with Manifold Bags

It should be noted that while the aforementioned MIL formulation is standard in the literature, Babenko et al. (2011) introduced the framework of MIL using manifold bags . Their work claimed that instead of finite sets of instances, bags are inherently better represented as low-dimensional manifolds in a high-dimensional feature space. They considered the scenario of classifying whether or not an image contained a face. In this scenario, the entire image was considered to be a bag and patches of the image represented individual instances. It was assumed that the collection of instances collectively formed a low-dimensional manifold. The representation of bags as low-dimensional manifolds over the domain of instances is related to the *Multiple-Instance Learning via Embedded Instance Selection* (MILES) algorithm

(Yixin Chen et al., 2006), which embeds bags into a feature space defined by similarities between instances. By giving more importance to the features representing samples in local neighborhoods around the most-likely-cause concept, MILES can be interpreted as representing bags according to local, manifold approximations centered on the instances which most-likely provide the bag-level labels. While related to bag-embedding methods such as MILES, the work by Babenko et al. (2011) laid the groundwork for learning with manifold bags and proved probably approximately correct (PAC) learnability under this (more general) framework. While it was worth mentioning this work as it aligns with the objective of manifold learning discussed in the proposed work, the remainder of this literature review focuses solely on the standard MIL formulation.

2.1.2 Tasks

Multiple instance learning in the literature can be broadly categorized into four tasks: classification, regression, ranking and clustering (Carbonneau et al., 2016). MIL classification can be performed at either the bag or instance level. The goal is to assign a class label to either the set of instances or the individual instances themselves. MIL regression consists of assigning a real-valued label to a bag (or instance) instead of a class label. A few methods have been proposed to rank bags or instances instead of assigning a class label or score. This problem differs from regression because the goal is not to obtain an exact real-valued label, but to compare the magnitude of scores to perform sorting. The clustering task can also be performed at the bag or instance-level. Bag-level clustering involves separating a set of unlabeled bags into distinct groupings based on a measure of similarity. Alternatively, clustering is often performed on the instances within individual bags in attempt to quantize the data into positive and negative concepts. Classification and ranking are the most pertinent tasks to the proposed work, and will thus be discussed in detail.

2.1.3 Multiple Instance Classification

The standard supervised learning task is to learn a classifier based on a training set of feature vectors, where each feature vector is paired with and associated class label. In the

multiple instance classification task, the goal is to learn a classifier based on a training set of bags, where each bag is a set of feature vectors known as instances. In this setting, each bag is paired with an associated binary class label; however, the labels of each instance in the sets are unknown.

2.1.3.1 Space Paradigms

The multiple instance classification problem has been formulated under three paradigms: instance-space, bag-space and embedded-space (Amores, 2013). Each paradigm is categorized according to how information presented in the MI data is exploited. In the *instance-space* paradigm, the discriminative information is considered to lie at the instance-level. An instance-level classifier is trained to separate the true positive instances from the true negative instances. Given an instance-level classifier, a bag-level classifier can be developed by simply aggregating the instance-level scores in a test bag. This paradigm is based on local, instance-level information. In the *bag-space* paradigm, the discriminative information is considered to lie at the bag-level. Under this paradigm, each bag is treated as a whole entity, and the learning process discriminates between entire bags. This paradigm is based on global, bag-level information. Considering that the bag space is a non-vector space, current BS methods make use of non-vectorial learning techniques which define distance metrics as a way to compare bags. In the *embedded-space* paradigm, each bag is mapped to a single feature vector which captures the relevant information about the entire bag. Consequently, the learning problem transforms into a standard supervised problem, where each feature vector is paired with an associated (bag-level) label. Similar to the bag-space paradigm, the embedded-space paradigm is also based on global, bag-level information. However, the difference between the two paradigms lies in the way bag-level information is extracted. In the bag-space paradigm, information is extracted implicitly through the definition of the distance or kernel function which maps two or more bags to a real number measuring the similarity between them. Alternatively, information is extracted explicitly in the embedded-space paradigm through the definition of the mapping

function, which aggregates the instances contained in an individual bag to form a single real-valued vector representing the bag in the embedded vector space. Figure 2-2 demonstrates the differences between standard supervised classification and the three MIL classification space paradigms. Apart from space paradigm, MIL classification methods in the literature can be organized according to the their primary approaches toward learning. A review of prominent MIL classification methods in the literature is provided, categorized according to learning approach.

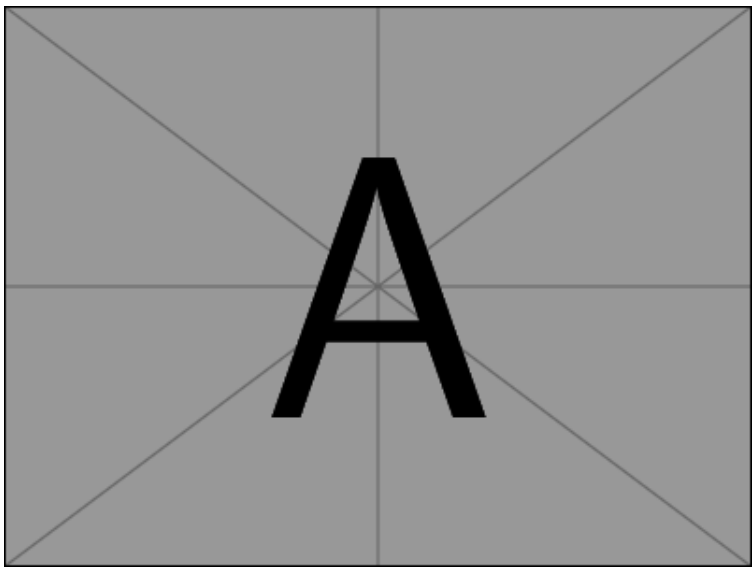


Figure 2-2: MIL classification space paradigm.

2.1.3.2 MIL Classification Approaches

MIL classification approaches in the literature can be categorized by the underlying principle used for learning. The categories discussed in this review are: Axis-Parallel Concepts, Maximum Likelihood, Distance-Based, Maximum Margin, Deep Learning, Probabilistic Graphical Methods, Dictionary Learning and Ensembles of Classifiers. Each approach is reviewed in the following.

Learning Axis-Parallel Concepts:. Learning Axis-Parallel Concepts are among the first group of methods used to solve MIL problems (Dietterich et al., 1997). The foundation of this category is based on the method of *Axis-Parallel Rectangles* (APR), proposed by Dietterich et

al. in the 1990's. An axis-parallel hyper-rectangle is a set of thresholds (one for each feature dimension) that is used to discriminate between two classes. It can also be viewed as an overlap or aggregation region of true positive instances in the feature space. The goal of APR is to find an axis-parallel hyper-rectangle in the feature space to represent the target concept (Ghaffarzadegan, 2018; Bocinsky et al., 2019; Jiao, 2017). A disadvantage of these approaches is that most of the data is ignored when working with large bags.

Maximum Likelihood:. Similar to traditional *Maximum Likelihood Estimation* (MLE), the objective of maximum likelihood in the MIL setting is to train a classifier which maximizes the likelihood of the data. The most prominent of these methods is *Diverse Density* (DD) (Maron and Lozano-Pérez, 1998), which considers each bag as a manifold describing the instances. The goal of learning is to discover a prototype from the training data which maximizes the DD measure. Diverse Density is essentially a measure of the intersection of positive bags minus the union of the negative bags. By maximizing DD, a point of intersection can be found which is close to at least one instance from the positive bags while being as far as possible from all points in the negative bags (Ghaffarzadegan, 2018). Zhang et al. later proposed *Expectation-Maximization Diverse Density* (EM-DD) (Zhang and Goldman, 2002). EM-DD extends DD by viewing the relationships between instances and their corresponding bag's labels as latent variables. In other words, it asks which instance in the set corresponds to the bag's label (Du, 2017). In the *E-step*, an the instance from each bag which is the most probable for providing the bag its label is selected. Then in the *M-step*, a new concept point is found by maximizing DD with gradient ascent. This process is iterated until a stopping criteria is met, however, it will stop naturally as there only a finite number of instance combination that the algorithm can pick. Multiple instance maximum likelihood approaches have been used in a variety of problems, such as: stock selection, person identification, scene classification, hyperspectral target classification and explosive hazard detection (Maron and Lozano-Pérez, 1998; Maron and Ratan, 1998; Zare et al., 2018, 2015; Bocinsky et al., 2019; McCurley et al., 2019).

Distance-Based:. A simple approach for classification in the supervised learning paradigm is to compare distances of test points to samples in the training set. Gartner et al. defined the MI-Kernel (Gärtner et al., 2002) by regarding each bag as a set of features and applying a set kernel directly to compare similarity. The *Citation-kNN* algorithm (Du, 2017; Carbonneau et al., 2016; Zhou and Zhang, 2003; Saehoon Kim and Seungjin Choi, 2010) is a nearest-neighbor style classifier which borrows the idea of scientific citers and references when considering a bag's label. *References* are simply the nearest neighbors of a bag, while *citers* are the bags which have the query bag in it's C-nearest neighbors. The vanilla citation-kNN uses the *minimal Hausdorff distance* to measure bag similarity. The Hausdorff distance between two bags $\boldsymbol{B}$ and $\boldsymbol{B}'$ is defined as:

$$H(\boldsymbol{B}, \boldsymbol{B}') = \max\{h(\boldsymbol{B}, \boldsymbol{B}'), h(\boldsymbol{B}', \boldsymbol{B})\} \tag{2-2}$$

and

$$h(\boldsymbol{B}, \boldsymbol{B}') = \max_{\boldsymbol{b} \in \boldsymbol{B}} \min_{\boldsymbol{b}' \in \boldsymbol{B}'} ||\boldsymbol{b} - \boldsymbol{b}'|| \tag{2-3}$$

where $\boldsymbol{b}$ and $\boldsymbol{b}'$ are instances in bags $\boldsymbol{B}$ and $\boldsymbol{B}'$, respectively. Intuitively, the minimal Hausdorff distance is the smallest value H such that every instance in $\boldsymbol{B}$ has a point of $\boldsymbol{B}'$ within distance H, and every instance in $\boldsymbol{B}'$ has an instance in $\boldsymbol{B}$ within a distance of H. Variants of citation-kNN include Bayesian citation-kNN and fuzzy citation-kNN (Du, 2017). Additionally, the minimal Hausdorff distance has been substituted for the Earth Mover's Distance (EMD), Chamfer distance and specialized bag-distance kernels (Amores, 2013).

Two alternative distance-based MIL classifiers are MIGraph and miGraph (Zhou et al., 2009). Both methods consider bags as graphs to capture interdependence between instances. The distance measures used to compare bags is what separates the two methods. MIGraph explicitly maps every bag to an undirected graph and uses a graph kernel or metric such as graph edit distance to discriminate between positive and negative bags. Alternatively, miGraph

implicitly constructs graphs by defining affinity matrices between instances and uses clique information to help distinguish positive from negative bags.

Maximum Margin:. The concept of weakly-supervised maximum margin learning has been explored under a variety of techniques, the primary being *Support Vector Machines* (SVM) and *metric embedding*, which will be discussed in Section 2.4. Andrews et al. (2002) proposed two SVM methods under the MIL framework, namely, mi-SVM and MI-SVM, for instance-level and bag-level classification, respectively. The goal of a SVM is to train a classifier to maximize the margin between a small subset of training examples (called *support vectors*) and the decision hyper-plane (Murphy, 2012). Both MIL SVM methods follow a similar procedure to EM-DD. In MI-SVM, each positive bag is represented by a single instance which is considered to be the "most positive instance" in the bag. Optimization alternates between learning a decision boundary with SVM and selecting the positive bag representatives given the new classifier. In contrast, mi-SVM considers all points in the positive bags while learning the decision boundary. Every point in the positive bags is provided a positive label. The instance labels are refined iteratively under the constraint of the standard MIL assumption, that at least one instance from each positive bag must lie on the positive side of the decision hyper-plane (Cao et al., 2016). MissSVM is a semi-supervised max-margin approach which considers the instances in positive bags as unlabeled, and enforces a constraint that at least one of them is positive (Zhou and Xu, 2007). DD-SVM and *Multiple-Instance Learning via Embedded Instance Selection* (MILES) are both embedding-space methods which convert MIL into a standard supervised problem (Yixin Chen et al., 2006). DD-SVM trains an SVM in a feature space constructed from a mapping defined by the local maximizers and minimizers of the DD function. MILES maps each bag into a feature space defined by the instances in the training bags via an instance similarity measure. A 1-norm SVM is applied to simultaneously select the important features and construct classifiers (Ruiz-Muñoz et al., 2015). Additionally, Xiao et al. (2017) developed an ensemble method in which the base classifier enforces a margin

between optimal hyper-spheres while enclosing at least one instance from each positive bag inside the ball.

Neural Networks and Deep Learning:. Recent developments in deep learning have also made their way into the MIL literature under the assumption that useful features can be learned by the networks using only bag-level labels. Gao et al. (2017) used *convolutional neural networks* (CNN) with count-based region selection to perform weakly-supervised object localization. Ilse et al. (2018) modeled the MIL problem as learning the Bernoulli distribution over the bag labels, where the label probability was parameterized by a neural network. Ilse's approach employs attention-based MIL pooling as a way to visualize which instances the network selects as being positive. A multi-instance multi-scale CNN to detect regions of interest in medical images was proposed by Li et al. (2019). Li's work introduced "top-k pooling" to aggregate feature maps of varying scales and spatial dimensions, allowing the model to be trained using weak, MIL annotations. Wang et al. (2020) explored the use of a U-Net segmentation network architecture to obtain pixel-level ground-cover classification from image-level labels. A discriminative *variational autoencoder* (VAE) was used by Ghaffarzadegan (2018) to maximize the difference between latent representations of positive and negative instances. Tu et al. (2019) developed an end-to-end *graph neural network* which treats each bag as a graph. Each bag is passed through the network to obtain a feature representation encapsulating the structural information present in the bag. Deep learning MIL approaches have been explored in a wide variety of applications, including: retinal image classification (Tu et al., 2019), histopathology classification (Ilse et al., 2018), object localization (Gao et al., 2017) and region-of-interest proposal in medical images (Li et al., 2019).

Probabilistic Graphical Methods:. *Probabilistic graphical models* (PGMs) are powerful tools used to capture inter-relations between random variables and learn structured models. In some problems, data exhibits an underlying structure between instances or bags that is more complex than simple co-occurrence. Capturing this structure may lead to better classification performance (Carbonneau et al., 2016). Deselaers and Ferrari (2010) proposed a multi-instance

conditional random field, MI-CRF. In this method, bags are modeled as nodes in a conditional random field (CRF), where each node can take one of the instances in the bag as its state. Classification corresponds to selecting one instance (positive bag) or selecting no instances (negative bag). Instance selection is formulated as inference in the CRF. This lets all bags to be considered jointly in training and testing. Thus, bags are jointly classified based on unary instance classifiers and pairwise dissimilarity measurements. Hajimirsadeghi and Mori (2017) introduced a max-margin classification scheme using Markov networks. Yuksel et al. (2015, 2012) developed a multiple instance *Hidden Markov Model* (MI-HMM) for use in landmine detection. In this scenario, each bag is associated with a label, however, the bags can be composed of time sequences of variable length. A noisy-OR relationship is assumed between the sequences within each bag and the joint probability of the bags of sequences and the corresponding labels for the bags is maximized with a stochastic expectation maximization. PGMs have proven to work well on MIL problems exhibiting time series data and data with structural dependence.

Dictionary Learning:. *Dictionary Learning* is a method of learning how to reconstruct a dataset from a much smaller set of building blocks called *atoms* (Cook, 2015). Given a training data set, the goal is to learn the set of atoms and sparse weights which reconstruct the data. At test, a bag or instance can be classified based on the atoms which effectively reconstruct the sample. Multi-Instance Dictionary Learning (MIDL) uses bag-level information with a least angle regression to alternatively learn a dictionary which represents the training data and regression weights for classification. Max-Margin Multiple Instance Dictionary Learning (MMDL) adopts the idea of the bag of words (BoW) model and trains a set of linear SVMs as codebooks (Jiao, 2017). Functions of Multiple Instances (FUMI) is a supervised technique which tries to learn target and background dictionaries such that a target instance can be written as a linear combination of a single target concept and multiple background atoms . This formulation considers target instances that may contain portions of background signature, as with sub-pixel target detection in hyperspectral imagery. A known problem with

FUMI is that it does not work well with noisy labels. To account for this problem, extended Functions of Multiple Instances (eFUMI), uses bag-level labels to identify the data (Jiao, 2017; Cook, 2015). A function is built in to determine whether a point labeled as target actually contains a portion of the target dictionary atoms. Task-driven extended Functions of Multiple Instances (TD-eFUMI) adopts the MI aspect of eFUMI and Task-Driven Dictionary learning to simultaneously learn target and background dictionaries in conjunction with a classifier (Cook et al., 2016). Zare et al. proposed the Multiple Instance Adaptive Cosine/Coherence Estimator and Spectral Matched Filter (MI-ACE and MI-SMF) (Zare et al., 2018). These methods learn discriminative prototypes under the multiple instance learning framework to classify instances. However, these methods inherently consider only a single target concept. In order to capture intra-class variation among target instances, Multi-Target MI-ACE/SMF were proposed (Bocinsky, 2019). Finally, Jaio et al. presented Multiple Instance Hybrid Estimator (MI-HE) to learn multiple target and background concepts by maximizing the probability that positive bags are labeled as positive and negative bags are labeled as negative under a noisy-OR model (Jiao et al., 2018; Bocinsky, 2019). Multiple instance dictionary learning problems have been applied successfully to sub-pixel hyperspectral target detection and landmine detection tasks (Bocinsky, 2019; Zare et al., 2015, 2018; Cook et al., 2016; Jiao et al., 2018).

Ensembles of Classifiers:. *Ensemble learning* paradigms train multiple versions of a base classifier and aggregate the results to achieve a stronger classifier than any of the individuals. Ensemble methods are typically broken into two realms: *bagging* and *boosting*. Bagging employs boostrap sampling to generate several training subsets from the original training set, then trains a learner on each data subset. The predictions from each component learner are aggregated in order to provide a final class score. Zhou and Zhang (2003) studied whether ensemble learning paradigms could be used to enhance MI learners by applying bagging on base MI learners, namely, Diverse Density and Citation K-NN. Random Forest classifiers operate under the bagging paradigm and use a technique called *divide-and-conquer*. These classifiers deploy an ensemble of decision trees which iteratively divide the feature space and make

simple thresholding decisions. The predicted class labels provided by each tree in the forest are combined to give a final class label. Leistner et al. (2010) proposed MIForest which combines the random forest learning algorithm with MIL. Since only bag-level labels are known, MIForest treats the label of each instance as a random variable defined over a space of probability distributions. The instance labels are disambiguated by iteratively searching for distributions which minimize the overall classification error.

The other paradigm of ensemble learning is called boosting. The goal of boosting is, using the entire training set, to find a weighted combination of weak learners (that may perform only slightly better than chance), such that the combination produces a strong classifier with high classification accuracy (Zhang et al., 2006). Several MI boosting approaches have been proposed in the literature. Section 2.1.4 discusses a popular variant, MIL-Boost, in detail.

2.1.4 Multiple Instance Boosting (MIL-Boost)

Gradient Boosting Overview. In the standard supervised learning setting, the goal of binary classification is to learn a classification function $h : \mathcal{X} \rightarrow \mathcal{L}$ which maps data in $\mathcal{X}$ to a binary class label $\mathcal{L} \in \{-1, +1\}$. The objective of boosting is to train a classifier of the form

$$\boldsymbol{h}(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \tag{2-4}$$

where each $h_t : \mathcal{X} \rightarrow \mathcal{L}$ is a *weak learner* whose performance may only be slightly above chance, and the weights α_t are each weak learners' relative importance (Babenko et al., 2008). *Boosting* is a specific case of *ensembling* which combines multiple weak learners into a single *strong* classifier with low classification error. In each phase of training, samples classified incorrectly are given more weight in order to improve classification performance in the next iteration. The response of each weak classifier h_t is given as the maximum response over all instances in the training set:

$$h_t = \arg\max_{h} \sum_{n=1}^{N} w_n h(\boldsymbol{x}_n) \tag{2-5}$$

MIL-Boost. Boosting under the MIL framework was originally proposed by Zhang et al. and is known as *MIL-Boost* (Zhang et al., 2006). Under MIL, it is assumed that every instance has a true label $l_{kn} \in \{-1, +1\}$. A bag is labeled positive if at least one of its instances are positive, and a bag is labeled negative if every instance is negative. This means that the label of a bag is provided as the max label over all instances in the bag:

$$L_k = \max_n(l_{kn}) \tag{2-6}$$

In this setting, the goal is to learn a classifier $\boldsymbol{h}$ using only bag-level labels such that $\max_n(\boldsymbol{h}(\boldsymbol{x}_{kn})) = L_k$. The score given to a sample is $l_{kn} = \boldsymbol{h}(\boldsymbol{x}_{kn})$ and $\boldsymbol{h}(\boldsymbol{x}_{kn}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}_{kn})$ which is a weighted sum of predicted labels from each of the weak classifiers. Obviously, the sign of the prediction from the strong classifier provides the class label for the instance.

A natural objective is to minimize the negative log-likelihood between instances and their predicted labels. This can also be done at the bag-level. The probability that an instance is positive is given by the standard logistic function

$$p_{kn} = \frac{1}{1 + \exp(-l_{kn})} \tag{2-7}$$

and the probability that bag k is positive is given by a "noisy OR", $p_k = 1 - \prod_{n \in k}(1 - p_{kn})$. Therefore, the likelihood assigned to a set of training bags is given by:

$$\mathcal{L}(\boldsymbol{h}) = \prod_{k=1}^{K} p_k^{L_k}(1 - p_k)^{(1-L_k)} \tag{2-8}$$

where $L_k \in \{0, 1\}$ is the actual label of the bag.

Following the idea of gradient boosting, the weight on each training instance is given as the derivative of the log-likelihood with respect to a change on the score given to the instance. Thus gradient descent can be used to update the the strong classifier. Pseudo-code for MIL-Boost is provided in Algorithm 1.

Algorithm 1 MIL-Boost

Input: Dataset $\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$

1: **for** $t = 1$ to T **do**
2: Compute weights $w_{kn} = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_{kn}}$
3: Train weak classifier h_t using weights $|w_{kn}|$
$h_t = \arg\min_h \sum_{kn} \mathbf{1}(h(\boldsymbol{x}_{kn} \neq L_k))|w_{kn}|$
4: Find α_t via line search to minimize $\mathcal{L}(\boldsymbol{h})$
$\alpha_t = \arg\min_\alpha \mathcal{L}(\boldsymbol{h} + \alpha h_t)$
5: Update strong classifier $\boldsymbol{h} \leftarrow \boldsymbol{h} + \alpha_t h_t$
6: **end for**

Each round of boosting consists of updating the weak learners according to the instances they misclassified and updating the strong classifier according to the new weights calculated over the weak learners. The goal of MIL-Boost is to be able to correctly classify each instance in a test bag as being positive or negative such that the label of the bag is given as the maximum label over all instances in the bag.

2.1.5 Multiple Instance Ranking

Another primary task in MIL is *ranking* (Carbonneau et al., 2016). The ranking problem is different from classification because, instead of providing a binary class label, the objective is to order a set of bags (Bergeron et al., 2008; Bergeron et al., 2012) or instances (Hu et al., 2008) according to preference for a particular task. Ranking is a key task under *Preference Learning*, or learning to predict preferences on a set of alternatives, which are often represented in the form of an order relation (Fürnkranz and Hüllermeier, 2003). The statement that $\boldsymbol{x}$ is preferred to $\boldsymbol{x}'$ can be simply expressed as an inequality relation $f(\boldsymbol{x}) > f(\boldsymbol{x}')$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are instances, and f defines a preference function. For example, given a news story about the Olympics, one might prefer to give it the label "sports" rather than "politics" or "weather". Alternatively, one might prefer to label one bag or instance as "positive" over another. In machine learning, the preference learning problem is often analyzed in two cases: *learning instance preference* and *learning label preference* (Chu and Ghahramani, 2005). Under the scenario of learning instance preferences, the training set consists of a set of pairwise preferences between instances. The objective is to learn the underlying ordering from the set

of pairwise distances (such as ordering bags from the "most positive" to "least positive"). Alternatively, the goal of label preference learning is to order a pre-defined set of labels for each individual instance (such as ordering the preference of "positive" or "negative" on each individual bag or instance) (Dekel et al., 2004; Aiolli and Sperduti, 2004).

Ranking under the multiple instance framework was proposed by Bergeron et al. (2008) for predicting hydrogen atom grouping in computational chemistry. The method proposed by Bergeron is called *MIRank* (Bergeron et al., 2012). MI ranking differs from MIC in that the label for each bag is not known. Instead, the MI ranking algorithms are provided preference information between pairs of bags. MIRank considers the partial ranking problem, which inherently exhibits three levels of structure: items (instances) belong to bags and bags belong to boxes. The objective is to learn a ranking function that can identify the preferred bag in each box. A concrete example where this framework can be applied is in learning positive target concepts. For a set of video frames (boxes), one may want to predict the smaller image chips (bags) that have the highest probability of containing a target object (positive instances). MIRank uses a linear prediction function to rank instances in individual bags. The ranking of instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in bags I and J is guided by the preference information between I and J. Work by Hu et al. (2008) introduced *multiple-instance ranking* based on a max-margin framework. In this setting, images were represented by sets of regions and the goal was to rank images according to relevance to a keyword. Assuming the preference relationship that $\boldsymbol{x}_m$ is preferable to $\boldsymbol{x}_n$ is denoted by $\boldsymbol{x}_m \succ \boldsymbol{x}_n$, the goal is to induce a ranking function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ that fulfills the set of constraints

$$\forall \boldsymbol{x}_m \succ \boldsymbol{x}_n : f(\boldsymbol{x}_m) > f(\boldsymbol{x}_n) \tag{2-9}$$

The value of $f(\boldsymbol{x}_n)$ is referred to as the ranking score of $\boldsymbol{x}_n$ and is typically a linear function $f(\boldsymbol{x}_n) = \langle \boldsymbol{w}, \boldsymbol{x}_n \rangle = \boldsymbol{w}^T \boldsymbol{x}_n$. Adding slack variable ζ_{mn}, the optimization problem can be solved

with with the following objective:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\zeta}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + \gamma \sum_{m,n} \zeta_{mn} \\ \text{s.t.} \quad & \forall \boldsymbol{x}_m \succ \boldsymbol{x}_n : \langle \boldsymbol{w}, \boldsymbol{x}_m \rangle \geq \langle \boldsymbol{w}, \boldsymbol{x}_n \rangle + 1 - \zeta_{mn} \\ & \forall m, n : \zeta_{mn} \geq 0 \end{aligned} \tag{2-10}$$

which is referred to as a *ranking SVM*. Assuming the optimal solution is $\boldsymbol{w}^*$, the ranking score of a test point $\boldsymbol{x}'$ is given as $f(\boldsymbol{x}') = \langle \boldsymbol{w}^*, \boldsymbol{x}' \rangle$. This framework assumes each sample $\boldsymbol{x}_n$ is a single instance. In multiple-instance ranking, we are given a set of preference relations between bag pairs and it is assumed that the score of a bag $\boldsymbol{B}_k$ is determined by the scores of the instances it contains

$$h(\boldsymbol{B}_k) = h\left(\{f(\boldsymbol{x}_n)\}_{n=1}^{N_k}\right) \tag{2-11}$$

Under this formulation, the objective can be re-written to consider bag scores as

$$\begin{aligned} \min_{f \in \mathcal{H},\boldsymbol{\zeta}} \quad & \frac{1}{2}||f||_{\mathcal{H}}^2 + \gamma \sum_{m,n} \zeta_{mn} \\ \text{s.t.} \quad & \forall \boldsymbol{B}_m \succ \boldsymbol{B}_n : h(\boldsymbol{B}_m) \geq h(\boldsymbol{B}_n) + 1 - \zeta_{mn} \\ & \forall m, n : \zeta_{mn} \geq 0 \end{aligned} \tag{2-12}$$

A bag's score is determined by the scores of its instances. Different functions for providing a bag score have been investigated, including using the max of instance scores, the mean of instance scores and the softmax of instance scores.

Besides MIRank and multiple-instance ranking, Asif et al. (2017) recently proposed *pyLEMMINGS*, which implements locally linear MI ranking by learning a large margin discriminant function from bags with corresponding integer rankings. While ranking has been successfully applied to text information retrieval, image retrieval (Hu et al., 2008) and bioinformatics (Asif et al., 2017), ranking under the MIL framework is still a relatively unexplored area of research.

2.2 Manifold Learning

Real-world remote sensing data such as hyperspectral imagery, ground-penetrating radar scans and sonar signals are naturally represented by high-dimensional feature vectors. However, in order to handle such real-world data adequately, its dimensionality usually needs to be reduced (van der Maaten et al., 2007; Belkin and Niyogi, 2004). The problem considered in this work is discovering feature representations that promote class discriminability for target or anomaly detection. This is typically achieved in one of two ways. First, features can be projected into a high-dimensional space (such as a Kernel Hilbert Space) using a kernel function. The second option, which is the focus of this work, is to transform the data into a new (often lower-dimensional) coordinate system which optimizes feature representations for discrimination (Vural and Guillemot, 2018).

The application of *dimensionality reduction* (DR) has proven useful in myriad applications in the literature, such as: visualization of high-dimensional data, classification, redundancy removal, compression and data management, improving computational tractability and efficiency, and reducing the effects of the Curse of Dimensionality (Bishop et al., 1998; Nickel and Kiela, 2017; Talmon et al., 2015; Tenenbaum et al., 2000; X. Geng et al., 2005; Palomo and Lopez-Rubio, 2017; Kohonen, 1990; Kegl et al., 2008; Bengio et al., 2012). In classification of object entities, it is often assumed that classes can be described by an *intrinsic* subset of representative features which demonstrate geometrical structure (Belkin et al., 2006). These structures are called intrinsic *manifolds*, and they represent the generating distributions of class objects exactly by the number of degrees of freedom in a dataset (Thorstensen, 2009; Belkin and Niyogi, 2004). Consider the example shown in Figure 2-3. This classic example demonstrated in (Thorstensen, 2009) shows samples from a pose-estimation dataset **CITE**. While each individual image is represented by a vector of features (pixel intensities in this case) in $\mathbb{R}^{4096}$, the dataset only exhibits three degrees of freedom: 1 light variation parameter and 2 rotation angles. Thus, it is intuitive that the dataset lies on a smooth, intrinsic submanifold spanning three dimensions which inherently capture the degrees of freedom in the data.

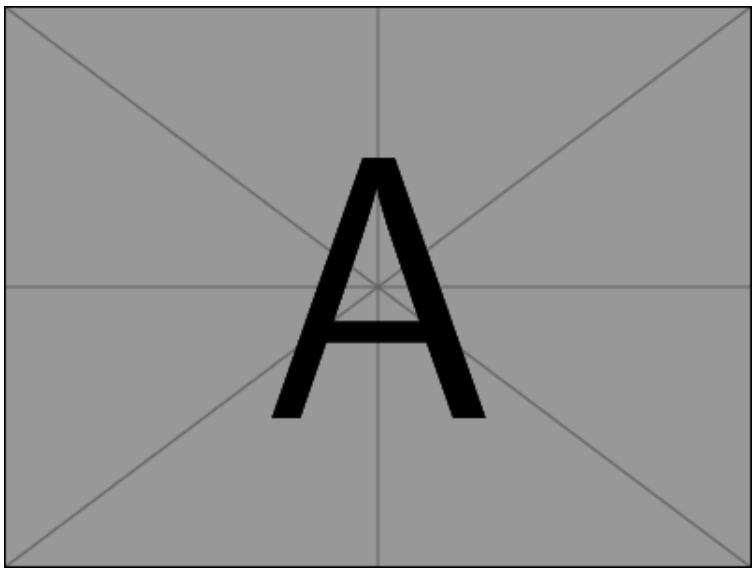


Figure 2-3: Placeholder for example of high D data lying on a low-dimensional sub-manifold.

The goal of manifold Learning is then to discover embedding functions which take data from the input feature space and transform it into a lower-dimensional (ideally intrinsic) coordinate system (also called a *latent space* in the literature) which captures the "useful" properties of the data, while enforcing constraints such as smoothness (the transformation function should not produce sporadic images), continuity (no discontinuous points on the hyper-surface), topological ordering (neighbors in the input space should also be neighbors in the embedded space) or class separability (samples from the same class should fall metrically close to each other in the embedded space and disparate classes should be distinctly far) (Vural and Guillemot, 2018).

This dissertation focuses on investigating the use of manifold learning to increase instance discriminability in the latent space, where labels are solely provided at the bag-level. While there is an expansive literature in unsupervised manifold learning methods, this document will pay special attention to both strictly- and semi-supervised methods, since they are typically adaptations of unsupervised approaches, as well as manifold learning under the MIL framework.

2.2.1 Definition and General Notation

Most studies perform classification or regression after applying unsupervised dimensionality reduction. However, it has been shown that there are advantages of learning the low-dimensional representations and classification/regression models simultaneously (Chao et al., 2019; Rish et al., 2008). Considering classification as the main goal of dimensionality reduction, this section provides a summary of the current literature in the area.

Given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{N \times D}$ where N is the total number of samples and D is the dimensionality of the input feature space, general dimensionality reduction seeks to find a representation $\boldsymbol{Z} \in \mathbb{R}^{N \times d}$ with $d \ll D$ that enhances the between-class separation while preserving the intrinsic geometric structure of the data(Vural and Guillemot, 2018). In other words, it is assumed that the data lie on a smooth manifold $\mathcal{X}$, which is the image of some parameter domain $\mathcal{Z} \subset \mathbb{R}^d$ under a smooth mapping $\Psi : \mathcal{Z} \to \mathbb{R}^D$. The goal of manifold learning is to discover an inverse mapping to the low-dimensional pre-image coordinates $\boldsymbol{z}_n \in \boldsymbol{z}$ corresponding to points $\boldsymbol{x}_n \in \boldsymbol{X}$. The data matrices $\boldsymbol{X} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T]$ and $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_N^T]$ are of size $N \times D$ and $N \times d$, respectively. Since these low-dimensional data representations are unknown, they are often referred to as *latent* vectors and the span in $\mathbb{R}^d$ is sometimes called the *latent feature space* or *latent space* for brevity (Murphy, 2012). The primary difference between traditional, unsupervised manifold learning and supervised approaches is that, in supervised manifold learning, data matrix $\boldsymbol{X}$ is accompanied with a corresponding label vector $\boldsymbol{l} = [l_1, \ldots, l_N]$ indicating the corresponding class labels of each sample in $\boldsymbol{X}$.

Manifold learning methods can be subdivided into a wide taxonomy of approaches, with *linear* and *nonlinear* at the root. Nonlinear approaches can be further divided into purely global methods and approaches that capture global structure solely from local information. We begin with a review of popular linear manifold learning techniques before moving into the realm of nonlinear approaches. Base, unsupervised methods are reviewed along with corresponding supervised and semi-supervised adaptations.

2.2.2 Linear Manifold Learning

A review of linear manifold learning approaches is provided. Linear approaches are advantageous over nonlinear because they allow for out-of-sample extensions. In other words, linear transformation matrices are learned which can be easily applied on data not included in the training set. However, linear approaches are limited in their abilities to capture irregular data surfaces (Kegl et al., 2008). Principal Component Analysis (PCA), Multi-dimensional Scaling (MDS), Nonnegative Matrix Factorization (NMF) and Fisher's Linear Discriminant Analysis (LDA) are reviewed. General approaches are discussed and supervised as well as nonlinear extensions are elaborated. Special focus is given to (LDA), as it is the only inherently supervised technique out of the included approaches.

2.2.2.1 Principal Component Analysis (PCA)

Unsupervised PCA. Principal Component Analysis (PCA) is arguably the most popular (and best-studied) technique for dimensionality reduction and manifold learning. It attempts to learn an orthogonal projection of the input data into a lower-dimensional space, known as the principal subspace, such that the variance of the projected data is maximized (Chao et al., 2019). In other words, each *principal axis*, or *principal component*, of the learned coordinate system is orthogonal to the other principal components. In summary, the problem of PCA is to discover basis vectors which linearly combine to reconstruct the data. In practice, data in the input feature space are projected into a new coordinate system of d dimensions, such that the variance along each principal axis is maximized and the reconstruction errors of the data are minimized in the mean-square sense (Thorstensen, 2009). Let V be a d-dimensional subspace of $\mathbb{R}^D$ and let $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_D$ be an orthonormal basis of $\mathbb{R}^D$ such that $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d$ is a basis of V. The goal of PCA is to find an orthogonal set of basis vectors $\boldsymbol{w}_n \in \mathbb{R}^D$ and corresponding latent coordinates $\boldsymbol{z}_n \in \mathbb{R}^d$ such that the average reconstruction error is minimized (Murphy, 2012)

$$J(\boldsymbol{W}, \boldsymbol{Z}) = \frac{1}{N} \sum_{n=1}^{N} ||\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n||^2 \tag{2-13}$$

where $\hat{\boldsymbol{x}}_n = \boldsymbol{W}\boldsymbol{z}_n$, subject to the constraint that $\boldsymbol{W}$ is *orthonormal*, or that $\boldsymbol{w}_i^T\boldsymbol{w}_j = 0, \forall i \neq j$ and $\boldsymbol{w}_i^T\boldsymbol{w}_i = 1$. This is equivalently written as

$$J(\boldsymbol{W}, \boldsymbol{Z}) = ||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z}||_F^2 \tag{2-14}$$

where $\boldsymbol{Z}$ is a $N \times d$ matrix with the $\boldsymbol{z}_n$ in its rows and $||\boldsymbol{A}||_F$ is the *Frobenius norm* of matrix $\boldsymbol{A}$, defined by

$$||\boldsymbol{A}||_F = \sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} a_{mn}^2} = \sqrt{tr(\boldsymbol{A}^T\boldsymbol{A})} = ||\boldsymbol{A}(:)||_2 \tag{2-15}$$

As noted by Murphy (Murphy, 2012), the optimal solution is obtained by setting $\hat{\boldsymbol{W}} = \boldsymbol{U}_d$, where $\boldsymbol{U}_d$ contains the eigenvectors corresponding to the d largest eigenvalues of the mean-subtracted, empirical data covariance matrix, $\hat{\boldsymbol{S}} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}})^T$, where $\hat{\boldsymbol{\mu}}$ is the empirical data mean. Therefore, the low-dimensional encoding of the data is given by $\boldsymbol{z}_n = \hat{\boldsymbol{W}}^T\boldsymbol{x}_n$, which is the orthogonal, linear projection of the data onto the column space spanned by the eigenvectors of the d largest eigenvalues of the empirical data covariance.

The example shown in figure 2-4 demonstrates the projection of 2-dimensional data onto the first principal axis. As can be seen from the figure, the first principal axis corresponds to the direction of maximal variance of the data. PCA from the viewpoint of variance maximization is often called the *analysis view* of PCA (Murphy, 2012).

PCA has been successfully applied to a large number of domains such as face recognition, coin classification and seismic series analysis (van der Maaten et al., 2007). It's success is partially because of its convenience of use. Embedding out-of-sample points with PCA is simple since the transformation is just a rotation and scaling. However PCA suffers from a few drawbacks. First, the dimensionality of the covariance matrix is proportional to the dimensionality of the data points. As a result, the computation of the eigenvectors may be infeasible or untrustworthy (singular) for high-dimensional data. Additionally, PCA focuses mainly on preserving large pairwise distances between data samples instead of retaining local

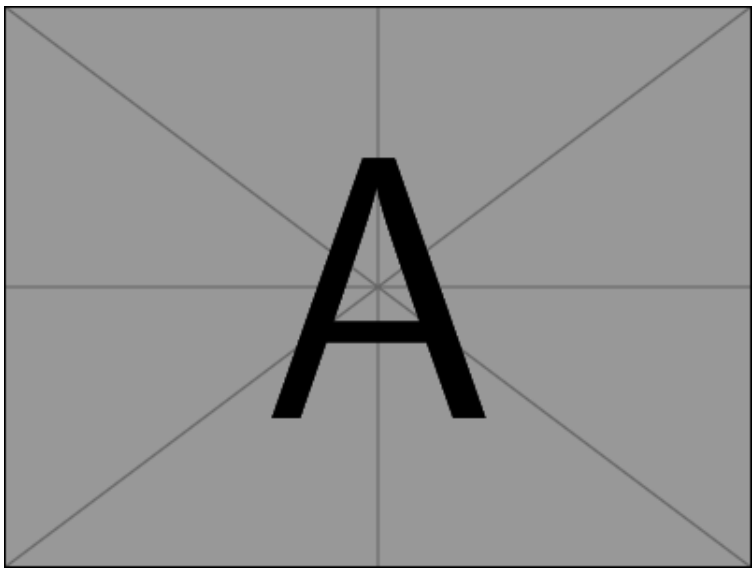


Figure 2-4: Placeholder for PCA projection example.

relationships, which may be important in certain applications. PCA also assumes Gaussian distributed data, which is unlikely in real-world applications. Finally, PCA is sensitive to feature magnitude. It is typical to standardized data before applying PCA as it can be misled by directions in which the variance is high simply because of the measurement scale.

Many extensions have been made to PCA, including the development of nonlinear versions (Sections 2.2.3.2and 2.2.4.1) (Schölkopf et al., 1999; Scholz et al., 2008) and formulating it as a factor analysis problem (Section **??**) (Tipping and Bishop, 1999).

Independent and Canoncial Component Analyses. It is also worth mentioning two popular approaches closely related to PCA, namely, *Independent Component Analysis* (ICA) and *Canonical Component Analysis* (CCA). ICA attempts to solve the *blind-source separation problem*, in which the goal is to deconvolve mixed signals into their constituent parts (Murphy, 2012; Hyvarinen and Oja, 2000; Tharwat, 2018). Instead of discovering the directions of maximum variance as done with PCA, ICA attempts to uncover the directions such that the data projected onto these directions have maximum statistical independence. On the other hand, CCA jointly considers multiple variables (multiple feature spaces) and tries to discover the correlations between them. CCA looks for directions in each feature space such that the

data projected onto the direction found in each space has the maximum possible correlation (Bach and Jordan, 2005). Thus, CCA can be used to simultaneously reduce the dimensionality of data in multiple feature spaces (Murphy, 2012).

2.2.2.2 Multi-Dimensional Scaling (MDS)

Unsupervised MDS. Most modern manifold learners have theoretical and algorithmic roots in one of three basic dimensionality reduction techniques: PCA, K-means and *Multidimensional Scaling* (MDS). Whereas PCA looks for linear projection bases which are constructed from the eigenvectors of a data covariance or scatter matrix, MDS tries to find a linear projection that preserves pairwise distances as well as possible. This idea is demonstrated by Figure 2-5, where the pairwise distances between samples in the 3-dimensional space are preserved in the 2-dimensional embedding space. While MDS does not construct an embedded manifold explicitly, it holds the status of being the grandfather of "one-shot" (non-iteratve) manifold learners, such as Isomap and Locally Linear Embedding (LLE), which are discussed later in this literature review (Kegl et al., 2008).

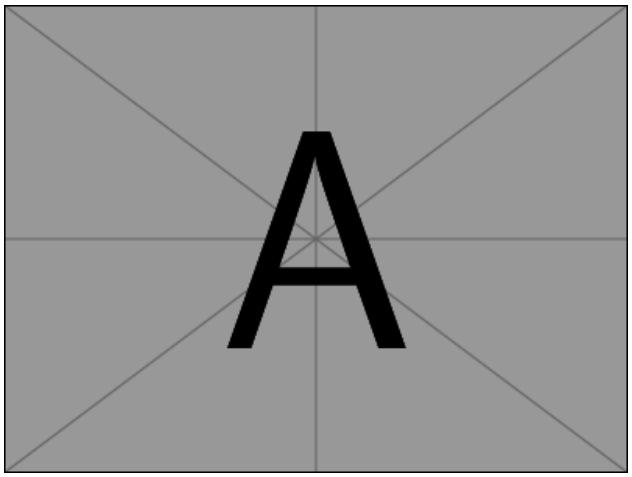


Figure 2-5: Example of MDS distance preservation.

The steps of MDS correspond exactly to those of PCA except that, instead of a scatter matrix $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$, MDS operates with a positive semi-definite, dissimilarity matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$, where N is the number of data samples and a real, symmetric Gram matrix

$\boldsymbol{K} = \boldsymbol{X}^T\boldsymbol{X}$ (inner-product matrix) where $\boldsymbol{K}_{mn}$ is the inner product between $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$. The problem of MDS is posed as finding d-dimensional Euclidean coordinates for each sample $\boldsymbol{x}_n$ in dataset $\boldsymbol{X}$ such that the Euclidean distances in the low-dimensional embedding space are proportional to the pairwise distances in the input space (Thorstensen, 2009; Sorzano et al., 2014). While the literature poses several cost functions for this task, this review focuses on classical MDS, which is described as follows:

First, the pairwise distance matrix $\boldsymbol{D}$ is computed such that

$$\boldsymbol{D}_{mn} = \mathcal{D}_{\mathcal{X}}(\boldsymbol{x}_m, \boldsymbol{x}_n) = ||\boldsymbol{x}_m - \boldsymbol{x}_n||^2 = (\boldsymbol{x}_m - \boldsymbol{x}_n)^T(\boldsymbol{x}_m - \boldsymbol{x}_n) \tag{2-16}$$

where $\mathcal{D}_{\mathcal{X}}(\cdot,\cdot)$ is a chosen dissimilarity metric (Euclidean distance for classical MDS). Then, the double-centered Gram matrix $\boldsymbol{K}$ is computed by

$$\boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H} \tag{2-17}$$

where

$$\boldsymbol{H} = \mathbb{I}_N - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T \tag{2-18}$$

with $\mathbb{I}_N$ denoting the $N \times N$ identity matrix and $\boldsymbol{e} = (1, \ldots, 1)^T$ the $N \times 1$ column vector of all ones. Multiplying $\boldsymbol{D}$ on both sides by $\boldsymbol{H}$ performs *double centering*, which subtracts the row and column means from $\boldsymbol{D}$ (and adds back the global mean which gets subtracted twice), so that both the row and column means of $\boldsymbol{K}$ are equal to zero. The objective is to find $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\} \in \mathbb{R}^d$ which minimizes the objective

$$J(\boldsymbol{D}_{\boldsymbol{X}}, \boldsymbol{D}_{\boldsymbol{Z}}) = ||\boldsymbol{K}_{\boldsymbol{X}} - \boldsymbol{D}_{\boldsymbol{Z}}||^2 = \left|\left|-\frac{1}{2}\boldsymbol{H}(\boldsymbol{D}_{\boldsymbol{X}} - \boldsymbol{D}_{\boldsymbol{Z}})\boldsymbol{H}\right|\right|^2 \tag{2-19}$$

Similarly to PCA, this can be solved by a generalized eigenvalue problem

$$\boldsymbol{K}v = \lambda v \tag{2-20}$$

such that

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}} \tag{2-21}$$

with $\boldsymbol{\Lambda}^{\frac{1}{2}} = diag(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_N})$ being a diagonal matrix with entries equal to the square roots of the eigenvalues of $\boldsymbol{K}$ sorted from largest to smallest ($\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$), and $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ the corresponding eigenvectors. The low-dimensional embedding coordinates $\boldsymbol{Z} \in \mathbb{R}^{N \times d}$ are obtained by $\boldsymbol{Z} = \{\sqrt{\lambda_1}\boldsymbol{v_1}, \ldots, \sqrt{\lambda_d}\boldsymbol{v_d}\}$ (Chao et al., 2019).

It has been proven that the eigenvalues of Gram matrix $\boldsymbol{K}$ and covariance $\boldsymbol{S}$ are the same, and that the space spanned by MDS and PCA are the identical for any $d \leq rank(\boldsymbol{K}) = rank(\boldsymbol{S})$. This implies that a rotation matrix $\boldsymbol{A}$ could be found such that $\boldsymbol{A}^T\boldsymbol{Z}_{MDS} = \boldsymbol{Z}_{PCA}$ (Sorzano et al., 2014). Additionally, MDS can be computed even if the data observation matrix $\boldsymbol{X}$ is unknown. All that is needed is the Gram matrix or a dissimilarity matrix. This feature potentially allows MDS to be applied in a variety of data-sensitive and privacy-concerned scenarios.

A pitfall of MDS is that it focuses on retaining global pairwise distances as opposed to local distances, which are typically much more important for capturing the geometry of the data (van der Maaten et al., 2007). Several MDS variants have been proposed to address this weakness. A popular variant is known as *Sammon Mapping* and is discussed in Section 2.2.3.5.

Supervised MDS. As with most manifold learning methods in the literature, MDS does not inherently consider class information when learning the embedding function. In attempt to promote class separability in the low-dimensional embedding space, Witten et al. proposed a *Supervised Multidimensional Scaling* (SMDS) (Witten and Tibshirani, 2011). this method follows the idea of traditional MDS where the goal is to find low-dimensional coordinate or *configuration points* $\boldsymbol{z}_n \in \mathbb{R}^d$, such that pairwise distances in the input feature space are preserved in the embedding space. Incorporating class label information, the goal of SMDS is to not only preserve distances, but ensure the coordinate values $z_{mk} > z_{nk}$ when $l_m > l_n, \quad \forall k = 1, \ldots, d$, where l are the instance-level labels and d is the dimensionality

of the embedding space. Considering the binary target classification case, SMDS can be formulated as

$$\min_{\boldsymbol{Z}} \quad \frac{1}{2}(1-\alpha)\sum_{m=1}^{N}\sum_{n=1}^{N}(\boldsymbol{D}_{mn}-||\boldsymbol{z}_m-\boldsymbol{z}_n||^2)+\alpha\sum_{m:l_m=1}\sum_{n:l_n=2}\sum_{k=1}^{d}\left(\frac{\boldsymbol{D_{mn}}}{\sqrt{d}}-(z_{nk}-z_{mk})^2\right) \tag{2-22}$$

This objective has two terms. The first is the traditional metric MDS *stress*. This term attempts to ensure that the Euclidean distances of two points in the embedding space is the same as the dissimilarity between the points in the input feature space. The second term is the supervised term which enforces that each dimension of the embedded configuration points be larger if belonging to the class with the larger label, and smaller if belonging to the class with a smaller-valued label. The term $\alpha \in [0,1]$ is a tuning parameter. When $\alpha = 0$, the objective reduces to the MDS stress function. As α increases, however, the objective becomes increasingly more supervised, focused on ensuring class separation of the training data.

A least square regression was applied to estimate the embedding function for out-of-sample test points. SMDS was successfully applied to tasks in data visualization, bipartite ranking and classification of prostate data and USPS handwritten digits.

2.2.2.3 Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is a tool for linear dimensionality reduction that, given a set of data $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, aims to decompose the matrix into a coefficient matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times d}$ and basis matrix $\boldsymbol{W} \in \mathbb{R}^{d\times D}$ (Gillis, 2014). In this case, the columns of matrix $\boldsymbol{W}$ are basis elements and the columns of matrix $\boldsymbol{Z}$ give the coordinates of data samples in the basis $\boldsymbol{W}$. Similar to PCA, the goal of NMF is to find a $\boldsymbol{Z}$ and $\boldsymbol{W}$ such that $\hat{\boldsymbol{X}} = \boldsymbol{ZW}$ approximates the data as close as possible, given that every element must be nonnegative. This notion is formalized by

$$\min_{\boldsymbol{Z},\boldsymbol{W}} ||\boldsymbol{X}-\boldsymbol{ZW}||_F^2 \quad s.t. \quad \boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0 \tag{2-23}$$

Another popular variant of NMF is to substitute the Frobenius norm for the Kullback-Leibler (KL) divergence, where $D(\boldsymbol{X}||\hat{\boldsymbol{X}}) = \sum_{m,n}(\boldsymbol{X}_{mn}\log\frac{\boldsymbol{X}_{mn}}{\hat{\boldsymbol{X}}_{mn}} + \boldsymbol{X}_{mn} - \hat{\boldsymbol{X}}_{mn})$. Given that $d \ll D$, it is intuitive that $\boldsymbol{Z} \in \mathbb{R}^{N\times d}$ provides the desired low-dimensional representations of high-dimensional data $\boldsymbol{X}$. Many approaches have been used to solve for the nonnegative matrices, including alternating least squares, projected gradient descent, coordinate descent and the Alternating Direction Method of Multipliers (ADMM) (Chao et al., 2019). NMF has been successfully applied to applications in image processing, text mining, hyperspectral imaging, air emission control, computational biology, blind source separation, single-channel source separation, clustering, music analysis and collaborative filtering (Gillis, 2014).

As described in (Chao et al., 2019), two groups of supervised NMF have been proposed in the literature according to the way label information is utilized. In *direct supervised NMF* approaches, label information is incorporated directly into the loss function to promote learning of well-separated coordinate representations for samples in different classes. Approaches taken under this framework include incorporating simple indicator variables to denote the class of a sample, integrating NMF with a SVM and formulating the problem under task-driven dictionary learning. An alternative approach to the direct supervised NMF is *discriminative NMF*. Discriminative NMF approaches are based on *Linear Discriminant Analysis* (LDA). Essentially, the objective for this class of algorithm is to decompose the data matrix such that the between-class distances of the low-dimensional coordinates $\boldsymbol{Z}$ are maximized, while the within-class distances are minimized. Supervised NMF approaches have been applied successfully to problems in acoustic separation, brain tumor detection and emotion classification.

2.2.2.4 Fisher's Linear Discriminant Analysis (LDA)

Classical LDA. *Linear Discriminant Analysis* (LDA) is a popular method for supervised, linear dimensionality reduction. LDA currently forms the basis for Multiple Instance Learning dimensionality reduction methods exhibited in the literature (Sun et al., 2010; Chai et al., 2014; Zhu et al., 2018; Xu et al.). Whereas PCA tries to project data into a space which maximizes variance, LDA considers class label information and tries to find a transformation

which both maximizes between-class (inter-class) dissimilarity and minimizes within-class (intra-class) scatter (Yan et al., 2007; Chao et al., 2019; Sun et al., 2010; Murphy, 2012). This is done by maximizing the ratio between the inter-class $\boldsymbol{S}_b$ and intra-class $\boldsymbol{S}_w$ scatter matrices, defined as:

$$\boldsymbol{S}_w = \sum_{k=1}^{K} \boldsymbol{S}_k \tag{2-24}$$

$$\boldsymbol{S}_k = \sum_{n \in C_k} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_k)^T \tag{2-25}$$

$$\boldsymbol{S}_b = \sum_{k=1}^{K} N_k (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})^T \tag{2-26}$$

Here, $\boldsymbol{S}_w$ is the global within-class scatter matrix which is defined as the sum over each individual class' scatter matrix $\boldsymbol{S}_k$, and $\boldsymbol{S}_k$ is essentially an outer product between all samples belonging to class C_k after subtracting the respective empirical class mean $\hat{\boldsymbol{\mu}}_k$. This scatter matrix would be the class covariance if it was normalized by the number of samples N_k in class C_k. However, this normalization constant does not affect the final solution and can thus be ignored. The between-class scatter $\boldsymbol{S}_b$ is defined by the sum of outer products of the differences between the empirical class means $\hat{\boldsymbol{\mu}}_k$ and the global data mean $\hat{\boldsymbol{\mu}}$, weighted by the number of samples in each class. The objective of LDA is then to solve for $\boldsymbol{W}^*$ which maximizes the ratio $J(\boldsymbol{W})$:

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} J(\boldsymbol{W}) = \arg\max_{\boldsymbol{W}} \frac{|\boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}|}{|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}|} \tag{2-27}$$

It has been shown that the optimal projection matrix $\boldsymbol{W}^*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the generalized eigenvalue problem

$$\boldsymbol{S}_b \boldsymbol{w} = \lambda \boldsymbol{S}_w \boldsymbol{w} \Rightarrow \boldsymbol{S}_w^{-1} \boldsymbol{S}_b \boldsymbol{w} = \lambda \boldsymbol{w} \tag{2-28}$$

Since $\boldsymbol{S}_b$ is the sum of K matrices of rank ≤ 1, this implies that $\boldsymbol{S}_b$ will be of rank $(K-1)$ or less and only $(K-1)$ of the eigenvalues λ will be non-zero. A low-dimensional coordinate representation $\boldsymbol{z}_n \in \mathbb{R}^{(K-1)}$ of sample $\boldsymbol{x}_n \in \mathbb{R}^D$ is given by the linear projection of $\boldsymbol{x}_n$ onto the hyper-plane parameterized by $\boldsymbol{W}^*$, $\boldsymbol{z}_n = \boldsymbol{W}^{*T}\boldsymbol{x}_n$ It should be noted that for LDA, the dimensionality of the latent space is not a free-parameter, but is always fixed at $d = (K-1)$, or one less than the number of classes present in the dataset. Equivalently, LDA can be derived by maximum likelihood for normal class-conditional densities where the covariances for each class are assumed to be equivalent (Murphy, 2012). For the special case of binary target classification, the LDA transformation will place every sample onto a single line in 1-dimension, and thus the LDA solution can be simplified:

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}} = \boldsymbol{S}_w^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \tag{2-29}$$

where $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\mu}}_2$ are the empirical means for classes 1 and 2, respectively. Figure 2-6 demonstrates the differences between PCA and LDA. While PCA projects data onto the axes exhibiting the maximal variation, LDA projects the data into a space which attempts to simultaneously enforce between-class separation and within-class compactness.

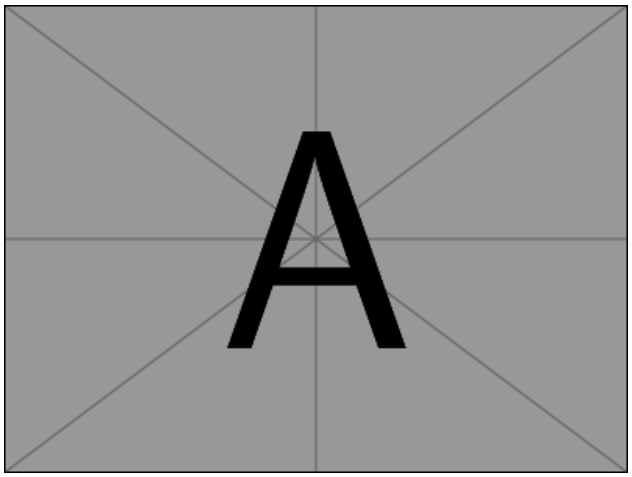


Figure 2-6: LDA example.

Although LDA is the basis for large number of discriminative dimensionality reduction approaches, it does not guarantee class separation in the embedding space. For example, LDA projects data into a space of at most $(K-1)$ dimensions, however, more features may be necessary for adequate class discrimination. Additionally, LDA is a parametric method which assumes unimodal Gaussian likelihoods. This implies that it may not be able to preserve complex data structure. Finally, LDA will fail if the discriminatory information is contained in the variance of the data instead of the mean. Despite these pitfalls, LDA has been successfully applied to object detection and recognition tasks (Wang et al., 2016) Many variations of LDA have been developed, such as Non-parametric LDA (Fukunaga and Mantock, 1983), Orthonormal LDA (Wang et al., 2016), Generalized LDA (Baudat and Anouar, 2000) and Multilayer Perceptrons (Webb and Lowe, 1990). Additionally, LDA serves as the foundation for many of the Multiple Instance Learning dimensionality reduction approaches in the current literature (Sun et al., 2010; Chai et al., 2014; Zhu et al., 2018).

2.2.3 Nonlinear Manifold Learning

Linear methods such as PCA and MDS are convenient for projecting out-of-sample test points into the embedding space. However, they are unable to capture the structure of data that are sampled from nonlinear manifolds (Kegl et al., 2008). This section will discuss a variety of nonlinear dimensionality reduction and manifold learning approaches. All methods reviewed assume the data is distributed along a d-dimensional sub-manifold $\mathcal{X}$ embedded in $\mathbb{R}^D$.

2.2.3.1 Kernelization

Although each of the manifold learning techniques previously discussed are inherently linear, nonlinear adaptations have been made. One easily extendable approach is to utilize kernel functions as means to provide nonlinearity in the embeddings.

Kernels. A *kernel function* is a real-valued function of two arguments $\kappa(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}$, for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, which maps vectors from the input feature space to a single value in $\mathbb{R}$. The

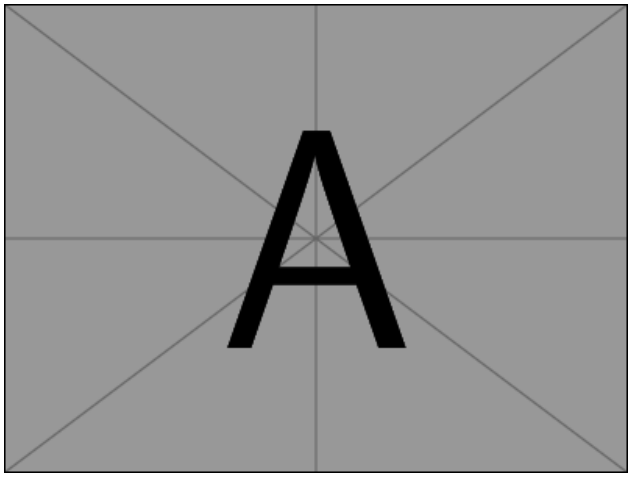


Figure 2-7: Example of a nonlinear manifold.

function is typically symmetric (i.e. $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}', \boldsymbol{x})$) and non-negative (i.e. $\kappa(\boldsymbol{x}, \boldsymbol{x}') \geq 0$), which implies that it can be interpreted as a measure of similarity (Murphy, 2012). The notion of kernels is very useful in certain applications where data representation is not straightforward, such as representing text documents or molecular structures which can have variable length. Additionally, this allows algorithms to operate directly on the kernel representations. This is useful in data-sensitive scenarios where direct access to the data may not be available.

A popular choice of kernel in manifold learning is the *radial basis function* (RBF) kernel, defined as:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{\beta}\right) \tag{2-30}$$

where β is the bandwidth of the isotropic function. Another popular kernel for text classification is *cosine similarity*, defined by:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \frac{\boldsymbol{x}^T \boldsymbol{x}'}{||\boldsymbol{x}||_2 ||\boldsymbol{x}'||_2} \tag{2-31}$$

This kernel measures the cosine of the angle between vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ after scaling them onto the unit hyper-sphere. If $\boldsymbol{x}$ and $\boldsymbol{x}'$ are strictly positive vectors (counts in the bag-of-words model, for example), then the kernel provides values in $[0, 1]$, where a value of 0 means the

feature vectors are orthogonal and, therefore, have no features in common, and a value of 1 means the vectors are the same.

Some of the nonlinear manifold learning methods in the literature require the kernel function to satisfy the requirement that the *Gram matrix*

$$\boldsymbol{K} = \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ & \vdots & \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \tag{2-32}$$

be positive definite for any set of inputs $\{\boldsymbol{x}_n\}_{n=1}^N$. This type of kernel is called a *Mercer kernel* or *positive definite kernel*. The importance of the Mercer Kernel is the following result, known as *Mercer's theorem*. This theorem states that if the Gram matrix is positive definite, its eigenvector decomposition can be written as

$$\boldsymbol{K} = \boldsymbol{U}^T \boldsymbol{\Lambda} \boldsymbol{U} \tag{2-33}$$

As derived by Murphy and Liu et al. (Murphy, 2012; Liu et al., 2010), it then follows that each entry of $\boldsymbol{K}$ can be computed as

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}') =< \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') > \tag{2-34}$$

meaning that the entries of the kernel matrix can be defined by the inner product of some feature vectors that are implicitly defined by the eigenvectors $\boldsymbol{U}$. If the kernel is Mercer, then there exists a function ϕ which maps $\boldsymbol{x} \in \mathcal{X}$ to $\mathbb{R}^D$ such that Equation 2-34 holds. Additionally, ϕ depends on the eigenfunctions of κ, meaning that D is a potentially infinite dimensional space. Additionally, instead of representing feature vectors in terms of kernels $\phi(\boldsymbol{x}) = [\kappa(\boldsymbol{x}, \boldsymbol{x}_1), \dots, \kappa(\boldsymbol{x}, \boldsymbol{x}_N)]$, algorithms can instead work with the input feature vectors $\boldsymbol{x}$ by replacing all inner products $< \boldsymbol{x}, \boldsymbol{x}' >$ with a call to the kernel function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$. This is called the *kernel trick*, and it turns out that many algorithms can be kernelized in this way.

Kernel functions play an important role in the dimensionality reduction literature for both applying nonlinearity to inherently linear problems and in defining similarity measures for graph-based manifold learning methods. Specifically, nonlinear adaptations of the inherently-linear PCA (Schölkopf et al., 1999), MDS (Webb, 2002) and LDA (Ghojogh et al., 2019) algorithms have been formulated. The kernelization of PCA is briefly described in the following.

2.2.3.2 Kernel PCA (KPCA)

Section 2.2.2.1 showed how PCA could be used to compute linear low-dimensional embeddings of data. This process involved finding the eigenvectors of the empirical data covariance matrix $\hat{\boldsymbol{S}} = \frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{x}}_n \hat{\boldsymbol{x}}_n^T = \frac{1}{N}\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}$, where $\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}$ is the mean subtracted feature vector. However, PCA can also be computed by finding the eigenvectors of the inner product matrix $\boldsymbol{X}\boldsymbol{X}^T$ (Murphy, 2012; Wang, 2012). This interpretation allows the production of nonlinear embeddings by taking advantage of the kernel trick. This approach is known as *Kernel PCA* (KPCA) (Schölkopf et al., 1999). Assuming the data is mapped to a new feature space in $\mathbb{R}^M$ by a nonlinear transformation $\phi(\boldsymbol{x})$, PCA could be performed in the new feature space. However, this computation can be extremely costly and inefficient. Instead, kernel methods can be used to simply the computation. Following the derivation defined in (Wang, 2012), first assume that the data in the new feature space has zero mean:

$$\frac{1}{N}\sum_{n=1}^{N} \phi(\boldsymbol{x}_n) = 0 \tag{2-35}$$

The covariance matrix of the projected features is a $M \times M$ matrix

$$\boldsymbol{S}_\phi = \sum_{n=1}^{N} \phi(\boldsymbol{x}_n)\phi(\boldsymbol{x}_n)^T \tag{2-36}$$

The eigenvectors $\boldsymbol{V}$ and eigenvalues λ for $\boldsymbol{S}_\phi$ satisfy

$$\boldsymbol{S}_\phi \boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k, \quad \forall k = 1, \ldots, M \tag{2-37}$$

From Equations 2-36 and 2-37, we have

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\boldsymbol{x}_n)\phi(\boldsymbol{x}_n)^T\boldsymbol{v}_k = \lambda_k\boldsymbol{v}_k \tag{2-38}$$

which can be re-written as

$$\boldsymbol{v}_k = \sum_{n=1}^{N}\alpha_{kn}\phi(\boldsymbol{x}_n) \tag{2-39}$$

Substituting Equation 2-39 into Equation 2-38, gives

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\boldsymbol{x}_n)\phi(\boldsymbol{x}_n)^T\sum_{n=1}^{N}\alpha_{kn}\phi(\boldsymbol{x}_n) = \lambda_k\sum_{n=1}^{N}\alpha_{kn}\phi(\boldsymbol{x}_n) \tag{2-40}$$

A $N \times N$ matrix $\boldsymbol{K}$ can be defined by

$$\boldsymbol{K}_{mn} = \kappa(\boldsymbol{x}_m, \boldsymbol{x}_n) = \phi(\boldsymbol{x}_m)^T\phi(\boldsymbol{x}_n) \tag{2-41}$$

which simplifies the problem to

$$\boldsymbol{K}^2\boldsymbol{\alpha}_k = N\lambda_k\boldsymbol{K}\boldsymbol{\alpha}_k \tag{2-42}$$

where $\boldsymbol{\alpha}_k$ is a column vector with entries $\alpha_{k1}, \ldots, \alpha_{kN}$. Each $\boldsymbol{\alpha}_k$ can be found by solving the eigenvalue problem

$$\boldsymbol{K}\boldsymbol{\alpha}_k = N\lambda_k\boldsymbol{\alpha}_k \tag{2-43}$$

Then the projection of a test point $\boldsymbol{x}$ onto the k^{th} principal component can be found by

$$z_k(\boldsymbol{x}) = \phi(\boldsymbol{x})^T\boldsymbol{v}_k = \sum_{n=1}^{N}\alpha_{kn}\kappa(\boldsymbol{x}_n, \boldsymbol{x}) \tag{2-44}$$

This formulation assumes that the projected data has zero mean. However, this is not generally the case and the mean cannot simply be subtracted in the projected space (Murphy, 2012). Therefore, KPCA can be performed on the centered Gram matrix, defined by

$$\tilde{\boldsymbol{K}} = \boldsymbol{H}\boldsymbol{K}\boldsymbol{H} \tag{2-45}$$

where

$$\boldsymbol{H} = \mathbb{I}_N - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T \tag{2-46}$$

is the $N \times N$ centering matrix. Pseudo-code for KPCA is given in Algorithm 2.

Algorithm 2 KPCA

Input: Gram matrix of training data $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, Gram matrix augmented with test data $\boldsymbol{K}_* \in \mathbb{R}^{N_*\times N}$, dimensionality of latent space d
Output: Embedded data coordinates $\boldsymbol{Z} \in \mathbb{R}^{N\times d}$

$\boldsymbol{H} \leftarrow \mathbb{I}_N - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T$
$\tilde{\boldsymbol{K}} \leftarrow \boldsymbol{H}\boldsymbol{K}\boldsymbol{H}$
$[\boldsymbol{U}, \boldsymbol{\Lambda}] \leftarrow \text{EIG}(\tilde{\boldsymbol{K}})$
for $n \in N$ **do**
    $\boldsymbol{v}_n \leftarrow \boldsymbol{u}_n/\sqrt{\lambda_n}$
end for
$\boldsymbol{H}_* \leftarrow \frac{1}{N_*}\boldsymbol{e}_*\boldsymbol{e}^T$
$\tilde{\boldsymbol{K}}_* \leftarrow \boldsymbol{K}_* - \boldsymbol{H}_*\boldsymbol{K}_* - \boldsymbol{K}_*\boldsymbol{H}_* + \boldsymbol{H}_*\boldsymbol{K}_*\boldsymbol{H}_*$
$\boldsymbol{Z} \leftarrow \tilde{\boldsymbol{K}}_*\boldsymbol{V}_{1:d}$

Whereas linear PCA is limited to $d < D$ components, KPCA can use up to N components. Using a nonlinear feature embedding function along with the kernel trick provides for an elegant solution for capturing global nonlinear data structure. As can be seen in Algorithm 2, embedding out-of-sample test points is done by appending rows to the Gram matrix, where new entries are defined between the test points and training data points, but not between test points and other test points. Embedded data coordinates are computed by multiplying the augmented Gram matrix with the top d scaled eigenvectors of the training Gram matrix.

2.2.3.3 Graph-based Methods

Nonlinear manifold learning methods typically rely on the use of computational graphs. These graphs represent data structure pooled from local neighborhoods of samples. *Spectral graph theory* focuses on constructing, analyzing and manipulating graphs. It has proved useful for object representation, graph visualization, spectral clustering, dimensionality reduction and numerous other applications in chemistry, physics, signal processing and computer science (Shuman et al., 2013; Bengoetxea, 2002). An overview of computational graphs

as well as prominent methods for graph construction in manifold learning are presented. Additionally, geodesic distance approximation from pairwise distances is reviewed. It should be noted that the work in (Yan et al., 2007) shows how each of the undermentioned graph-based manifold learning approaches can be succinctly described under a general graph-based framework for dimensionality reduction. It is encouraged that readers turn to that work for additional information on the mathematical relationships between the algorithms, as well as how linearization, kernelization and tensorization are applied in the graph-based setting.

Terminology. Many dimensionality reduction methods in the literature are interested in analyzing relationships between samples defined on an undirected, weighted graph $G = \{\mathcal{V}, \mathcal{E}, \boldsymbol{W}\}$, which consists of a finite set of *vertices* $\mathcal{V}$ (also called *nodes* or *points*) with cardinality $\mathcal{V} = N$, a set of *edges* $\mathcal{E} \subset \mathcal{V} \times \mathcal{V} = [\mathcal{V}]^2$ (also known as *arcs* or *lines*) and a weighted *adjacency* or *affinity* matrix $\boldsymbol{W}$ (Shuman et al., 2013; Livi and Rizzi, 2013; Bengoetxea, 2002). The the size or *order* of a graph is defined by the number of nodes $|\mathcal{V}|$ and edges $|\mathcal{E}|$. If two vertices in G, say $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{V}$, are connected by an edge $e \in \mathcal{E}$, this is denoted by $e = (\boldsymbol{u}, \boldsymbol{v})$ and the two vertices are said to be *adjacent* or *neighbors*. When edges do not have a direction, they are coined as undirected. A graph solely containing this type of connection is termed as an *undirected graph*. When all edges have directions, meaning $(\boldsymbol{u}, \boldsymbol{v})$ and $(\boldsymbol{v}, \boldsymbol{u})$ are distinguishable, the graph is said to be *directed*. In the literature, the term *arc* is typically used to denote connections between nodes in directed graphs, while *edge* is used when they are undirected. The graph-based methods included in this literature review focus on analyzing affinities between data samples in undirected graphs. Moreover, a *path* between any two nodes in $\boldsymbol{u}, \boldsymbol{u'} \in \mathcal{V}$ is a non-empty sequence of k different vertices $< \boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k >$ where $\boldsymbol{u} = \boldsymbol{v}_0, \boldsymbol{u}' = \boldsymbol{v}_k$ and $(\boldsymbol{v}_{i-1}, \boldsymbol{v}_i) \in \mathcal{E}$, $i = 1, 2, \ldots, k$. Additionally, a graph is said to be *acyclic* if there are no cycles between its edges, regardless of whether it is directed or undirected.

When using graphs for dimensionality reduction, vertices usually represent features of individual samples, and edges express relationships between them. The most straight-forward

way to construct a graph is to instantiate edges between every vertex in the graph, where each edge is weighted by the distance between the vertices it connects according to a pre-defined metric. This type of graph is called *full mesh*. Weights on edges are captured in the graph adjacency matrix $\boldsymbol{W}$. When weights are not naturally defined by an application, a common way to the define the weight of an edge connecting vertices $\boldsymbol{u} \sim \boldsymbol{u}'$ is by a symmetric affinity function $W_{\boldsymbol{u},\boldsymbol{u}'} = K(\boldsymbol{u}; \boldsymbol{u}')$; typically a *radial basis function (RBF)* or *heat kernel*, defined as:

$$\boldsymbol{W}_{\boldsymbol{u},\boldsymbol{u}'} = w_{\boldsymbol{u},\boldsymbol{u}'} = \exp\left(-\frac{||\boldsymbol{u} - \boldsymbol{u}'||^2}{\beta}\right) \tag{2-47}$$

where β is the non-negative *bandwidth* of the kernel. Vertices will have a nonzero weight only if they fall within the nonzero mapping domain of the kernel. Additionally, a threshold could be set to truncate the weights of neighbors far from individual samples.

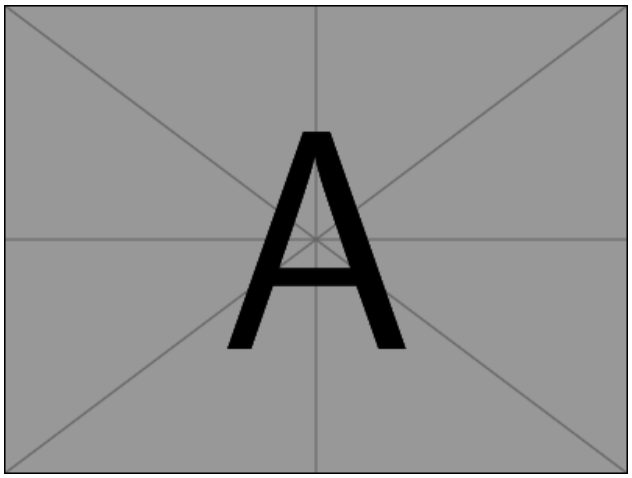


Figure 2-8: Examples of K-nearest neighbor and ϵ-ball graphs

K-Nearest Neighbor Graph. In a K-nearest neighbor graph, every data point (vertex) $\boldsymbol{x}_n \in \boldsymbol{X}$ is connected by edges to its K-nearest neighbors, where $K \in \mathbb{Z}^+$ is fixed. An example of a K-nearest neighbor graph is depicted in a of Figure 2-8. The downside of this graph is that it might impose edges between neighbors that should not actually be connected, as in the case where a sample is metrically distant from all of its nearest neighbors. Although, this feature may actually be useful in domains such as outlier detection, where low adjacency

weights indicate that the sample is far form the sampling distribution. Two alternative K-nearest neighbor graphs, a symmetric and mutual neighbors, might instead by utilized. In the symmetric K-nearest neighbors graph, two vertices $\boldsymbol{u}$ and $\boldsymbol{u}'$ if $\boldsymbol{u}$ is among the K-nearest neighbors of $\boldsymbol{u}'$ *or* $\boldsymbol{u}'$ is among the neighbors of $\boldsymbol{u}$. The mutual K-nearest neighbors graph, however, only connects vertices $(\boldsymbol{u}, \boldsymbol{u}')$ if $\boldsymbol{u}$ is among the K-nearest neighbors of $\boldsymbol{u}'$ *and* $\boldsymbol{u}'$ is among the K-nearest neighbors of $\boldsymbol{u}$. The weights on each edge are provided as the similarity of the adjacent nodes.

ϵ-Neighborhood Graph. Another method for graph construction is to use ϵ-neighborhoods (or ϵ-balls). In this graph, two vertices $(\boldsymbol{u}, \boldsymbol{u}')$ are connected by an edge if and only if the distance between them is equal to or smaller than some value ϵ, $\mathcal{D}_{\mathcal{U}}(\boldsymbol{u}, \boldsymbol{u}') \leq \epsilon$. This idea is represented in b of Figure 2-8. In both the K-nearest and ϵ-neighborhood graphs, a parameter controlling the number of edges in the graph, K or ϵ, must be chosen. These parameters are highly influential for graph construction and can thus greatly affect dimensionality reduction quality. Contrary to the K-nearest neighbor graph, an ϵ-neighborhood will not create connections between distant vertices. However, when the data is sampled sparsely from a highly-curved manifold, the ϵ-neighbor graph will not be able to appropriately capture the geometry (Thorstensen, 2009).

Geodesic Distance Approximation. The ultimate goal of manifold learning is to uncover an underlying low-dimensional sub-manifold which is embedded in $\mathbb{R}^D$. Many dimensionality reduction methods in the literature discover projections of data into a low-dimensional space which preserve topological ordering of the data (Kegl et al., 2008). These processes require a notion of distance between samples. *Euclidean distance* is a popular metric which captures the straight-line disparity between two points. As shown in Figure 2-9, however, samples that are actually distant on the manifold may appear deceptively close in the high-dimensional input feature space, as measured by Euclidean distance (Tenenbaum et al., 2000). *Geodesic distance*, also called *curvilinear* or *shortest-path distance*, Figure 2-9, on the other

hand, follows the curvature of a manifold and may provide a better measure of dissimilarity between data samples. Geodesic distance can be estimated by the shortest path through a graph constructed by assuming the distances between neighbors is locally Euclidean (Sorzano et al., 2014). This can be conceptualized by a simple example. The Earth is a sphere and naturally has curvature. Two people standing in a room, however, would estimate the distance between themselves by a straight line. Thus, in a very local region on the Earth, the measure of curvature would be negligible and the true distances between objects could be estimated with Euclidean distance. The same concept is true for manifolds where, if data is sampled densely enough, geodesic distance can be approximated by the shortest-path through a neighborhood graph where the dissimilarities between neighbors is assumed to be locally Euclidean. Geodesic distance can be estimated efficiently by methods such as Dijskstra's or Floyd's shortest-path algorithms (Tenenbaum et al., 2000).

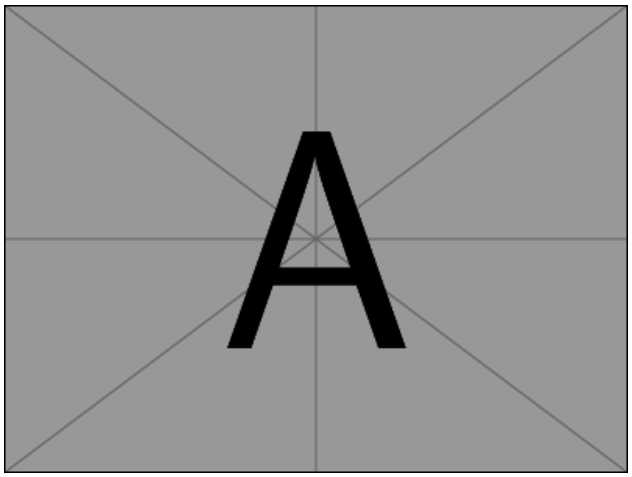


Figure 2-9: Demonstration of geodesic distance

2.2.3.4 Isomap

Traditional Isomap. While MDS has proven to be successful in a variety of applications, it suffers from the fact that is solely aims to retain pairwise Euclidean distances and does not consider the distributions of neighboring samples. This implies that MDS is not able to capture the geometry of high-dimensional data which lies on or near to a curved manifold, such as the

Swiss roll dataset (van der Maaten et al., 2007; Chao et al., 2019). Isometric Feature Mapping (Isomap) (Tenenbaum et al., 2000) is a technique which resolves this problem by attempting to preserve pairwise geodesic distances between datapoints. Isomap can be considered as a generalization of classical MDS in which the pairwise distance matrix is replaced by a matrix of pairwise geodesic distances approximated by distances in the graph (Thorstensen, 2009). The classic, unsupervised algorithm consists of a few steps:

1. Given a set of input data $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N \subset \mathbb{R}^D$, construct a sparse neighborhood graph (such as the K-nearest or ϵ-ball graphs discussed previously) where each edge is weighted by the Euclidean distance between the neighbors it connects:

$$\boldsymbol{W}_{mn} = w_{mn} = ||\boldsymbol{x}_m - \boldsymbol{x}_n||^2 \tag{2-48}$$

   where $\boldsymbol{W}$ is the graph adjacency matrix.

2. Next, the geodesic distances between all pairs of samples is computed by finding the shortest paths between the points through the graph. This is commonly done with Dijkstra's or Flyod's shortest-path algorithms (Tenenbaum et al., 2000).

3. These geodesic distances form a pairwise distance matrix which is substituted into classical MDS as described in Section 2.2.2.2. This provides the low-dimensional embedding coordinates $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_N^T] \in \mathbb{R}^{N \times d}$ of high-dimensional input data $\boldsymbol{X} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T] \in \mathbb{R}^{N \times D}$, where $d \ll D$.

While Isomap has been successfully applied in the areas of financial analysis (Ribeiro et al., 2008), facial and object recognition (Zhang et al., 2018), visualization and classification tasks (Vlachos et al., 2002), a few important weaknesses are prevalent. First, Isomap may be topologically unstable. That is, it may construct erroneous connections in the neighborhood graph. This is known as short-circuiting, and it can severely impair the performance of Isomap. Several approaches have been proposed to nullify the short-circuiting problem, such as removing datapoints with large total flows or by removing nearest neighbors that violate local linearity of the neighborhood graph (van der Maaten et al., 2007). Another weakness of Isomap is that it may not perform correctly if there are holes in the manifold, as this causes the geodesic distances of some samples to appear further on the manifold than they truly are. A third weakness is that Isomap can fail if the manifold is non-convex. Therefore, we see that

Isomap can perform very well due to theoretical guarantees on qualities such as convergence, as long as the manifold is isometric to a convex open set of $\mathbb{R}^d$, $\mathcal{D}_{\mathcal{X}}(\boldsymbol{u}, \boldsymbol{u}') = \mathcal{D}_{\mathcal{Y}}(f(\boldsymbol{u}), f(\boldsymbol{u}'))$, meaning that the geodesic distances in the graph are almost equal to the Euclidean distances in the embedding space $\mathbb{R}^d$. Continuing, an additional drawback of Isomap is the fact that it requires the decomposition of a large, dense Gram matrix which scales with the number of training data points. If the dataset grows too large, a solution will no longer be tractable. Furthermore, the constraint on $\mathcal{X}$ to be isometric to a convex open set of $\mathbb{R}^d$ is rarely met. As mentioned in (Thorstensen, 2009), these problems may be circumvented by sparsifying large datasets using landmarks, as with Landmark Isomap (Silva and Tenenbaum, 2003) and looking at conformal maps, as is done in Conformal Isomap (Silva and Tenenbaum, 2002). Finally, as with most nonlinear manifold learning techniques, it is nontrivial to embed out-of-sample data points into the lower dimensional feature space.

Supervised Isomap Approaches. As with most traditional manifold learning methods in the literature, Isomap is not inherently well-suited for classification tasks. However, supervised approaches which consider class label information have been adopted to increase class separability in the latent embedding space. The work by Vllachos et al. (Vlachos et al., 2002) was the first to investigate a supervised adaptation of Isomap. Two supervised Isomap procedures were proposed which combine Isomap with a nearest neighbor classifier. These methods, Iso+Ada and WeightedIso take label information into consideration to scale the computed Euclidean distances utilized by Isomap by a constant factor according to class label. The idea is to make points closer in the embedding space if they have the same class label and farther if they have opposing class labels. Ribeiro et al. (Ribeiro et al., 2008) proposed an enhanced supervised Isomap (ES-Isomap) in which the dissimilarity matrix is weighted according to rules which consider class label information. The dissimilarity matrix (considered as the adjacency matrix), $\boldsymbol{W}$, which was the same used in the Supervised Isomap method (X. Geng et al.,

2005), is defined as:

$$\boldsymbol{W}(\boldsymbol{x}_m, \boldsymbol{x}_n) = \begin{cases} \sqrt{1 - \exp \frac{-\mathcal{D}^2(\boldsymbol{x}_m, \boldsymbol{x}_n)}{\beta}}, & l_m = l_n \\ \sqrt{\exp \frac{\mathcal{D}^2(\boldsymbol{x}_m, \boldsymbol{x}_n)}{\beta}} - \alpha, & l_m \neq l_n \end{cases} \tag{2-49}$$

where $\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ denotes the distance measure between samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, β is used to prevent $\boldsymbol{W}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ from increasing too quickly when $\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ is large and is typically set according to the density of the data, α is a constant in $[0, 1]$ which controls the dissimilarity between points in different classes and keeps the graph from becoming disconnected, and l_m and l_n are the corresponding class labels of samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, respectively. In Equation 2-49, the dissimilarity between two points is greater than or equal to one if their class labels are different and less than 1 if the points have the same class label. Therefore, the between-class dissimilarity will always be larger than the within-class, which is an important property for classification tasks.

Li and Guo proposed Supervised Isomap with Explicit mapping (SE-Isomap) in (Chun-Guang Li and Jun Guo, 2006). SE-Isomap enforces discriminability on the matrix of geodesic distances, as compared to the Euclidean distance matrix used in the aforementioned approaches, to learn an explicit mapping to the low-dimensional embedding space. Finally, Zhang et al. (Zhang et al., 2018) developed a semi-supervised Isomap to utilize both labeled and unlabeled data points in training. This method aims at minimizing pairwise distances of within-class samples in the same manifold while maximizing the distances over different manifolds.

2.2.3.5 Sammon Mapping

Classical scaling, a convex technique for multidimensional scaling, was introduced in Section 2.2.2.2. As discussed, a pitfall of MDS is that it focuses on retaining global pairwise distances as opposed to local distances, which are typically much more important for capturing the geometry of the data (van der Maaten et al., 2007). Several MDS variants have been proposed to address this weakness. A popular variant is *Sammon Mapping* (Sammon, 1969).

The classical scaling cost function looks to minimize the difference between the pairwise distance matrices of input data and their corresponding low-dimensional representations. This cost puts emphasis on retaining the global data structure. Sammon mapping adapts the classical scaling cost function by weighting the contribution of each sample pair $(\boldsymbol{x}_m, \boldsymbol{x}_n)$ by the inverse of their pairwise distances in the high-dimensional input space $\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)$. In this way, the objective becomes

$$J(\boldsymbol{X}, \boldsymbol{Z}) = \frac{1}{\sum_{m,n} \mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)} \sum_{m \neq n} \frac{(\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n) - \mathcal{D}(\boldsymbol{z}_m, \boldsymbol{z}_n))^2}{\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)} \tag{2-50}$$

where $\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ denotes the Euclidean distance between high-dimensional input samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, $\mathcal{D}(\boldsymbol{z}_m, \boldsymbol{z}_n)$ is the Euclidean distance between low-dimensional data coordinates $\boldsymbol{z}_m$ and $\boldsymbol{z}_n$ and the constant in the front is added to simplify the gradient of the objective. This cost function gives more weight to preserving distances between samples that are close in the input space. Minimization of the Sammon objective function is typically performed with gradient descent or a pseudo-Newton method (Sorzano et al., 2014).

A weakness (and strength) of Sammon mapping is that it will give much more importance to retaining a very small distance, say 10^{-5}, as compared to 10^{-4}. Despite this, Sammon mapping has been successfully applied to visualization tasks and has reported on applications using gene data and geospatial information (van der Maaten et al., 2007).

2.2.3.6 Maximum Variance Unfolding (MVU)

Section 2.2.3.2 described how Kernel PCA (Schölkopf et al., 1999; Wang, 2012) allows PCA to be performed in a feature space defined by a kernel function κ. However, the choice of kernel function is arbitrary and may not be optimal for learning the intrinsic structure of a set of data. To resolve this issue, *Maximum Variance Unfolding* (MVU, formerly known as *Semidefinite Embedding*) attempts to learn an appropriate kernel matrix (Weinberger et al., 2004) to be used in conjunction with KPCA. This is done by defining a neighborhood graph over the data and retaining the pairwise distances through the embedding to a low-dimensional space (as done in Isomap (2.2.3.4) (Tenenbaum et al., 2000)). Unlike Isomap, however,

MVU attempts to "unfold" the data data manifold by maximizing the Euclidean distances between data points under the constraint that the local geometry of the data manifold is unperturbed (van der Maaten et al., 2007). The optimization can be solved using semidefinite programming. MVU begins by forming a k-nearest neighbor graph G. It then tries to maximize the sum of Euclidean distances between all samples such that the distances inside G are unchanged. This equates to the to following maximization problem:

$$\max_{\boldsymbol{Z}} \sum_{m,n} ||\boldsymbol{z}_m - \boldsymbol{z}_n||^2 \quad s.t. \quad ||\boldsymbol{z}_m - \boldsymbol{z}_n||^2 = ||\boldsymbol{x}_m - \boldsymbol{x}_n||^2 \quad \forall (m,n) \in G \tag{2-51}$$

This objective can be solved through the optimization of a semidefinite programming problem (SDP) for the kernel matrix $\boldsymbol{K}$ (Weinberger et al., 2004):

$$\begin{aligned} &\max_{\boldsymbol{K}} \quad Tr(\boldsymbol{K}) \quad s.t. \\ &1. \quad \boldsymbol{K} \succ 0 \\ &2. \quad \sum_{m,n} \boldsymbol{K}_{m,n} = 0 \\ &3. \quad \boldsymbol{K}_{mm} + \boldsymbol{K}_{nn} - 2\boldsymbol{K}_{mn} = ||\boldsymbol{x}_m - \boldsymbol{x}_n||^2 \quad \forall (m,n) \in G \end{aligned} \tag{2-52}$$

with $\boldsymbol{K}$ defined as the outer product matrix of the low-dimensional data coordinates $\boldsymbol{Z}$. The solution of the SDP provides a kernel matrix which is used in Kernel PCA. The low-dimensional embedding coordinates $\boldsymbol{Z}$ are found by eigendecomposition of the kernel, as described in Section 2.2.3.2.

Similarly to Isomap, MVU may suffer from short-circuiting due to optimization constraints which impair successful manifold unfolding. Despite this weakness, MVU has been applied successfully to applications on microarray data and sensor localization (van der Maaten et al., 2007).

2.2.3.7 Locally Linear Embedding (LLE)

Locally Linear Embedding (LLE) was first introduced by Roweis and Saul (Roweis and Saul, 2000) and it is, along with Isomap, a foundational graph-based approach for nonlinear

manifold learning. Whereas Isomap attempts to preserve global pairwise distances, LLE attempts to preserve solely local properties of the data (van der Maaten et al., 2007). Since LLE looks solely at local neighborhoods around samples, it is much less sensitive to short-circuiting than Isomap. Furthermore, the preservation of local properties allows the algorithm to effectively embed non-convex manifolds. Essentially, LLE looks to represent each data sample as a linear combination of its k-nearest neighbors. This fits a hyperplane through every datapoint and it's nearest neighbors, thus assuming the intrinsic manifold is locally linear. The local linearity assumption implies that the reconstruction weights $\boldsymbol{w}_n$ of high-dimensional sample $\boldsymbol{x}_n$ are invariant to transformations such as translation, rotation and scaling. Therefore, it is assumed that the same reconstruction weights $\boldsymbol{w}_n$ that can be used to represent $\boldsymbol{x}_n$ in the high-dimensional space will also reconstruct it's low-dimensional representation $\boldsymbol{z}_n$ from its corresponding neighbors in the low-dimensional space (Roweis and Saul, 2000; Sorzano et al., 2014; Chao et al., 2019). Therefore, low-dimensional data representations $\boldsymbol{Z}$ are found by minimizing the objective

$$J(\boldsymbol{W}, \boldsymbol{Z}) = \sum_{n=1}^{N} \left|\left| \boldsymbol{z}_n - \sum_{k=1}^{K} w_{nk} \boldsymbol{z}_k \right|\right|^2 \quad s.t. \quad \frac{1}{N} \boldsymbol{Z}^T \boldsymbol{Z} = \mathbb{I}_d \tag{2-53}$$

The constraint on the covariance of the embedded data representations is included to avoid the trivial solution $\boldsymbol{Z} = \boldsymbol{0}$. Solving for the low-dimensional data representations is done in three steps (Thorstensen, 2009). First, a k-nearest neighborhood graph G is constructed from the high-dimensional data $\boldsymbol{X}$. Next, the reconstruction weight vector $\boldsymbol{w}_n$ that best represents each point $\boldsymbol{x}_n$ as a linear combination of its k-nearest neighbors is found by optimizing the problem

$$\min_{\boldsymbol{W}} \sum_{n=1}^{N} \left|\left| \boldsymbol{x}_n - \sum_{k=1}^{K} w_{nk} \boldsymbol{x}_k \right|\right|^2 \quad s.t. \quad \sum_{k=1}^{K} w_{nk} = 1 \tag{2-54}$$

Optimization of Equation 2-54 can be solved directly by using the method of Lagrange Multipliers. Finally, the low-dimensional embedding coordinates are found by minimizing the quadratic error function for $\boldsymbol{Z}$ according to Equation 2-53. Roweis and Saul Roweis and Saul (2000) showed that the coordinates that minimize the objective can be found as

the eigenvectors corresponding to the d smallest nonzero eigenvalues of the inner product $(\mathbb{I}_N - \boldsymbol{W})^T(\mathbb{I}_N - \boldsymbol{W})$ where $\boldsymbol{W}$ is a $N \times N$ sparse matrix whose entries are equal to the corresponding reconstruction weight if $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ are connected in the neighborhood graph and 0 otherwise.

The popularity of LLE has led to the development of linear variantsDonoho2003HessianEigenmaps, namely *Neighborhood Preserving Projections* (NPP) and *Orthogonal Neighborhood Preserving Projections* (Pang et al., 2005; Kokiopoulou and Saad, 2007), and has been applied successfully to applications in super-resolution and sound localization (van der Maaten et al., 2007). However, LLE tends to collapse large portions of the data close to each other in the embedding space due to the constraint on the covariance matrix. This may also result in undesired scalings of the manifold. Despite these effects, supervised approaches based on LLE have also been developed (Chao et al., 2019; Li et al., 2009). Existing approaches can be summarized by the LDA idea, that points in the same class should be embedded more closely to each other while points in opposing classes should be well-separated in the low-dimensional space. This notion is realized by altering the dissimilarity matrix used to construct the neighborhood graph.

2.2.3.8 Laplacian Eigenmaps (LE)

Classical LE. Similar to LLE, Laplacian Eigenmaps, or *Spectral Embedding*, is a nonlinear dimensionality reduction technique which aims to preserve local structure of data (Belkin and Niyogi, 2003; Raducanu and Dornaika, 2012; van der Maaten et al., 2007). Using *spectral graph theory*, LE computes low-dimensional representations of data in which the dissimilarities between datapoints and their neighbors (according to an affinity measure) are minimized. The name *Laplacian Eigenmaps* is derived by the use of Laplacian regularization in the optimization procedure (Thorstensen, 2009). Given a set of N samples $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N \subset \mathbb{R}^D$, the first step of LE is to define a *neighborhood graph on the samples*. This graph, also called an *affinity* or *adjacency* matrix can be constructed in a variety of ways, such as K-nearest neighbor, ϵ-ball, full mesh, or by weighting each edge $\boldsymbol{x}_m \sim \boldsymbol{x}_n$ by a symmetric affinity function

$W_{mn} = K(\boldsymbol{x}_m; \boldsymbol{x}_n)$, typically a radial basis or heat kernel:

$$\boldsymbol{W}_{mn} = w_{mn} = \exp\left(-\frac{||\boldsymbol{x}_m - \boldsymbol{x}_n||^2}{\beta}\right) \tag{2-55}$$

where the kernel bandwidth β is typically set as the variance of the dataset (Raducanu and Dornaika, 2012; Thorstensen, 2009).

The goal is to uncover the latent data representations $\{\boldsymbol{z}_n\}_{n=1}^N \subset \mathbb{R}^d$ where $d \ll D$ which minimizes the objective

$$J(\boldsymbol{W}, \boldsymbol{Z}) = \frac{1}{2}\sum_{m,n} ||\boldsymbol{z}_m - \boldsymbol{z}_n||^2 w_{mn} = tr(\boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{Z}) \tag{2-56}$$

with $\boldsymbol{W}$ denoting the symmetric affinity matrix, $\boldsymbol{D}$ the diagonal weight matrix whose entries are the sum of the rows (or columns since $\boldsymbol{W}$ is symmetric) of $\boldsymbol{W}$ (i.e. $d_{mm} = \sum_n w_{mn}$, and is 0 otherwise). The graph Laplacian matrix is provided as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$. The matrix $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_N^T]$ is the $N \times d$ embedding matrix and $tr(.)$ denotes the trace of a matrix. The n^{th} row of matrix $\boldsymbol{Z}$ provides the vector $\boldsymbol{z}_n$, which is the latent representation of sample $\boldsymbol{x}_n$. This objective discourages projecting similar points in the input feature space to disparate regions of the embedding space by enforcing heavy penalization.

The latent sample coordinates $\boldsymbol{Z}$ are found as the solution to the optimization problem:

$$\min_{\boldsymbol{Z}} tr(\boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{Z}) \quad s.t. \quad \boldsymbol{Z}^T \boldsymbol{D} \boldsymbol{Z} = \boldsymbol{I}, \boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{e} = \boldsymbol{0} \tag{2-57}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{e} = (1, \ldots, 1)^T$. The first constraint eliminates the trivial solution $\boldsymbol{Z} = \boldsymbol{0}$ (scaling) and the second constraint avoids the trivial solution $\boldsymbol{Z} = \boldsymbol{e}$ (uniqueness). By applying the Langrange multiplier method and using the fact that $\boldsymbol{L}\boldsymbol{e} = \boldsymbol{0}$, the low-dimensional data representations can be found by solving the generalized eigenvalue problem:

$$\boldsymbol{L}\boldsymbol{v} = \lambda \boldsymbol{D}\boldsymbol{v} \tag{2-58}$$

The column vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ are the solutions of Equation 2-58, ordered to the corresponding eigenvalues, in ascending order, $\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_N$. The embedding of the input samples given by the matrix $\boldsymbol{Z}$, is obtained by concatenating the eigenvectors of the d smallest non-zero eigenvalues. $\boldsymbol{Z}$ is a $N \times d$ matrix, where $d < N$ is the dimensionality of the embedded space. From observation, it is clear that the embedding dimensionality is limited by the number of samples N.

Linear LE (LPP). Because of the representation ability of LE, a linear approach called *Locality Preserving Projections* (LPP) was proposed by He and Niyogi (He and Niyogi, 2003). Similarly to LE, LPP builds a neighborhood graph which incorporates local information of the data. LPP optimizes a similar objective to LE, however, it is assumed that the low-dimensional data representations come from a linear transformation of the high-dimensional input data. The primary benefit of LPP over LE is that, while it shares many of the representation properties of LE, it is defined everywhere in the latent space as opposed to just over the training data points. This allows for simple embeddings of out-of-sample test points. Moreover, one can perform LPP in the original space or in a reproducing kernel Hilbert space through the use of Mercer kernels as described in Section 2.2.3.1.

Supervised LE (S-LE). In order to adopt LE for classification,Raducanu and Dornaika (Raducanu and Dornaika, 2012) proposed a supervised LE which minimizes the margin between samples with similar class labels and maximizes the margin between samples with opposing class labels. Supervised LE utilizes discriminative information contained in the class labels when finding the nonlinear embedding (spectral projection).

In order to discover both geometrical and discriminative manifold structure, supervised LE splits the global graph into two components: the within-class graph G_w and the between-class graph G_b. To define the margin, they define two subsets, $N_w(\boldsymbol{x}_n)$ and $N_b(\boldsymbol{x}_n)$ for each sample $\boldsymbol{x}_n$. These two subsets contain the neighbors of $\boldsymbol{x}_n$ sharing the same label and having different

labels, respectively, which have a similarity higher than the average.

$$N_w(\boldsymbol{x}_n) = \{\boldsymbol{x}_m | l_m = l_n, \exp\left(-\frac{||\boldsymbol{x}_n - \boldsymbol{x}_m||^2}{\beta}\right) > AS(\boldsymbol{x}_n)\} \tag{2-59}$$

$$N_b(\boldsymbol{x}_n) = \{\boldsymbol{x}_m | l_m \neq l_n, \exp\left(-\frac{||\boldsymbol{x}_n - \boldsymbol{x}_m||^2}{\beta}\right) > AS(\boldsymbol{x}_n)\} \tag{2-60}$$

where $AS(\boldsymbol{x}_n) = \frac{1}{N}\sum_{n=1}^{N} \exp\left(-\frac{||\boldsymbol{x}_n - \boldsymbol{x}_m||^2}{\beta}\right)$ denotes the average similarity of the sample $\boldsymbol{x}_n$ to the rest of the data. From Equations 2-59 and 2-60 it is clear that the neighborhoods for each data sample are not necessarily the same size. As a result, this function constructs the affinity graph according to both the local density and similarity between data samples in the input feature space.

With the two sets defined, the within-class and between-class weight matrices $\boldsymbol{W}_w$ and $\boldsymbol{W}_b$ are formed from the adjacency graphs G_w and G_b, respectively. These weight matrices are defined as:

$$W_{w,mn} = \begin{cases} \exp\left(-\frac{||\boldsymbol{x}_n - \boldsymbol{x}_m||^2}{\beta}\right), & \text{if } \boldsymbol{x}_n \in N_w(\boldsymbol{x}_m) \text{ or } \boldsymbol{x}_m \in N_w(\boldsymbol{x}_n) \\ 0, & \text{otherwise} \end{cases} \tag{2-61}$$

$$W_{b,mn} = \begin{cases} 1, & \text{if } \boldsymbol{x}_n \in N_b(\boldsymbol{x}_m) \text{ or } \boldsymbol{x}_m \in N_b(\boldsymbol{x}_n) \\ 0, & \text{otherwise} \end{cases} \tag{2-62}$$

and the global affinity matrix, $\boldsymbol{W}$, can be written as:

$$\boldsymbol{W} = \boldsymbol{W}_w + \boldsymbol{W}_b \tag{2-63}$$

In order to obtain the low-dimensional representations $\boldsymbol{z}_n$ of the input data $\boldsymbol{x}_n$, the following objective functions can be optimized for $\boldsymbol{Z}$:

$$\min \frac{1}{2} \sum_{m,n} ||\boldsymbol{z}_m - \boldsymbol{z}_n||^2 W_{w,mn} = tr(\boldsymbol{Z}^T \boldsymbol{L}_w \boldsymbol{Z}) \tag{2-64}$$

$$\max \frac{1}{2} \sum_{m,n} ||\boldsymbol{z}_m - \boldsymbol{z}_n||^2 W_{b,mn} = tr(\boldsymbol{Z}^T \boldsymbol{L}_b \boldsymbol{Z}) \tag{2-65}$$

where $\boldsymbol{L}_w = \boldsymbol{D}_w - \boldsymbol{W}_w$ and $\boldsymbol{L}_b = \boldsymbol{D}_b - \boldsymbol{W}_b$ indicate the corresponding graph Laplacians of the within-class and between-class affinity graphs, respectively. The matrix $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_N^T]$ contains the low-dimensional representations of the input samples in its rows.

By merging the two objective functions, the final optimization problem is formulated as:

$$\arg\max_{\boldsymbol{Z}} \left\{ \gamma tr(\boldsymbol{Z}^T \boldsymbol{L}_b \boldsymbol{Z}) + (1-\gamma) tr(\boldsymbol{Z}^T \boldsymbol{W}_w \boldsymbol{Z}) \right\} \quad s.t. \quad \boldsymbol{Z}^T \boldsymbol{D}_w \boldsymbol{Z} = \boldsymbol{I} \tag{2-66}$$

The term γ is a scalar value in $[0, 1]$ which determines the trade-off between pulling similar samples toward each other in the latent space and pushing heterogeneous points away. A value of $\gamma = 1$ forces the objective to solely focus on maximizing the margin between dissimilar points. Alternatively, a value of $\gamma = 0$ priorities the objective on embedding homogeneous samples in close spatial proximity. By defining matrix $\boldsymbol{B} = \gamma \boldsymbol{L}_b + (1-\gamma) \boldsymbol{W}_w$, the problem becomes:

$$\arg\max_{\boldsymbol{Z}} \left(\boldsymbol{Z}^T \boldsymbol{B} \boldsymbol{Z} \right) \quad s.t. \quad \boldsymbol{Z}^T \boldsymbol{D}_w \boldsymbol{Z} = \boldsymbol{I} \tag{2-67}$$

The low-dimensional embedding matrix $\boldsymbol{Z}$ can be found by solving the generalized eigenvalue problem:

$$\boldsymbol{B}\boldsymbol{v} = \lambda \boldsymbol{D}_w \boldsymbol{v} \tag{2-68}$$

The column vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N$ are the generalized eigenvectors of Equation 2-68 arranged by descending eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_d$. Then the $N \times d$ embedding matrix $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_N^T]$ is provided by concatenating the obtained eigenvectors $\boldsymbol{Z} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d]$.

The primary difference between the classic LE and S-LE is that traditional LE solely attempts to preserve the spatial relationships between samples, and thus, does not consider label information when learning the embeddings. Alternatively, S-LE aims at aiding discriminant analysis by collapsing the distance between samples with the same label that are in close spatial proximity and pushing away spatial neighbors with differing class labels. This is done through the utilization of two affinity graphs: the within-class and between-class graphs. As with most graph-based methods, LE results vary highly according to the choice of neighborhood size. However, choosing the size of K or ϵ in advance can be very difficult. S-LE does not require user-defined graph parameters, other than those associated with the chosen affinity measure. Instead, graph edges are chosen according to an adaptive neighborhood for each sample. Both methods, however, suffer from inherent difficulties associated with nonlinear manifold learning, namely, selecting the intrinsic embedding dimensionality and handling out-of-sample extensions.

Despite these nuances, LE (and its variants) have been successfully applied in nonlinear dimensionality reduction tasks for facial recognition, spectral clustering and object classification van der Maaten et al. (2007).

Apart from S-LE, other methods have been explored to integrate label information into Laplacian Eigenmaps. A review of supervised dimensionality reduction methods by Chao et al. (Chao et al., 2019) explains that author's have optimized the affinity matrix using label information after constructing from spatial proximity, proposed deep learning-based approaches to achieve supervised LE and integrated label information into the affinity matrix construction process.

The special feature exhibited by all Laplacian Eigenmap methods is the use of laplacian regularization, which enforces properties such as smoothness and provides a level of resistance toward the influences of outliers. This useful feature has been applied in a variety of supervised and semi-supervised tasks, such as hyperspectral and synthetic aperture radar remote sensing classification (Ratle et al., 2010; Ren et al., 2017), classification of synthetic data (Tsang and

Kwok, 2007), zero-shot learning (Meng and Zhan, 2018) and reinforcement learning (Li et al., 2015).

2.2.3.9 Hessian LLE

Hessian LLE (HLLE) (Donoho and Grimes, 2003) is a variant of LLE that minimizes the curvature of the high-dimensional manifold when embedding it into the low-dimensional space, under the constraint that the intrinsic manifold is locally isometric (van der Maaten et al., 2007; Thorstensen, 2009). This is achieved by the eigenanalysis of the local Hessian matrix $\mathcal{H}$, which measures the curvature of the manifold around every data point. HLLE begins by identifying the k-nearest neighbors for each sample $\boldsymbol{x}_n$ using Euclidean distance (local linearity). Because of the linearity assumption, a d-dimensional basis for the local tangent space of each sample can be found by PCA or singular value decomposition (SVD) on each sample's corresponding neighbors. Then, a local Hessian $\boldsymbol{H}^n$ of the manifold is estimated at each point $\boldsymbol{x}_n$ in the local tangent space. The data Hessian matrix is constructed from the individual estimators derived from the local tangent coordinates and is given as

$$\mathcal{H}_{mo} = \sum_n \sum_r ((\boldsymbol{H}^n)_{rm}(\boldsymbol{H}^n)_{ro}) \tag{2-69}$$

The symmetric matrix $\mathcal{H}$ gives an estimate of the curvature of the high-dimensional data manifold. The low-dimensional embedding coordinates $\boldsymbol{Z}$ of the input data are given by the eigenvectors of the corresponding d smallest eigenvalues of $\mathcal{H}$.

HLLE shares many of the characteristics of Laplacian Eigenmaps, except that it replaces the manifold Laplacian with the manifold Hessian. Because of this, HLLE suffers from the same weaknesses as Laplacian Eigenmaps (van der Maaten et al., 2007). Desipite its weaknesses, HLLE has been applied successfully to sensor localization tasks (Patwari and Hero, 2004).

2.2.3.10 Local Tangent Space Alignment (LTSA)

Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2002) shares similarities with HLLE in that it describes local properties of high-dimensional data using the local tangent

space of each data point (van der Maaten et al., 2007). The basic idea of LTSA is to use the tangent space in the neighborhood of a data point to represent the local geometry of the data, and then to align the local tangent spaces to construct a global coordinate system for the nonlinear manifold. In this way, the algorithm consists of two steps: 1.) constructing the principal manifold that is tangential to each data point and 2.) finding the global coordinate system that describes the set of data samples in a low-dimensional space (Zhang and Zha, 2002).

Just as with Hessian LLE, LTSA begins by computing d-dimensional bases for the local tangent spaces of each of the data points $\boldsymbol{x}_n$. This can be done by applying PCA or SVD on the neighbors of each sample. This provides a linear transformation matrix $\boldsymbol{U}_n$ from the neighborhood of $\boldsymbol{x}_n$ to the local tangent space Θ_n. It is assumed that there exists a linear mapping $\boldsymbol{M}_n$ from the local tangent space coordinates $\boldsymbol{\theta}_n$ to the low-dimensional representations $\boldsymbol{z}_n$. The linear transformation matrix and low-dimensional data representations are found by the following minimization problem (Sorzano et al., 2014):

$$\min_{\boldsymbol{M}_n, \boldsymbol{Z}_n} \sum_{n=1}^{N} ||\boldsymbol{Z}_n \boldsymbol{H} - \boldsymbol{M}_n \Theta_n||_F^2 \tag{2-70}$$

where $\boldsymbol{H}$ is a centering matrix. It was shown in (Zhang and Zha, 2002) that the low-dimensional embedding coordinates $\boldsymbol{Z}$ can be constructed by the eigenvectors corresponding to the d smallest nonzero eigenvalues of an alignment matrix $\boldsymbol{B}$.

Similar to other sparse spectral manifold learning techniques, the objective function of LTSA is vulnerable to the presence of the trivial solution (van der Maaten et al., 2007). However, LTSA has been successfully applied to applications in facial recognition (Zhang et al., 2007) and microarray data (van der Maaten et al., 2007). Moreover, supervised approaches for LTSA were proposed in (Li et al., 2005) and (Ma et al., 2010).

2.2.3.11 Diffusion Maps

The success of "kernel eigenmap approaches" such as LLE, LE, HLLE and LTSA led to the development of a manifold learning method called *Diffusion Maps* (DM) (Coifman

and Lafon, 2006). Diffusion maps is a framework that originates from the field of dynamical systems and is based on defining a Markov random walk on the graph of the data. By performing a random walk for a number of time-steps, a measure for the proximity of the data points is obtained (van der Maaten et al., 2007). Essentially, a DM embeds data into a lower-dimensional space such that the Euclidean distance between points is approximated by the diffusion distance in the original feature space.

In the diffusion maps framework, a graph of the data is first constructed which measures the connectivity between all points in the graph. The *connectivity* of two samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ is defined as the probability of transitioning from $\boldsymbol{x}_m$ to $\boldsymbol{x}_n$ in one step of the random walk (Hajek, 2015). The weights and edges in the graph are computed by a heat-kernel (commonly called *radial-basis function* RBF), which provides a weight matrix $\boldsymbol{W}$ with entries

$$\boldsymbol{W}_{mn} = w_{mn} = \kappa(\boldsymbol{x}_m, \boldsymbol{x}_n) = \exp\left(-\frac{||\boldsymbol{x}_m - \boldsymbol{x}_n||^2}{\beta}\right) \tag{2-71}$$

where β is the bandwidth of the isotropic function. By picking an appropriate kernel width, this constructs a matrix which measures the similarity within a certain neighborhood around each data point. Outside of the neighborhood, the function quickly goes to zero. Since this matrix should define the probabilities of jumping between data points, a sum-to-one constraint is enforced such that the one-step transition probability matrix is defined as

$$\boldsymbol{P}^{(1)}_{mn} = p^{(1)}_{mn} = \frac{w_{mn}}{\sum_{o=1}^{N} w_{mo}} \tag{2-72}$$

This matrix represents the probability of a transition from one data sample to another in a single time-step. Relating DM to Laplacian Eignemaps, this matrix can also be interpreted as the normalized graph Laplacian matrix (Thorstensen, 2009). Taking advantage of the Markov assumption, the forward transition probability matrix for t time-steps $\boldsymbol{P}^{(t)}$ is given by $(\boldsymbol{P}^{(1)})^t$ (Hajek, 2015) and a *diffusion distance* can be defined as

$$D^{(t)}(\boldsymbol{x}_m, \boldsymbol{x}_n) = \sum_{o=1}^{N} ||\boldsymbol{P}^{(t)}_{mo} - \boldsymbol{P}^{(t)}_{on}||^2 \tag{2-73}$$

From Equation 2-73, it can be observed that the diffusion distance is small if there are many high-probability paths of length t between two samples. By integrating over all paths through the graph, DM is more robust to short-circuiting and noise perturbation than other approaches, such as the geodesic distance utilized in the Isomap algorithm (van der Maaten et al., 2007; Delaporte et al., 2008; Tenenbaum et al., 2000). The objective of a *diffusion map* is then to embed the high-dimensional data coordinates into a lower-dimensional space such that the diffusion distance in the input space becomes Euclidean distance in the embedding space. It was shown in (Coifman and Lafon, 2006; Delaporte et al., 2008) that the low-dimensional data representations $\boldsymbol{Z}$ that retain the diffusion distances in the Euclidean embedding space can be formed by the eigenvectors of the d largest, nontrivial eigenvalues of the eigenproblem

$$\boldsymbol{P}^{(t)}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{2-74}$$

Since the graph is fully-connected, the largest eigenvalue is trivial (i.e. $\lambda_1 = 1$) and its corresponding eigenvector $\boldsymbol{v}_1$ is discarded. Thus, $\boldsymbol{Z} \in \mathbb{R}^{N\times d}$ is given by

$$\boldsymbol{Z} = \left[\lambda_2\boldsymbol{v}_2, \lambda_3\boldsymbol{v}_3, \ldots, \lambda_{d+1}\boldsymbol{v}_{d+1}\right] \tag{2-75}$$

Diffusion maps have been successfully applied to tasks in shape matching and gene expression analysis (van der Maaten et al., 2007).

2.2.4 Principal Curves, Surfaces and Manifolds

Principal Curves (PCs) were first proposed in a PhD thesis by Hastie in 1984 and are simply defined as lines or surfaces that pass through the "middle of a cloud representing a data set" (Kegl et al., 2008; Gorban and Zinovyev, 2008). In terms of probability distributions, a principal curve satisfies the self-consistency property, which implies that any point on the curve is the average of all data points projected onto it. In fact, Principal Component Analysis (Section 2.2.2.1) is a special case of PCs where the "middle structure" of data is assumed as a straight line (or hyper-plane) instead of a curve. Assuming data with zero mean, PCA

looks for a hyper-plane passing through the origin with direction $\boldsymbol{w}_1$ with a linear mapping function $f(\boldsymbol{z}_n) = \hat{\boldsymbol{x}}_n = \boldsymbol{w}_1^T \boldsymbol{z}_n$ best approximates the data in the mean-square sense (Sorzano et al., 2014; Murphy, 2012). The point $f(\boldsymbol{z}_n)$ is the orthogonal projection onto the line parameterized by $\boldsymbol{w}_1$. A new sample can be constructed as $\boldsymbol{x}_n^{'} = \boldsymbol{x}_n - f(\boldsymbol{z}_n)$. The previous procedure is repeated $d-1$ more times to discover the d "most important" lines for representing the entirety of the dataset. Dimensionality reduction is achieved by replacing the data $\boldsymbol{x}_n$ with the corresponding collection of parameters $\boldsymbol{z}_n$ needed to project the sample onto its principal hyper-planes. As noted in (Sorzano et al., 2014), the objective is simply a linear regression on the input data. However, this process does not have to be restricted to lines (or hyper-planes). In general, it can be assumed that a given sample $\boldsymbol{x}_n$ may be approximated by $\boldsymbol{x}_n = f(\boldsymbol{z}_n) + \boldsymbol{\epsilon}$ where $f(\boldsymbol{z}_n)$ is an arbitrary (but sufficient) nonlinear mapping, and $\boldsymbol{\epsilon}$ is a term that captures the approximation error. One could substitute for a fixed family of curves (parabolic, hyperbolic, exponential, polynomial, etc.). This is exactly a nonlinear regressor, and several methods based on artificial neural networks have been proposed (Nonlinear PCA (Kramer, 1991), autoencoder networks (Goodfellow et al., 2016), self-organizing networks (Kohonen, 1990; Fritzke, 1994)) This section will review popular methods for principal manifold estimation, including approaches based on deep-learning, self-organizing maps, neural gases and elastic maps.

2.2.4.1 Deep Learning

Many natural phenomena behave in a nonlinear way, meaning that the observed data define a curved subspace in the original feature space (Scholz et al., 2008). One method used to model this nonlinearity is to utilize a *deep feedforward network*, also often called *feedforward neural network* or *multilayer perceptron* (MLP) (Goodfellow et al., 2016). The goal of a feedforward network is to approximate a function f which maps an input $\boldsymbol{x}$ to a desired output $\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{\theta})$. These models are called "networks" because they are typically represented as a composition of functions (usually perceptrons). In the standard perceptron model, the output is formed as a weighted combination of the inputs which is then passed through a nonlinear

activation function. Weights on the input signals are adjusted based on an error signal back-propagated through the network. This model is similar to human neurons, where some input pathways (synapses) are strengthened for particular tasks. The multilayer perceptron has been proven as a *universal approximator* (Haykin, 2009; Principe et al., 1999). This means that (under certain conditions), neural models have the ability to approximate any continuous function. Thus, feedforward neural networks have shown remarkable performance in countless applications (Liu, 2017; Dhillon and Verma, 2019; Abiodun et al., 2018; Shahid et al., 2019; Alom et al., 2019; Tschannen et al., 2018; Yuan et al., 2019; Chen et al., 2019), including manifold learning through use of a special type of neural architecture called an *autoencoder*.

Autoencoders and Nonlinear PCA.

Autoencoder neural networks (AE) fall into the category of neural architectures known as *autoassociative networks* (Rojas, 1996). The goal of an autoencoder is, given an input sample, to produce an exact copy at its output. Internally, the network has a hidden layer $\boldsymbol{h}$ that describes the latent code used to represent the input (Goodfellow et al., 2016). Essentially, the network has two parts. The first part represents the *encoder* $\Phi_{encode} : \mathcal{X} \rightarrow \mathcal{Z}$, which maps the data sample coordinates $\boldsymbol{x} \in \mathbb{R}^D$ to the corresponding coordinates $\boldsymbol{z}$ in the d-dimensional subspace controlled by the architecture of the middle layer (or *bottleneck*). The second part of the network denotes the inverse function, or *decoder* $\Phi_{decode} : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$. The job of the decoder is to learn an inverse mapping from $\boldsymbol{z}$ to the original data sample. Thus, Φ_{decode} approximates the assumed data generation process. In general, the AE is constrained either through its architecture or through a sparsity constraint. A loss function $\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ measures how closely the AE can reconstruct the output. A commonly used AE loss is MSE, which penalizes the estimated output from being different from the input, $\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$ (Ball et al., 2017). Figure 2-10 demonstrates an arbitrary feedforward autoencoder. The first half of the network projects the input data into a lower-dimensional space which is controlled by the size of the bottleneck layer. The decoder then attempts to reconstruct the input from the latent code.

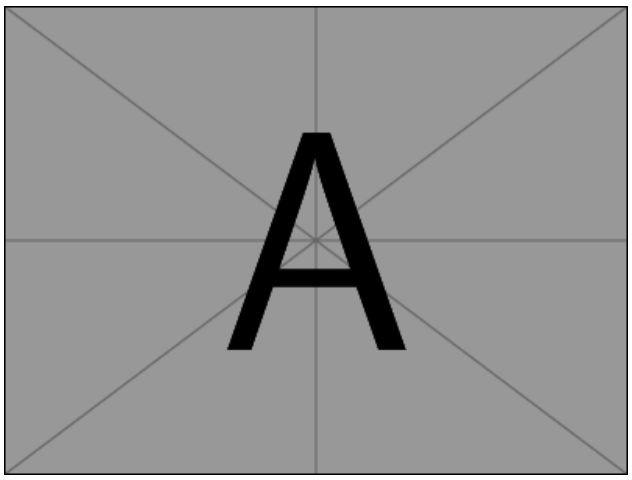


Figure 2-10

Early investigation of modern autoencoders was performed by LeCun, Bourlard, Kamp, Hinton and Zemel beginning in 1986 (Goodfellow et al., 2016; Bengio et al., 2012) and showed that having hidden layers in both the encoder and decoder enables the network to learn a *Nonlinear PCA* (NLPCA) (Scholz et al., 2008). Similarly, it has been shown that an autoencoder with only a single hidden layer will learn an equivalent transformation matrix to traditional PCA (Bengio et al., 2012).

Autoencoders suffer from the same ailments as other neural networks, such as requiring large amounts of data to cope with extreme parameterization. Moreover, vanilla AEs are unsupervised models and thus do not consider class information when determining the underlying latent features. So while AEs are capable of learning powerful feature representations, they may not be optimal for discrimination tasks. Despite these drawbacks, autoencoders have been applied successfully in countless tasks (Tschannen et al., 2018; Yuan et al., 2019; Chen et al., 2019; Kosiorek et al., 2019), including: feature representation learning, denoising, outlier detection, domain adaptation and anomaly detection in remote sensing (Bengio et al., 2012). Additionally, alternative autoencoder architectures have been developed, such as the variational AE (VAE) which assume the bottleneck layer to be the parameters of the latent code generating distributions (Ghaffarzadegan, 2018; Dai et al., 2017) and graph autoencoders (GAE) (Wu et al., 2019) which are capable of learning latent embeddings of graph-structured data.

Graph Convolutional Networks. *Geometric deep learning*, as described by (Bronstein et al., 2017), encapsulates emerging techniques which generalize deep neural models to non-Euclidean domains such as graphs and manifolds. This umbrella of models includes *graph autoencoders* (GAEs), which map graph nodes into a latent feature space and decode graph information from the latent node representations (Wu et al., 2019; Liu et al., 2018; Pan et al., 2018). Similarly to the standard autoencoder, GAEs can be used to learn lower-dimensional representations of the input data (graphs in this case) or to generate new data from latent represention. The review by (Wu et al., 2019) provides a detailed summary of GAE models in the literature to date.

2.2.4.2 Self-Organizing Map (SOM)

Self-Organizing Maps (SOM) or *Kohonen Networks* (Kohonen, 1990) are among the most well-known methods for discrete principal manifold estimation (Yin, 2007; Sorzano et al., 2014). SOMs belong to the category of neural networks which use a technique of *competitive learning* called *self-organization* to learn spatially-organized intrinsic representations of features (Rauber et al., 2002). Self-orgnanizing maps are initialized by a pre-defined set of neurons which are typically arranged in a 2 or 3-dimensional grid. Each neuron is associated with a set of weights which represent their corresponding locations in the input space. When presented an input pattern (or *stimulus*), neurons compete among themselves for the representation of the input. Winning neurons strengthen their weights or relationships with this input using and adaption of *Hebb's Rule*, which states that the change to a synaptic weight is proportional to the correlation between the input and its associated output (Rumelhart and Zipser, 1985). As samples are introduced to the network during training, only a neighborhood of cells give an active response to the current input sample. These neurons are pulled toward the location of the input pattern according to how well they represent the input. Given enough training iterations, the neurons in the lattice will disperse such that the spatial locations or coordinates of cells in the network correspond to different modes of the input distribution. In this manner,

Kohonen networks are able to utilize simple heuristics to form discrete topological mappings of the input space which represent the principal data manifold (Yin, 2007).

The self-organizing map is also a form of *vector quantization* (VQ). The purpose of VQ is to approximate a continuous probability density function $p(\boldsymbol{x})$ of input vectors $\boldsymbol{x}$ using a finite number of codebook vectors, $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$. After the "codebook" is chosen, the approximation of $\boldsymbol{x}$ involves finding the reference vector, $\boldsymbol{w}_c$ closest to $\boldsymbol{x}$. The "winning" codebook vector for sample $\boldsymbol{x}$ satisfies the following:

$$||\boldsymbol{x} - \boldsymbol{w}_c|| = \min_k ||\boldsymbol{x} - \boldsymbol{w}_k|| \tag{2-76}$$

The algorithm operates by first initializing a spatial lattice of codebook elements (also called "units"), where each unit's representative is in $\boldsymbol{w}_k \in \mathbb{R}^D$ where D is the dimensionality of the input samples $\boldsymbol{x}$. The training process proceeds as follows. A random sample is selected and presented to the network at iteration t and each unit determines its activation by computing dissimilarity (typically Euclidean distance). The unit who's codebook vector provides the smallest dissimilarity is referred to as the *winner*.

$$c(t) = \arg\min_k \mathcal{D}(\boldsymbol{x}(t), \boldsymbol{w}_k(t)) \tag{2-77}$$

Both the winning vector and all vectors within a neighborhood of the winner are updated toward the sample by

$$\boldsymbol{w}_k(t+1) = \boldsymbol{w}_k(t) + \alpha(t) \cdot h_{ck}(t) \cdot [\boldsymbol{x}(t) - \boldsymbol{w}_k(t)] \tag{2-78}$$

where $\alpha(t)$ is a learning rate which decreases over time and $h_{ck}(t)$ is a neighborhood function which is typically unimodal and symmetric around the location of the winner which monotonically decreases with increasing distance from the winner. A radial basis kernel is typically chosen for the neighborhood function as

$$h_{ck}(t) = \exp\left(-\frac{||\boldsymbol{w}_c - \boldsymbol{w}_k||^2}{\beta(t)}\right) \tag{2-79}$$

where the top expression represents the Euclidean distance between units c and k with $\boldsymbol{w}_k$ representing the 2-D or 3-D location of unit k in the lattice. The neighborhood kernel's bandwidth is typically initialized to a value which covers a majority of the input space and decreases over time such that solely the winner is adapted toward the end of the training procedure.

The SOM essentially performs density estimation of high-dimensional data and represents it in a 2 or 3-D representation. At test time, the dissimilarity between each unit in the map and an input sample are computed. This dissimilarity can be used to effectively detect outliers, thus making the SOM a robust method which can provide confidence values for it's representation abilities of the principal manifold. A downside of the SOM, however, is its lack of proven convergence criteria.

Despite the lack of theoretical training guarantees, SOMs have been applied to thousands of application, which are too numerous to list here (Yin, 2007). Examples, however, include tasks in: speech recognition, finance, computer network traffic analysis, manufacturing, image analysis (Rauber et al., 2002; Palomo and Lopez-Rubio, 2017), robotics, control of diffusion processes, optimization problems, adaptive telecommunications, image compression, sentence understanding, radar classification of sea-ice (Kohonen, 1990), cross-modal information processing, text and document mining, gene expression data analysis and discovery, novelty detection, computer animation, principal curve and surface discovery, data visualization (Yin, 2007), hand-written word classification (Chiang and Gader, 1997) and explosive hazard detection (Frigui and Gader, 2009). Moreover, numerous extensions to SOMs have been proposed, including adaptations which allow the lattice to grow and shrink to fit the input data space (Rauber et al., 2002), an approach for space visualization (Yin, 2007) and a supervised approach known as *Learning Vector Quantization* (LVQ) (Kohonen, 1995).

2.2.4.3 Growing Neural Gas (GNG)

Growing Neural Gas (GNG) networks (Fritzke, 1994) are similar to the standard SOM, except that network topology is not fixed (Sorzano et al., 2014). This approach is called

a "neural gas" because the codebook vectors are allowed to move around the data space similar to the Brownian motion of gas molecules in a closed container (Peña et al., 2007). The primary difference between GNG and SOM is that the network is allowed to shrink and grow to better fill the data space (Palomo and Lopez-Rubio, 2017; Palomo and Lopez-Rubio, 2016). The dynamic topology is typically controlled by an edge age-based strategy. Similarly to the SOM, growing neural gas networks train based on heuristics and have not been proven to converge to an optimal solution. Additionally, constraints must be put into place to avoid over-fitting the data space. Despite these nuances, GNGs have proven successful in a variety of tasks. This is likely because the automatic topology learning exhibited by these networks allows them to learn complex manifolds which may have locally-different intrinsic dimensionality (Kegl et al., 2008). Thus, GNG networks have been applied successfully to tasks in motion detection (Sun et al., 2017), visualization of high-dimensional data, web mining, classification, image deblocking (Lopez-Rubio and Palomo, 2011), computer vision, robotics, color quantization, clustering in videos (Palomo and Lopez-Rubio, 2017) and foreground detection (Palomo and Lopez-Rubio, 2016).

2.2.4.4 Elastic Maps and Nets

Elastic Maps and Nets (Gorban and Zinovyev, 2008) (also called *Principal Graphs*) are a mid-way between self-organizing maps (Section 2.2.4.2) and the generative topographic mapping (Section 2.2.5.2). Essentially, elastic maps represent a mathematical analogy of a set of elastic springs embedded in the data space which is used to approximate a low-dimensional principal manifold (Gorban and Zinovyev, 2008). Just as with the SOM, elastic maps are represented by a set of neurons (nodes), $\boldsymbol{w}_k \in \mathbb{R}^D$, $k = 1, 2, \ldots, K$. Each sample in the data set is assigned to a class based on its closest neuron

$$C_k = \{\boldsymbol{x} | \boldsymbol{w}_k \text{ is the closest codebook vector to } \boldsymbol{x}\} \tag{2-80}$$

The *approximation energy* for the entire map measures the representation ability of each codebook vector to approximate the data assigned to it

$$U_A = \frac{1}{2}\sum_{k=1}^{K}\sum_{\boldsymbol{x}\in C_k}||\boldsymbol{x}-\boldsymbol{w}_k||^2 \tag{2-81}$$

This is analogous to the energy of the springs with unit elasticity which connect each data point to the codebook which is the most representative. Additional structure is also enforced on the map to simulate stretching and bending. Specifically, some pairs of nodes $(\boldsymbol{w}_m, \boldsymbol{w}_n)$ are connected by *elastic edges* and some triplets of neurons $(\boldsymbol{w}_m, \boldsymbol{w}_n, \boldsymbol{w}_o)$ form *bending ribs*. The *stretching energy* and *bending energy* on these sets E and R are then defined by

$$U_E = \lambda\frac{1}{2}\sum_{(\boldsymbol{w}_m,\boldsymbol{w}_n)\in E}||\boldsymbol{w}_m-\boldsymbol{w}_n||^2 \tag{2-82}$$

$$U_R = \mu\frac{1}{2}\sum_{(\boldsymbol{w}_m,\boldsymbol{w}_n,\boldsymbol{w}_o)\in R}||\boldsymbol{w}_m-2\boldsymbol{w}_n+\boldsymbol{w}_o||^2 \tag{2-83}$$

where λ and μ are the stretching and bending moduli, respectively. The total energy of the map is given as

$$U_T = U_A + U_E + U_R \tag{2-84}$$

Essentially, the first term accounts for the fidelity of the data representations, the second term reinforces smoothness of the manifold by favoring similar neighbors and the third term imposes smoothness by stating that a codebook must be similar to the average of its neighboring nodes (Sorzano et al., 2014). The position of the nodes $\{\boldsymbol{w}_k\}$ is given by the *mechanical equilibrium* of the elastic map, or the locations which minimize the total energy U_T. In practice, this is typically solved by the Expectation-Maximization algorithm, which guarantees a local minimum of U_T (Gorban and Zinovyev, 2008).

Unlike the SOM, but similarly to the GNG, elastic nets can add or delete neurons adaptively. This allows them to fit very complex manifolds. The elastic map approach has led

to practical applications in data visualization, data recovery, visualization of genetic texts and recovery of geophysical time series (Gorban and Zinovyev, 2008).

2.2.5 Probabilistic Latent Variable Models (LVM)

The probabilistic interpretation of latent feature learning can be formulated as attempting to discover the generating distribution for the low-dimensional factors of variation (latent random variables) that describe a distribution over the observed, high-dimensional data (Bengio et al., 2012; Murphy, 2012). While many manifold learning and dimensionality reduction models are regarded as defining a projection from a D-dimensional feature space into a lower d-dimensional space, a latent variable model is defined by a mapping *from* the latent space *into* the high-dimensional data space (Bishop et al., 1998). The goal of latent variable models in terms of low-dimensional feature learning is to discover the parameters of an invertible mapping from the latent generating distribution to the high-dimensional data distribution which maximize the likelihood of the observed data.

Probabilistic methods provide two possible modeling paradigms for inferring latent variables, directed and undirected graphical models, which differ in their parameterizations of the joint distribution $p(\boldsymbol{x}_n, \boldsymbol{z}_n)$. The remainder of this section focuses on directed graphical models which follow a particular formulation called *Factor Analysis*.

2.2.5.1 Factor Analysis (FA) and Probabilistic PCA (PPCA)

Factor Analysis (FA) is a special case of a directed latent variable model that assumes the distribution of high-dimensional random variables is induced by a generalized linear mapping of the latent random variables (Murphy, 2012; Tipping and Bishop, 1999; Rish et al., 2008; Gnen, 2013). A common assumption is that the high-dimensional data are Gaussian distributed and the the conditional likelihood is given as

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \tag{2-85}$$

where $\boldsymbol{W}$ is a $D \times d$ matrix known as the *factor loading matrix* and $\boldsymbol{\Psi}$ is a $D \times D$ covariance matrix. $\boldsymbol{\Psi}$ is assumed to be diagonal since the goal of of the model to to "force" $\boldsymbol{z}$ to explain

the correlation between the data instead of "baking it in" to the covariance. $\boldsymbol{\theta}$ denotes the parameters of the model. Typically a prior is defined which captures any presumed knowledge about the distribution of $\boldsymbol{z}$. The data likelihood conditioned solely on the parameters is obtained by integrating over the latent variables

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \int p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z}) d\boldsymbol{z} \tag{2-86}$$

and the model parameters can be obtained by *maximum likelihood*.

By assuming $\boldsymbol{\Psi} = \beta \mathbb{I}_D$, the factor analysis model provides a probabilistic formulation of PCA (PPCA) (Tipping and Bishop, 1999). A typical choice on this model is a Gaussian-Gaussian conjugate prior pair where the mean of the data likelihood is a linear function of the latent inputs. As an example, the prior over the hidden data representations can be expressed as a Gaussian distribution

$$p(\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{z}_n|\boldsymbol{\mu}_0, \boldsymbol{S}_0) \tag{2-87}$$

and the data likelihood is provided as a multivariate Gaussian (given in Equation 2-85). Classical PCA assumes the data covariance to be $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$ with $\beta \to 0$, thus the model is deterministic. Alternatively, when $\beta > 0$,the projection is no longer orthogonal since it is shrunk toward the prior mean. The trade-off is that the reconstructions of $\boldsymbol{x}_n$ will be closer to the data mean.

While the typical approach for fitting PCA is to use eigen-decomposition or Singular Value Decomposition (SVD), PPCA can be fit with the Expectation-Maximization (EM) algorithm (which may be more computationally efficient for high-dimensional data (Murphy, 2012)). Once the model parameters have been estimated, dimensionality reduction can be performed by applying the inverse mapping

$$\boldsymbol{z}_n = \boldsymbol{W}^T(\boldsymbol{x}_n - \boldsymbol{\mu}) \tag{2-88}$$

The projections onto the discovered principal axes, however, may not be optimal in the mean-squared sense because they will not be orthogonal.

(Murphy, 2012) showed examples on how supervision has been applied to PCA. *Supervised PCA* or *Bayesian factor regression* is a model like PCA, except that the target variable (or label), l_n is taken into account when learning the low-dimensional embedding. For the case of binary classification, the Bayesian model can be decomposed into the following elements:

$$p(\boldsymbol{z}_n) = \mathcal{N}(0, \mathbb{I}_d) \tag{2-89}$$

$$p(l_n|\boldsymbol{z}_n) = \text{Ber}(\text{sigm}(\boldsymbol{w}_l^T \boldsymbol{z}_n)) \tag{2-90}$$

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{W}_x \boldsymbol{z}_n + \boldsymbol{\mu}_x, \beta \mathbb{I}_D) \tag{2-91}$$

where the distribution of a label given a latent sample is defined by a Bernoulli distribution. By placing weighting terms on the components of the likelihood function, Supervised PCA can become discriminative.

Moreover, Gaussian Processes (GPs) have been used to learn nonlinear mappings between the input and latent feature spaces (van der Maaten et al., 2007; Lawrence, 2005), effectively creating nonlinear PPCA. A supervised GP latent factor models for dimensionality reduction was proposed by (Gao et al., 2011).

2.2.5.2 Generative Topographic Mapping (GTM)

The *Generative Topographic Mapping* (GTM) is a nonlinear latent variable model proposed by (Bishop et al., 1998) as alternative to the self-organizing map. Specifically, the GTM provides solutions to known deficencies exhibited by SOMs, such as the lack of an objective function, proofs of convergence, theoretical basis for choosing a learning rate parameter and guarantee of topological ordering. The GTM defines a nonlinear, parametric mapping from a d-dimensional latent space to a D-dimensional data space $\boldsymbol{x} \in \mathbb{R}^D$, where typically $d \ll D$ (Peña et al., 2007). A continuous and differentiable function $f(\boldsymbol{x}; \boldsymbol{W})$ maps

every point in the latent space to a point in the data space. This function can be any arbitrary, parametric mapping such as a feedforward neural network. To learn the parameters, GTM attempts to maximize the negative log-likelihood of the latent coordinates given the high-dimensional data and model parameters. Assuming a probability distribution $p(\boldsymbol{z})$, the GTM derivation begins by assuming each high-dimensional sample $\boldsymbol{x}$ is generated from a nonlinear mapping with additive Gaussian noise

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W}, \beta) = \left(\frac{1}{2\pi\beta}\right)^{D/2} \exp\left(-\frac{1}{2\beta}||f(\boldsymbol{z}; \boldsymbol{W}) - \boldsymbol{x}||^2\right) \tag{2-92}$$

where β is the variance of the radially-symmetric Gaussian function centered on $f(\boldsymbol{z}; \boldsymbol{W})$. The density over the input data space is obtained by integrating over the latent space by

$$p(\boldsymbol{x}|\boldsymbol{W}, \beta) = \int p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W}, \beta) p(\boldsymbol{z}) d\boldsymbol{z} \tag{2-93}$$

which is inherently intractable. By defining $p(\boldsymbol{z})$ as a set of K equally weighted delta functions on a regular grid,

$$p(\boldsymbol{z}) = \frac{1}{K} \sum_{k=1}^{K} \delta(\boldsymbol{z} - \boldsymbol{z}_k) \tag{2-94}$$

the integral turns into a summation

$$p(\boldsymbol{x}|\boldsymbol{W}, \beta) = \frac{1}{K} \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{z}_k, \boldsymbol{W}, \beta) \tag{2-95}$$

which represents the model as a constrained mixture of Gaussians. The data log-likelihood is given by

$$J(\boldsymbol{W}, \beta) = \sum_{n=1}^{N} \ln\left(\frac{1}{K} \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{z}_k, \boldsymbol{W}, \beta)\right) \tag{2-96}$$

Given a set of data $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, the log-likelihood function in Equation 2-96 can be maximized by the Expectation-Maximization algorithm.

There are two obvious limitations of the GTM. First, the computational complexity grows exponentially with the assumed intrinsic dimensionality. As noted by (Peña et al., 2007),

however, this is typically not an issue if the goal is visualization. Additionally, poor initialization can lead the algorithm into local optima. However, the GTM has been used extensively for visualization tasks, clustering and applications in molecular biology (Bishop et al., 1998; Peña et al., 2007; Sorzano et al., 2014; Kegl et al., 2008). Additionally, successful variations of the GTM have been developed, such as the Topographic Product of Experts, Harmonic Topographic Map, Topographic Neural Gas and Inverse-Weighted K-Means Topology-Preserving Map (Peña et al., 2007).

2.2.5.3 Manifold Charting

Charting is the problem of assigning a low-dimensional coordinate system to data points in a high-dimensional sample space. *Manifold Charting*, introduced by (Brand, 2003), constructs low-dimensional data representations by aligning mixtures of factor analyzers (MoFA) or mixtures of probabilistic PCA (MoPPCA) models (van der Maaten et al., 2007). Essentially, manifold charting minimizes a convex cost function measuring the amount of disagreement between the linear models on the global coordinates of the data. This is done in two steps: 1.) computing a mixture of locally linear models on the data and 2.) aligning the models in order to obtain the low-dimensional data representations. Manifold charting begins by performing EM to learn a mixture of factor analyszers in order to obtain M low-dimensional data representations $\boldsymbol{w}_{nm}$ and corresponding responsibilities r_{nm} for each sample $\boldsymbol{x}_n$. The responsibilities r_{nm} describe how much sample $\boldsymbol{x}_n$ belongs to model m, and satisfy $\sum_{m=1}^{M} r_{nm} = 1$. A linear mapping $\boldsymbol{M}$ is found from the local data representations $\boldsymbol{w}_{nm}$ to the global coordinates $\boldsymbol{z}_n$ that minimize the objective

$$J(\boldsymbol{Z}) = \sum_{n=1}^{N} \sum_{m=1}^{M} r_{nm} ||\boldsymbol{z}_n - \boldsymbol{z}_{nm}||^2 \tag{2-97}$$

where $\boldsymbol{z}_n = \sum_{o=1}^{M} r_{no} \boldsymbol{z}_{nm}$ and $\boldsymbol{z}_{nm} = \boldsymbol{w}_{nm} \boldsymbol{M}$. This objective function implies that all models for which a datapoint has a high responsibility should agree on the final coordinate for that datapoint. As shown by (van der Maaten et al., 2007), the cost function can be reformulated as a generalized eigenvalue problem in which the d smallest nonzero eigenvectors form the

linear transformation matrix from the local data representations $\boldsymbol{W}$ to the globally-aligned, low-dimensional data representations $\boldsymbol{Z}$.

Manifold charting has been shown to be more robust to noise than alternative approaches such as Isomap and LLE Brand (2003), however, the main weakness of manifold charting is that fitting the MoFA is susceptible to the presence of local maxima in the log-likelihood function.

2.2.6 High-Dimensional Data Visualization

While not directly related to classification, it is important for the reader to be aware of the current SOA in manifold learning. At the time this document was written, much work was being performed in dimensionality reduction for the visualization of extremely large datasets consisting of hundreds (or thousands) of features. Specifically, the remainder of this section discusses three related, SOA approaches for data visualization. t-SNE, LargeVis and UMAP are compared in terms of their differences in use of manifold learning for the visualization of large, high-dimensional datasets.

2.2.6.1 Stochastic Neighbor Embedding (SNE and t-SNE)

t-Distributed Stochastic Neighbor Embedding was proposed by (van der Maaten and Hinton, 2008) as an extension to SNE, which was developed by Hinton and Roweis in 2006. SNE is intended as a visualization tool for high dimensional datasets. Essentially, SNE finds low-dimensional data coordinates by defining pairwise probabilities between all samples in the input data space. It then assumes that the low-dimensional embedding coordinates should be distributed according to the same conditional probabilities. SNE begins by defining conditional probabilities between all points in the training set which measure their likelihood of being spatial neighbors. Given a datapoint $\boldsymbol{x}_n$, the probability of selecting another point $\boldsymbol{x}_m$ as its neighbor is given as

$$p_{m|n} = \frac{\exp(-||\boldsymbol{x}_n - \boldsymbol{x}_m||^2/2\beta_n)}{\sum_{o \neq p} \exp(-||\boldsymbol{x}_p - \boldsymbol{x}_o||^2/2\beta_n)} \tag{2-98}$$

where β_n is the variance of the RBF kernel centered on sample $\boldsymbol{x}_n$. Therefore, points in a local neighborhood to sample $\boldsymbol{x}_n$ will have a higher probability of being selected as a neighbor. Similarly, the pairwise conditional probabilities between all low-dimensional samples can be defined by

$$q_{m|n} = \frac{\exp(-||\boldsymbol{z}_n - \boldsymbol{z}_m||^2)}{\sum_{o \neq p} \exp(-||\boldsymbol{z}_p - \boldsymbol{z}_o||^2)} \tag{2-99}$$

Since the similarity of a sample with itself is not important, $p_{n|n}$ and $q_{n|n}$ are set to zero. The intuition behind this formulation is that, if the low-dimensional coordinates $\boldsymbol{z}_n$ and $\boldsymbol{z}_m$ correctly model the similarity between the high-dimensional data points $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$, the conditional probabilities $p_{m|n}$ and $q_{m|n}$ will be equal (van der Maaten and Hinton, 2008). Thus, it is intuitive that the Kullback-Leibler (KL) divergence (which measures the cross-entropy between probability densities) would make an appropriate cost function between the high and low-dimensional probabilities. The objective function of SNE is given by

$$J = \sum_{n=1}^{N} KL(P_n||Q_n)) = \sum_{n=1}^{N} \sum_{m=1}^{N} p_{m|n} \log \frac{p_{m|n}}{q_{m|n}} \tag{2-100}$$

where P_n denotes the conditional probability distribution over all other high-dimensional datapoints given $\boldsymbol{x}_n$, and Q_n represents the conditional probability distribution over all other low-dimensional embedding coordinates given $\boldsymbol{z}_n$. Given that the KL divergence is asymmetric, it can be observed that using widely separated points in the embedding space to model points close in the input space accrues a large error. However, the cost of using nearby points in the embedding space to represent points far in the data space is small. Therefore, the SNE cost focuses on retaining the local structure of data in the embedding space. This problem can be optimized using gradient descent.

A problem with is that, due to the curse of dimensionality, points in "middle distances" get squashed closely together in the low-dimensional embedding space. This is known as the *crowding problem*, and stems from the fact that in high-dimensional spaces, the surface of the a hypersphere (Guassian distribution) grows much more quickly with its radius as

compared to a hypersphere in a low-dimensional space. t-SNE handles this problem by forcing the optimization to "spread-out" the medium distance points. This is done by utilizing a symmetric KL-divergence and by giving the low-dimensional density a longer tail. Using a Student t-distribution, the conditional neighbor probabilities of the low-dimensional data coordinates become

$$q_{m|n} = \frac{(1 + ||\boldsymbol{x}_n - \boldsymbol{x}_m||^2)^{-1}}{\sum_{o \neq p} 1 + ||\boldsymbol{x}_p - \boldsymbol{x}_o||^2)^{-1}} \tag{2-101}$$

and optimization of the KL-divergence is performed with gradient descent.

Whereas many manifold learning and dimensionality reduction methods in the literature can only capture global or local data structure, t-SNE is effectively able to capture both. Thus, t-SNE has been applied successfully to 2D and 3D visualization of datasets consisting of millions of samples with hundreds of features (Tang et al., 2016; McInnes et al., 2018). However, t-SNE suffers from a few drawbacks. t-SNE is bottlenecked by the size of the dataset due to the construction of a k-NN graph. Moreover, the efficiency of the graph visualization step deteriorates significantly as the data becomes large and the performance of t-SNE is highly dependent on its hyperparameters. Also, since t-SNE directly optimizes the embeddings, it cannot be applied directly to new data. t-SNE lead as the SOA data visualization tool for approximately ten years after its inception, but because of its deficiencies, LargeVis (Tang et al., 2016) and UMAP McInnes et al. (2018) were eventually proposed.

2.2.6.2 LargeVis

LargeVis was proposed by (Tang et al., 2016) for visualizing large-scale and high-dimensional data in a low-dimensional space (typically 2D or 3D). Specifically, LargeVis addresses the scaling inefficiencies exhibited by t-SNE. LargeVis handles k-NN graph construction by adopting the "a neighbor of my neighbor is my neighbor" approach. Essentially, the algorithm builds a few random projection trees to quickly construct a nearest neighbor graph. Because the resulting graph is likely not very accurate, a search is performed for each node in the graph and neighbors of neighbors are also considered as neighbor candidates. Weights

and edges are provided exactly as done by SNE in Equation 2-98. Once a k-NN graph is constructed, a probabilistic model is used to discover the low-dimensional embedding coordinates which preserve the similarities of samples in the high-dimensional space. The likelihood of the graph is given by

$$\begin{aligned} J &= \prod_{(m,n)\in E} p(e_{mn}=1)^{w_{mn}} \prod_{(m,n)\in E} (1-p(e_{mn}=1))^{\gamma} \\ &\propto \sum_{(m,n)\in E} w_{mn} \log p(e_{mn}=1) + \sum_{(m,n)\in \bar{E}} \gamma \log (1-p(e_{mn}=1)) \end{aligned} \tag{2-102}$$

where $p(e_{mn}=w_{mn}) = p(e_{mn}=1)^{w_{mn}}$ is the likelihood of observing a weighted edge, $\bar{E}$ is the set of vertices that are not observed and γ is a weight assigned to the pairs of vertices without edges between them. Maximizing the first part of Equation 2-102 ensures that similar datapoints in the input data space will be close in the embedding space. The second part of the equation models the likelihood of all vertex pairs without edges. Maximizing this part ensures that dissimilar points are embedded far from each other. The graph likelihood is maximized efficiently by asynchronous stochastic gradient descent.

LargeVis was able to provide 2D and 3D visualizations of millions of data points consisting of hundreds of features from datasets representing hand-written digits, social networks, text documents and images Tang et al. (2016). While LargeVis provides a significant increase in computational efficiency as compared to t-SNE, it still places more importance on preserving local distances, as compared to a combination of local and global.

2.2.6.3 Uniform Manifold Approximation and Projection (UMAP)

Similarly to LargeVis, *Uniform Manifold Approximation and Projection* (UMAP) (McInnes et al., 2018) attempts to address the deficiencies exhibit by t-SNE. Specifically, t-SNE has difficulties working with high dimensional data diectly due to memory limitations and can only practically embed data into two or three dimensions (Tang et al., 2016). UMAP uses manifold approximation and local fuzzy simplicial set representations to construct a topological representation of the high dimensional data. The layout of data in the low dimensional space

is then optimized to minimize the error between the two topological representations. While UMAP may appear wildly different from t-SNE at first glance, there are actually only a few key differences which make it more computationally efficient and scalable.

As with t-SNE (Section 2.2.6.1) and LargeVis (Section 2.2.6.2), UMAP can be described in two phases. A weighted nearest neighbor graph is constructed in the first phase and a mimetic, low-dimensional representation is computed in the second. The differences between t-SNE, LargeVis and UMAP amount to the specific details on how the neighborhood graphs are constructed and how the low-dimensional embeddings are computed.

t-SNE defines the high-dimensional input probabilities by normalizing and symmetrizing RBF kernel similarities, which utilizes Euclidean distance (Equation 2-98). The symmetrized neighbor probabilities are defined by

$$p_{mn} = \frac{p_{m|n} + p_{n|m}}{2N} \tag{2-103}$$

The similarities in the low-dimensional embedding space are then defined by a Student t-distribution with a single degree of freedom on the Euclidean distance (Equation 2-101). The objective function of t-SNE is given as the KL-divergence between the high and low-dimensional probability distributions

$$J_{t-SNE} = \sum_{m \neq n} p_{mn} \log \frac{p_{mn}}{q_{mn}} = \sum_{m \neq n} p_{mn} \log p_{mn} - p_{mn} \log q_{mn} \tag{2-104}$$

Because the computation of both p_{mn} and q_{mn} requires calculation over all pairs of training points, different approaches have been suggested to improve efficiency. LargeVis, for example, considers only the approximate nearest neighbors for each sample and abandons the graph weight normalization. Instead of KL-divergence, LargeVis maximizes the graph likelihood

$$J_{LV} = \sum_{m \neq n} p_{mn} \log w_{mn} + \gamma \sum_{m \neq n} \log \left(1 - w_{mn}\right) \tag{2-105}$$

where p_{mn} and w_{mn} are defined as in Barnes-Hut t-SNE. γ is a parameter which controls the repulsion force (second part of the objective) relative to the attractive force (first part). The

first part of this objective resembles the optimizable portion of the KL-divergence used by t-SNE, except with the substitution of w_{mn} for q_{mn}.

UMAP's cost function can be described by the cross-entropy between v and w

$$J_{UMAP} = \sum_{m \neq n} v_{mn} \log \left(\frac{v_{mn}}{w_{mn}} \right) + (1 - v_{mn}) \log \left(\frac{1 - v_{mn}}{1 - w_{mn}} \right) \tag{2-106}$$

which can be rearranged into two constant terms and two optimizable terms

$$J_{UMAP} = \sum_{m \neq n} v_{mn} \log v_{mn} + (1 - v_{mn}) \log (1 - v_{mn}) - v_{mn} \log w_{mn} - (1 - v_{mn}) \log (1 - w_{mn}) \tag{2-107}$$

and has a similar form to LargeVis without the γ term and the requirement of matrix-wise normalization in the high-dimensional space. The high-dimensional data similarities $v_{m|n}$ in Equation 2-107 are the local fuzzy simplicial set memberships based on smoothed nearest neighbor distances

$$v_{m|n} = \exp \left(\frac{-\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n) - \rho_n}{\beta_n} \right) \tag{2-108}$$

Just as with LargeVis, $v_{m|n}$ is calculated from only the k approximate nearest neighbors and $v_{m|n} = 0$ for all other samples. $\mathcal{D}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ is an arbitrary distance between high-dimensional samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ which is not required to be Euclidean distance. ρ_n is the distance to the nearest neighbor of sample n and β_n controls the bandwidth of the exponential distribution. UMAP leverages ideas of earlier-developed approaches such as Laplacian Eigenmaps (Section 2.2.3.8) which assume that data on the manifold are uniformly distributed. While this constraint can easily be enforced in the embedding space, the training sets of remotely sensed data typically do not exhibit uniform distributions. Because of this, using a fixed metric may leave samples unconnected in the graph. ρ_n plays an important role in that it ensures local connectivity by defining a locally adaptive kernel for each datapoint which allows the metric to vary between each point.

Symmetrization is performed by the fuzzy set union using the probabilistic t-conorm

$$v_{mn} = (v_{m|n} + v_{n|m}) - v_{m|n}v_{n|m} \tag{2-109}$$

and the low dimensional data similarities are given by

$$w_{mn} = (1 + a||\boldsymbol{z}_m - \boldsymbol{z}_n||_2^{2b})^{-1} \tag{2-110}$$

where a and b are user-defined hyper-parameters. Setting $a = b = 1$ results in the Student t-distribution used by t-SNE. With all terms defined, the objective function in Equation 2-107 can be optimized efficiently with stochastic gradient descent.

Experiments by (McInnes et al., 2018) showed that UMAP was both faster than t-SNE (due to efficient neighbor search and lack of normalization) and was capable of preserving more global data structure. While the KL-divergence cost used in t-SNE heavily penalizes mapping close points in the high-dimensional space with far points in the low-dimensional space, it is less concerned with mapping points far in the high-dimensional space with samples close in the embedding space. This reveals that t-SNE places emphasis on locality preservation but does not guarantee that global structure will be retained through the embedding. UMAP, on the other hand, penalizes for both cases through the use of a cross-entropy cost function. This effectively allows UMAP to retain (with trade-offs) both local and global data structure. Additionally, whereas t-SNE which uses random initialization, UMAP initializes the low-dimensional data coordinates through the Laplacian Eigenmaps algorithm which assumes uniformity in the embedding space. Removal of initialization randomness tends to make UMAP embeddings more repeatable. All in all, UMAP was shown to be competitive with the SOA t-SNE and LargeVis algorithms for data visualization tasks while providing improved speed and flexibility. UMAP has been used for visualization of data in bioninformatics, materials science and machine learning (McInnes et al., 2018).

2.2.7 Summary of Manifold Learning Algorithms

This literature review discusses a myriad of manifold learning techniques rooted in a variety of fundamental approaches. However, this review barely scratches the surface of proposed techniques. With such an expansive corpus of methods, it is often difficult to distinguish which approaches are "best" for any given application. This is, of course, application dependent and the optimal features for data visualization are likely very different from those for classification. However, each method reviewed demonstrates a unique approach toward learning which brings its own strengths and weaknesses. Table 2-1 provides the fundamental strategies used by the manifold learning algorithms reviewed in this chapter, ranging from classic to SOA approaches. While manifold learning techniques are inherently unsupervised, this review showed that many supervised and semi-supervised methods have been developed from base algorithms. However, the current state of the literature is greatly lacking in manifold learning and dimensionality reduction approaches designed for classification which can effectively address label uncertainty. *In fact, none of the methods reviewed in Section 2.2 are inherently capable of reliably learning useful, discriminative low-dimensional features from weakly-supervised data.* Section 2.3 discusses dimensionality reduction and manifold learning methods designed specifically for discrimination under the weak learning paradigm.

Table 2-1: Overview of manifold learning methods and their properties.

Overview of Unsupervised Manifold Learning		
Linear		
Method	**Strategy**	**Intro.**
PCA	Preserves variance of data, global structure. Section 2.2.2.1	1901
MDS	Preserves pairwise distances, global structure. Section 2.2.2.2	1958

LDA	Minimizes within-class distance and maximizes between-class distance, global structure. Section 2.2.2.4	1936
NMF	Decomposes data into low-dimensional coordinates and basis vectors, minimizes approximation error. Section 2.2.2.3	1994
NPP	Linear variant of LLE, preserves local neighborhoods, local structure. Section 2.2.3.7	2005
LPP	Linear variant of LE, preserves local neighborhoods, local structure. Section 2.2.3.8	2003
Nonlinear		
Method	**Strategy**	**Intro.**
KPCA	Performs PCA in a feature space defined by a kernel function, preserves variance of data, global structure. Section 2.2.3.2	1999
MVU	Learns a kernel function to be used with KPCA that "unfolds" the manifold, preserves variance of data, global structure. Section 2.2.3.6	2004
Isomap	Preserves geodesic distances, global structure. Section 2.2.3.4	2000
Sammon Mapping	Preserves pairwise distances. More emphasis is placed on samples that are close in the input space. Section 2.2.3.5	1969
LLE	Preserves local neighborhoods, local structure. Section 2.2.3.7	2000
Laplacian Eigenmaps	Preserves local neighborhoods, local structure. Section 2.2.3.8	2003

Hessian LLE	Minimizes curvature of high-dimensional manifold estimated from local neighborhoods. Section 2.2.3.9	2003
LTSA	Learns manifold in local tangent spaces then aligns to form global coordinate system. Section 2.2.3.10	2002
Diffusion Maps	Preserves diffusion distance. Section 2.2.3.11	2006
Neural Networks		
Method	**Strategy**	**Intro.**
Autoencoders	Neural network where desired output is the input. Network decreases in dimensionality then increases to input size. Effectively performs nonlinear PCA. Section 2.2.4.1	1986
SOM	Distributes fixed lattice of neurons to cover input feature space using competitive learning. Section 2.2.4.2	1990
GNG	Distributes lattice of neurons to cover input feature space using competitive learning. The network is able to add and delete neurons to fill the data space. Section 2.2.4.3	1994
Elastic Maps and Nets	Uses map of neurons to cover the input feature space. Simulates network of springs and is optimized to the mechanical equilibrium of the system. Section 2.2.4.4	2008
Probabilistic		
Method	**Strategy**	**Intro.**
PPCA	Learns parameters of linear mapping from latent space to input space by maximizing data likelihood of a factor analysis model. Section 2.2.5.1	1999

GTM	Learns a mapping from the latent space to the data space by assuming the data distribution as a constrained mixture of Gaussians. Probabilistic alternative to the SOM. Section 2.2.5.2	1998
Manifold Charting	Computes mixture of locally linear models on the data then aligns local low-dimensional models to find global coordinates. Section 2.2.5.3	2003
	High-Dimensional Data Visualization	
t-SNE	Defines pairwise probabilities of samples being neighbors. Minimizes the KL-divergence between high and low-dimensional data densities. Retains global and local data structure. Section 2.2.6.1	2008
LargeVis	Efficiently constructs neighborhood graph weighted by conditional probabilities of being neighbors. Maximizes the graph likelihood in the embedding space. Preserves local distances. Section 2.2.6.2	2016
UMAP	Uses fuzzy simplicial sets to define locally adaptive neighborhood graph. Minimizes the cross-entropy between the high and low-dimensional similarity distributions. Preserves global and local structure. Section 2.2.6.3	2018

2.3 Weakly Supervised Manifold Learning and Dimensionality Reduction

Although the specific feature vectors being used in remote sensing applications can be very high-dimensional, the underlying structure of a given dataset set is usually governed by only a few variables. Either implicitly or explicitly, most learning algorithms exploit this underlying structure to make learning and inference possible. If it is available, relevant information

for a specific task can generally be incorporated to provide supervision and improve the performance of unsupervised methods. Supervised methods for nonlinear dimensionality reduction assume that the samples lie on a manifold parameterized by multiple latent factors. However, different from traditional manifold learning (where the goal is to preserve the relationships between samples), these methods find the most discriminative low-dimensional representations for classification tasks (Wu, 2015). The optimal embedding uses class label information to minimize distances between nearby points with the same class label while separating samples of different classes. Although supervised manifold learning often outperforms unsupervised methods for classification tasks, this learning cannot be done directly in MIL because of the uncertainty on the labels (Carbonneau et al., 2016). Moreover, fully-annotated samples are often difficult or impossible to obtain in many remote sensing applications (Zare et al., 2018). Even with the successes of manifold learning, most the of the previous work has mainly focused on either fully supervised or unsupervised learning. Existing work in weakly supervised learning on manifolds has primarily considered the semi-supervised setting (Z. Zhang et al., 2008; Chen et al., 2018; Zhang et al., 2014; Hong et al., 2019; Navaratnam et al., 2007; Stanley III et al., 2018; Tuia and Camps-Valls, 2015; Wang and Mahadevan, 2010, 2011). Methods in this category usually incorporate partially provided image labels and propagate the labels over the manifold approximated by the neighborhood graph on the images. In a broad sense, however, different situations of weak supervision (specifically, Multiple Instance Learning), have not been well studied (Wu, 2015). The existing MIL manifold learning in the literature can be broken into two paradigms: LDA-based approaches and sparse, orthogonal matrix-based techniques (Zhu et al., 2018). Thus, all existing approaches are linear, meaning they may not work well if the underlying bag manifold exhibits curvature. Alternative weakly-supervised dimensionality reduction approaches have been proposed, however, they do not adhere to the constraints of MIL. The current literature for weakly supervised dimensionality reduction is reviewed in the following sections, beginning with reviews of MIL DR approaches, namely: MIDR, MidLABS,

CLFDA, MIDA and MI-FEAR. Approaches using alternative definitions of weak learning are addressed at the end of the section.

2.3.1 MIDR

The first true MIL manifold learning experimentation was performed in (Sun et al., 2010) under the orthogonal matrix-based paradigm. To show the need for MIL-specific methods, Sun et al. showed that Principal Component Analysis (PCA) failed to incorporate bag-level label information and thus provided poor separation between positive and negative bags. Additionally, traditional Linear Discriminant Analysis (LDA) was used to project bags into a latent space which maximized between-bag separation, while minimizing within-bag dissimilarity. However, LDA often mixed the latent bag representations due to the uncertainty of negative sample distributions in the positive bags. These results have been shown many times in the literature (Chao et al., 2019; Vural and Guillemot, 2018), so they were to be expected. However, they motivated the need for specialized manifold learning methods that are directly applicable with MIL. Therefore, Sun et al. proposed *Multiple Instance Dimensionality Reduction* (MIDR), which optimizes an objective through gradient descent to discover sparse, orthogonal projection vectors in the latent space in conjunction with the Multiple Instance Logistic Regression classifier. The goal of MIDR is to discover a projection matrix $\boldsymbol{W} \in \mathbb{R}^{D \times d}$ which will increase discriminability between positive and negative bags in the latent embedding space. If given the k^{th} training bag, $\boldsymbol{B}_k = \{\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,n_k}\}$ with corresponding binary bag-level label $L_k \subset \{0, 1\}$, MIDR attempts to find a matrix $\boldsymbol{W}$ such that the projection of $\boldsymbol{B}_k \subset \mathbb{R}^D$ by $\boldsymbol{W}^T\boldsymbol{B}_k \subset \mathbb{R}^d$ increases the separation between positive and negative bags. The intuition is that the probability of the k^{th} bag being positive $\Pr(L_k = 1|\boldsymbol{W}^T\boldsymbol{B}_k)$ should be close to one if it is positive and close to zero otherwise. This can be achieved by minimizing the squared loss between the actual and predicted label of each bag

$$\min_{\boldsymbol{W}} \sum_{k=1}^{K} (\Pr(L_k = 1|\boldsymbol{W}^T\boldsymbol{B}_k) - L_k)^2 \tag{2-111}$$

Taking advantage of the standard assumption, the posterior probability of a bag can be written in terms of the posterior probabilities of its instances

$$\Pr(L_k = 1|\boldsymbol{W}^T\boldsymbol{B}_k) = \max_n \Pr(l_{k,n} = 1|\boldsymbol{W}^T\boldsymbol{x}_{k,n}) \tag{2-112}$$

Equation 2-111 then becomes

$$\min_{\boldsymbol{W}} \sum_{k=1}^{K} (\max_n \Pr(l_{k,n} = 1|\boldsymbol{W}^T\boldsymbol{x}_{k,n}) - L_k)^2 \tag{2-113}$$

From the objective, it is clear that in order to minimize the squared loss, the distances between the key positive instances and all negative instances should be as large as possible. Additionally, $\boldsymbol{W}$ is required to be orthogonal in order to guarantee the resulting latent features are uncorrelated (to remove redundancy) as well as sparse (to improve interpretability). This implies that the new feature representations of instances $\boldsymbol{x}_{kn}$ in bag $\boldsymbol{B}_k$ are formed by linear combinations of the features in the input feature space. The MIDR optimization problem can be written succinctly as defined in (Zhu et al., 2018):

$$\min_{\boldsymbol{W},\boldsymbol{\beta}} f(\boldsymbol{W}, \alpha) + \gamma||\boldsymbol{W}||_1 \quad \text{s.t.} \quad \boldsymbol{W}^T\boldsymbol{W} = \mathbb{I}_d \tag{2-114}$$

where γ is a positive number which controls the balance between the sparsity term $||\boldsymbol{W}||_1$ and the fitting term $f(\boldsymbol{W}, \alpha) = \sum_{k=1}^{K}(P_k(\boldsymbol{W}, \boldsymbol{\beta}) - L_k)^2$. In this case,

$$\begin{aligned} P_k(\boldsymbol{W}, \boldsymbol{\beta}) &= softmax_\alpha(P_{k,1}(\boldsymbol{W}, \boldsymbol{\beta}), \ldots, P_{k,n_k}(\boldsymbol{W}, \boldsymbol{\beta})) \\ &= \frac{\sum_{n=1}^{n_k} P_{k,n} e^{\alpha P_{k,n}(\boldsymbol{W},\boldsymbol{\beta})}}{\sum_{n=1}^{n_k} e^{\alpha P_{k,n}(\boldsymbol{W},\boldsymbol{\beta})}} \end{aligned} \tag{2-115}$$

is the softmax approximation over n of $\max(P_{k,1}(\boldsymbol{W}, \boldsymbol{\beta}), \ldots, P_{k,n_k}(\boldsymbol{W}, \boldsymbol{\beta}))$. A popular way to estimate the posterior probability is by logistic regression

$$P_{k,n}(\boldsymbol{W}, \boldsymbol{\beta}) = \Pr(l_{k,n} = 1|\boldsymbol{W}^T, \boldsymbol{x}_{k,n}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^T\boldsymbol{W}^T\boldsymbol{x}_{k,n})} \tag{2-116}$$

Pseudo-code for MIDR is provided in Algorithm 3.

Algorithm 3 MIDR

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\boldsymbol{L} = \{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, sparsity parameter γ

Output: Projection matrix $\boldsymbol{W}$

```
while Not converged do
    Train multiple instance logistic regression h using data W^T B, L
    for B_k ∈ B do
        P_k ← probability h(B_k)
    end for
    Optimize Equation 2-114 for new W
end while
```

In (Sun et al., 2010), gradient descent was used to optimize the objective. An alternating optimization scheme was employed that switched between estimating the parameters of the MI Logistic Regression and solving for a new sparse, orthogonal embedding matrix. It was found that the newly developed method outperformed unsupervised instance-level dimensionality reduction approaches (applied to bags) for bag-level classification. MIDR was later revisited by Zhu et al. where the optimization problem was reformulated using the *intertial proximal alternating linearized minimization* (iPALM) method (Zhu et al., 2018). The advantage of *Multiple Instance Augmented Lagrangian Multiplier* (MI-ALM) this approach is that the problem variables can be managed sperately and updated effectively. Additionally, the global convergence of MIDR was proved.

2.3.2 MidLABS

The other existing approaches for dimensionality reduction with multiple instance learning follow a LDA scheme. *Multi-Instance Dimensionality reduction by Learning a mAximum Bag margin Subspace* (MidLABS) (Ping et al., 2010) applies LDA to find a projection vector which simultaneously maximizes between-class scattering and minimizes within-class scattering to separate positive and negative bags in the embedding space. While most MIL approaches assume instances are independently and identically distributed (IID), MidLABS represents each bag as a neighborhood graph in order to take advantage of data structure and jointly constructs the scatter matrices by evaluating the scattering between bags. MidLABS optimizes

the following objective:

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T(\sum_{L_i \neq L_j} \boldsymbol{K}_{ij})\boldsymbol{w}}{\boldsymbol{w}^T(\sum_{L_i = L_j} \boldsymbol{K}_{ij})\boldsymbol{w}} \tag{2-117}$$

By defining a bag distance measure, $\boldsymbol{K}$, the the between-class and within-class scatter matrices can be constructed as

$$\boldsymbol{S}_b = \sum_{L_i \neq L_j} \boldsymbol{K}_{ij} \tag{2-118}$$

and

$$\boldsymbol{S}_w = \sum_{L_i = L_j} \boldsymbol{K}_{ij} \tag{2-119}$$

It can be observed that this problem follows LDA exactly, and can thus be solved as the generalized eigenvalue problem:

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}} \tag{2-120}$$

In order to take structural information into account, the customized bag distance measurement is defined by:

$$\boldsymbol{K}_{ij} = \frac{\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} (\boldsymbol{x}_{ia} - \boldsymbol{x}_{jb})(\boldsymbol{x}_{ia} - \boldsymbol{x}_{jb})^T}{n_i n_j} + C \frac{\sum_{c=1}^{m_i} \sum_{d=1}^{m_j} (\boldsymbol{e}_{ic} - \boldsymbol{e}_{jd})(\boldsymbol{e}_{ic} - \boldsymbol{x}_{jd})^T}{n_i^2 n_j^2} \tag{2-121}$$

where $\boldsymbol{x}_{ia}$ is the a^{th} instance in the i^{th} bag, n_i is the total number of instances in the i^{th} bag, $\boldsymbol{e}_{ic}$ is the c^{th} edge in the i^{th} bag and m_i is the total number of edges in the i^{th} bag. This bag distance measurement represents each bag by as an ϵ-graph, where each instance is treated as a node. An edge exists between two nodes if the Euclidean distance between them is lower than a threshold, ϵ. (Latham, 2015). This measurement compares the closeness of bags by summing over all pairs of instances between the bags. Pseudo-code for MidLABS is provided in Algorithm 4.

Algorithm 4 MidLABS

Input: Multiple-instance dataset $\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$

Output: Linear projection vector $\boldsymbol{w}$

1: **for** bag $\boldsymbol{B}_k$ in $\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$ **do**

2: $\quad G_k \leftarrow \epsilon$-graph for $\boldsymbol{B}_k$

3: **end for**

4: Define $\boldsymbol{K}_{ij} = \frac{\sum_{a=1}^{n_i}\sum_{b=1}^{n_j}(\boldsymbol{x}_{ia}-\boldsymbol{x}_{jb})(\boldsymbol{x}_{ia}-\boldsymbol{x}_{jb})^T}{n_i n_j} + C\frac{\sum_{c=1}^{m_i}\sum_{d=1}^{m_j}(\boldsymbol{e}_{ic}-\boldsymbol{e}_{jd})(\boldsymbol{e}_{ic}-\boldsymbol{x}_{jd})^T}{n_i^2 n_j^2}$

5: $\boldsymbol{S}_b = \sum_{L_i \neq L_j} \boldsymbol{K}_{ij}$

6: $\boldsymbol{S}_w = \sum_{L_i = L_j} \boldsymbol{K}_{ij}$

7: Solve the generalized eigenvalue problem $\boldsymbol{S}_b\boldsymbol{w} = \lambda\boldsymbol{S}_w\boldsymbol{w}$

8: Sort eigenvectors $\boldsymbol{w}$ by their eigenvalues λ

9: $\boldsymbol{w} \in \mathbb{R}^{D\times 1} \leftarrow$ top eigenvector

2.3.3 MIDA

In 2014, Chia et al. introduced Multiple-Instance Discriminant Analysis (MIDA) (Chai et al., 2014). MIDA has the same objective as MidLABS, which is to discover a linear projection basis which separates bags in the embedding space. Both MIDA and MidLABS can be considered as MI extensions of LDA. However, the way these algorithms construct their scatter matrices is very different. While MidLABS constructs its scatter matrices at the bag level by directly evaluating the scattering amongst bags, MIDA uses instance-level information to formulate the within-class and between-class scatter. In other words, MIDA selects a prototype for each bag and utilizes the selected instances as bag representatives for constructing the scatter. The mean of all negative instances is used as the negative class prototype. The difficulty with this approach is the ambiguity on which instances are truly positive. The other major difference between MidLABS and MIDA is that MIDA does not consider structural information of data when formulating the scatter matrices. To select candidate positive prototypes, MIDA initializes a set by selecting the most-likely positive instances as those with the lowest density in a Gaussian likelihood estimated by all instances in the negative bags. An iterative optimization

procedure is applied which trades-off between candidate positive instance selection and learning the projection weights into the latent space which maximizes the separation between bags with different labels and minimizes the distances between bags with the same label. Pseudo-code for MIDA is provided in Algorithms 5 and 6.

Algorithm 5 Initialization of MIDA

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, parameter β

Output: Initialized positive prototypes $\boldsymbol{x}^+ = \{\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_{N^+}^+\}$

1: **for** bag $\boldsymbol{B}_k$ in $\boldsymbol{B}^+$ **do**

2: $\quad \boldsymbol{x}_k^+ \leftarrow \arg\min_{\boldsymbol{x}_{kn}} C \sum_{m=1}^{N^-} \sum_{o=1}^{N_m^-} \exp\left(\frac{||\boldsymbol{x}-\boldsymbol{x}_{mo}^-||_2^2}{\beta}\right) \forall n = 1, \ldots, N_k^+$

3: **end for**

Algorithm 6 Projection vector calculation process of MIDA

Input: Multiple-instance dataset $\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, Initialized positive prototypes $\boldsymbol{x}^+ = \{\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_{N^+}^+\}$

Output: Linear projection vector $\boldsymbol{w}$

1: $\boldsymbol{\mu}^+ \leftarrow \frac{1}{N^+} \sum_{k=1}^{N^+} \boldsymbol{x}_k^+$

2: $\boldsymbol{\mu}^- \leftarrow C \sum_{m=1}^{N^-} \sum_{o=1}^{N_m^-} \boldsymbol{x}_{mo}^-$

3: $\boldsymbol{S}_w^+ \leftarrow \sum_{k=1}^{N^+} (\boldsymbol{x}_k^+ - \boldsymbol{\mu}^+)(\boldsymbol{x}_k^+ - \boldsymbol{\mu}^+)^T$

4: $\boldsymbol{S}_w^- \leftarrow \sum_{m=1}^{N^-} (\boldsymbol{x}_m^- - \boldsymbol{\mu}^-)(\boldsymbol{x}_m^- - \boldsymbol{\mu}^-)^T$

5: $\boldsymbol{S}_w \leftarrow \boldsymbol{S}_w^+ + \boldsymbol{S}_w^-$

6: $\boldsymbol{S}_b \leftarrow \sum_{k=1}^{N^+} \sum_{m=1}^{N^-} (\boldsymbol{x}_k^+ - \boldsymbol{x}_m^-)(\boldsymbol{x}_k^+ - \boldsymbol{x}_m^-)^T$

7: Solve the generalized eigenvalue problem $\boldsymbol{S}_b \boldsymbol{w} = \lambda \boldsymbol{S}_w \boldsymbol{w}$

8: Sort eigenvectors $\boldsymbol{w}$ by their eigenvalues λ

9: $\boldsymbol{w} \in \mathbb{R}^{D \times 1} \leftarrow$ top eigenvector of $\frac{\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}$

2.3.4 CLFDA

Finally, Citation Local Fisher Linear Discriminant Analysis (CLFDA) (Saehoon Kim and Seungjin Choi, 2010) incorporates citation and reference information into local Fisher Discriminant Analysis, thus it can also be treated as a MI extension to LDA. Contrary to the previously-mentioned approaches, CLFDA operates at the instance-level, attempting to find a subspace where positive and negative instances are maximally-separable (Latham, 2015). CLFDA can be viewed as complimentary to MIDA (Chai et al., 2014). Whereas, MIDA tries to seek true positive instances in positive bags, CLFDA attempts to detect incorrectly labeled

instances (false positives) in positive bags. CLFDA pre-labels all instances with the label of their bag $l_{kn} = L_k, \forall n = 1, \ldots, n_k$, then utilizes neighborhood information to detect false positive instances. In other words, it uses the assumption that if an instance in a positive bag is close to many points in the negative bags, it is likely also negative. CLFDA defines *references* simply as the nearest neighbors to $\boldsymbol{x}_n$, while $citers$ are defined as the samples which have $\boldsymbol{x}_n$ in their C-nearest neighbors, or $citers(\boldsymbol{x}_n) = \{\boldsymbol{x}_m | \boldsymbol{x}_n \in CNN(\boldsymbol{x}_m), m = 1, \ldots, N\}$. The steps of CLFDA are to 1.) construct a $\max(R, C)$-NN graph, where the $\max(R, C)$-NN graph is a K-NN graph with $K = \max(R, C)$. This is used to detect false positives and re-label them as negative. If the ratio of instances from negative bags versus instances from positive bags in the references and citers of $\boldsymbol{x}_n$ exceeds a threshold τ, then $\boldsymbol{x}_n$ is given a negative label. This is only done for instances from positive bags, as the SMI assumptions states that all instances in negative bags should be negative. 2.) Construct scatter matrices using the provided positive and negative instances. 3.) Find projection weights which simultaneously maximize the distances between positive and negative bags and minimizes the distances between bags with the same class labels using *Local Fisher Discriminant Analysis* (LFDA). LFDA is a version of LDA designed to address multi-modal data distributions. The primary difference between LDA and LFDA is that when maximizing and minimizing the between-class and within-class scatter matrices, respectively, the scatter contribution of a single instance pair is weighted by the locality of the pair. This prevents instances from different clusters sharing the same label from being forced into the same space. This also allows multiple dimensions to be analyzed in the embedding space, whereas LDA is limited to a single dimension for binary classification. Pseudo-code for CLFDA is given in Algorithm 7 (Latham, 2015). A potential downside of CLFDA, however, is that it has been shown that pre-labeling all instances in positive bags as positive often leads to poor results, especially when there are large numbers of negative instances in the positive bags (Chai et al., 2014).

As previously mentioned, all existing MIL dimensionality reduction approaches in the literature are solely linear. However, many modalities exhibit nonlinear variation. Therefore,

Algorithm 7 CLFDA

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, parameters C, R, τ

Output: Projection matrix $\boldsymbol{W}$

$\boldsymbol{X} \leftarrow$ instances in $\boldsymbol{B}$
$G \leftarrow \max(R, C)$-nearest neighbor graph of $\boldsymbol{X}$
for $n = 1 \rightarrow N$ **do**
    $R_n \leftarrow R$-nearest references of instance $\boldsymbol{x}_n$
    $C_n \leftarrow C$-nearest citers of instance $\boldsymbol{x}_n$
    $N_n^- \leftarrow$-number of instances from negative bags in $R_n + C_n$
    $N_n^+ \leftarrow$-number of instances from positive bags in $R_n + C_n$
    if $\frac{N_n^-}{N_n^+} \geq \tau$ or $\boldsymbol{x}_n$ is from a negative bag **then**
        instance label $l_n \leftarrow -1$
    else
        $l_n \leftarrow +1$
    end if
end for
$\boldsymbol{A}^b \leftarrow$ between-class affinity matrix
$\boldsymbol{A}^w \leftarrow$ within-class affinity matrix
Between-class scatter matrix $\boldsymbol{S}_b = \frac{1}{2}\sum_{m,n=1}^{N} \boldsymbol{A}_{mn}^b (\boldsymbol{x}_m - \boldsymbol{x}_n)(\boldsymbol{x}_m - \boldsymbol{x}_n)^T$
Within-class scatter matrix $\boldsymbol{S}_w = \frac{1}{2}\sum_{m,n=1}^{N} \boldsymbol{A}_{mn}^w (\boldsymbol{x}_m - \boldsymbol{x}_n)(\boldsymbol{x}_m - \boldsymbol{x}_n)^T$
$\boldsymbol{W} \in \mathbb{R}^{D \times d} \leftarrow$ top d eigenvectors of $\frac{\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}$

nonlinear manifold learning approaches should be developed which operate under the multiple instance learning framework.

2.3.5 MI-FEAR

While the previously mentioned approaches were based on feature extraction, multiple instance dimensionality reduction has also been investigated under the feature selection paradigm. Specifically, Latham used feature ranking to determine the most important features for instance-level classification (Latham, 2015). Feature ranking considers each feature independently according to a scoring function, thus revealing properties of the individual features by which they may be compared and ranked. The challenge of learning a good feature ranking under an instance-space metric is that the instance labels are not available under the MIL framework. *Multiple-Instance Feature Ranking* (MI-FEAR) assumes one-sided noise by

giving every instance the label of its corresponding bag. This essentially transforms the weakly-supervised problem into a supervised one. Pseudo-code for MI-FEAR is provided in Algorithm 8.

Algorithm 8 MI-FEAR

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\boldsymbol{L} = \{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, evaluation metric $\mathcal{L}$, learning algorithm $\mathcal{A}$, dimensionality of reduced-dimensional space d
Output: $[\theta_i$ for θ_i in $V[1, \ldots, d]]$

1: $\boldsymbol{X}, \boldsymbol{Y}^{\gamma} \leftarrow$ supervised dataset of bag-labeled instances using $(\boldsymbol{B}, \boldsymbol{L})$
2: $V \leftarrow [\,]$
3: **for** feature θ_i in feature set Θ **do**
4: $\quad \boldsymbol{X}^{\theta} \leftarrow \boldsymbol{X}$ with all features, excluding θ_i
5: $\quad h_{\theta} \leftarrow$ output of $\mathcal{A}$ when trained on input $(\boldsymbol{X}^{\theta}, \boldsymbol{Y}^{\gamma})$
6: $\quad V[i] \leftarrow$ performance of h_{θ} on input $(\boldsymbol{X}^{\theta}, \boldsymbol{Y}^{\gamma})$ according to $\mathcal{L}, \theta_i$
7: **end for**
8: $V \leftarrow \textsc{SortDecreasing}(V)$

Each feature θ, is given a score based on the performance of a learned hypothesis h_{θ} operating on a single feature. The performance is determined an evaluation metric $\mathcal{L}$. Once a score has been assigned to every feature, the features are ranked according to relative importance and the top d features are retained as the feature set. As stated in Section 2.3.4, the characteristic accuracy of MI-FEAR is based on noisy labels, where each instance is given its corresponding bag-level label. This approach has proven non-ideal in the literature for determining accurate instance-level classification. However, a benefit of feature selection is that, unlike feature extraction, the reduced-dimensional space retains its interpret-ability. In other words, the representations of individual features do not change, meaning they keep their real-world relevance, if applicable. On the other hand, feature selection requires that a subset of useful features for classification already exist in the training set. Empirical results showed that consistently achieved lower instance-level classification accuracy than CLFDA and MidLABS, but had a significantly lower run-time.

2.3.6 Comparison Table of MI Dimensionality Reduction Methods

Table 2-2 shows a comparison between the multiple instance dimensionality reduction methods reviewed in Section 2.3.

Table 2-2: Summary of multiple instance dimensionality reduction approaches.

Multiple Instance Dimensionality Reduction			
Method	**Summary**	**Reduction Method**	**Classification Level**
MIDR (2010)	Finds a sparse, orthogonal linear projection matrix optimized for bag-level logistic regression. Section 2.3.1.	Linear, orthogonal projection	bag-level
MidLABS (2010)	Finds linear projection vector using LDA defined from bag-similarity kernel. Section 2.3.2.	LDA	bag-level
MIDA (2014)	Learns linear projection vector using LDA defined from bag representative vectors. Section 2.3.3.	LDA	instance-level
CLFDA (2010)	Learns linear projection matrix using local discriminant analysis defined from instance scatter. Instance labels are provided as bag labels and refined. Section 2.3.4.	LFDA	instance-level
MI-FEAR (2015)	Incrementally leaves out feature and evaluates performance loss to provide feature score. Section 2.3.5.	Feature selection	instance-level

2.3.7 General Weak Supervision

Alternative approaches to weakly supervised dimensionality reduction that do not follow the MIL framework have also been proposed. For example, Wu studied weakly supervised

manifold learning for manifold factorization using image-level labels, where the labels were the variation of interest. The primary idea was to use weak labels to find image pairs that should be more similar after removing unwanted image variation. Wu used an alignment method constrained by Hessian regularization to learn a manifold regression, such that new test images would be projected smoothly into a space near neighboring input images (Wu, 2015). Gaur et al. used weakly supervised manifold learning to perform dense semantic object correspondence (Gaur and Manjunath, 2017). The objective of the semantic object correspondence problem is to compute dense association maps for a pair of images such that the same object parts get matched, even for very differently appearing object instances. The goal of Gaur's work was to learn a manifold such that features belonging to the same semantic object parts were projected closer to each other on the manifold. This was achieved by re-purposing deep convolutional features from a classification network, where the labels were weak segmentations of object parts. These features were then projected onto a manifold using LDA. *Hierarchical Agglomerative Clustering* (HAC) was performed in the embedding space and the labels were refined with respect to geodesic distance on the manifold. This method was inspired by the *manifold assumption*, which states that similar objects (even though disparate in the input feature space), should share an intrinsic manifold. In this way, feature embeddings are learned by an optimization process which is rewarded for projecting features closer on the manifold if they have low feature-space dissimilarity. Additionally, the optimization penalizes feature clusters whose geometric structure is inconsistent with the observed geometric structures of object parts.

An alternative approach for discriminative dimensionality reduction with weak labels is through the application of metric embedding, which is discussed in detail in Section 2.4. While they can be fully supervised, metric embedding techniques often consider groups of samples (usually two or three), jointly, to learn an embedding function. Under this type of weak supervision, only a notion of semantic similarity is needed, as compared to direct class labels.

2.4 Metric Embedding

The concepts of "near" and "far" are very powerful and useful utilities in everyday life. They classify the relationship between two "primitives" as being similar or dissimilar, as well as the degree of compatibility (Thorstensen, 2009). As an example, a medical doctor might consider a machine learning researcher and a software engineer as being similar (near), because they both perform research for computer applications. However, the same researcher and engineer would likely consider their jobs as being very disparate (far) based on the details of their work. In order to capture this abstraction of distance in a mathematical construct, a *metric space* is defined.

Definition 2.4.1. Metric Space A *metric space* is an ordered pair $(\mathcal{X}, \mathcal{D})$ where $\mathcal{X}$ is a set and $\mathcal{D}$ is a metric on $\mathcal{X}$, or $\mathcal{D} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$, the following holds:

1. Non-negativity: $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$
2. Identity: $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) = 0 \iff \boldsymbol{x} = \boldsymbol{y}$
3. Symmetry: $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{D}(\boldsymbol{y}, \boldsymbol{x})$
4. Triangle Inequality: $\mathcal{D}(\boldsymbol{x}, \boldsymbol{z}) \leq \mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) + \mathcal{D}(\boldsymbol{y}, \boldsymbol{z})$

The non-negativity rule states that the metric evaluated on two instances must have a positive value or be equal to zero. From the Identity rule, we can see that the metric may only be defined as zero if the two instances being evaluated are exactly the same (thus the dissimilarity is zero). The Symmetry rule states that a metric evaluated between two instances must be the same regardless of ordering. Finally, the Triangle Inequality says, intuitively, that the direct distance between two instances $\boldsymbol{x}$ and $\boldsymbol{z}$ is smaller than the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ plus the distance between $\boldsymbol{y}$ and $\boldsymbol{z}$. Both sides of the inequality will be equal if and only if $\boldsymbol{y}$ lies on the path between $\boldsymbol{x}$ and $\boldsymbol{z}$ on which the metric is defined.

The goal of *metric embedding learning* it to learn a function $f_\theta(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^d$ which maps semantically similar points from the data input feature space of $\mathbb{R}^D$ onto *metrically close* points in $\mathbb{R}^d$. Similarly, f_θ should map semantically different points in $\mathbb{R}^D$ onto metrically distant points in $\mathbb{R}^d$. The function f_θ is parameterized by θ and can be anything ranging from

a linear transformation to a complex non-linear mapping as in the case of deep artificial neural networks (Hermans et al., 2017). Let $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a metric function measuring similarity or dissimilarity in the embedded space. For succinctness, $\mathcal{D}_{m,n} = \mathcal{D}(f_\theta(\boldsymbol{x_m}), f_\theta(\boldsymbol{x_n}))$ defines the dissimilarity between samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, after being embedded. It should be noted that, unlike *metric learning* where the objective is to learn an appropriate metric to measure dissimilarity between samples in opposing classes, metric embedding attempts to learn a transformation function such that samples in the embedding space adhere to a pre-defined measure of similarity (Hermans et al., 2017). Metric embedding is often realized through weakly-supervised learning, where instead of labels, the data is accompanied with sets of preferences.

2.4.1 Ranking Loss

Unlike other loss function such as mean-squared error or cross-entropy whose objective is to directly compare labels, values or sets of values assigned to a given input, the objective of *ranking loss* (also called *margin loss*) functions is to measure relative distances between sets of inputs. To use a ranking loss in a learning scenario, sets of inputs (usually two or three) are embedded through a transformation function into a defined metric space. A metric is used to measure similarity (often Euclidean distance), and the transformation function is updated such that points which have a higher semantic similarity label are embedded more closely, and points which are dissimilar are pushed further apart, according to the metric. In this framework, the actual values of the embedded features are ignored. Instead, only the distances between them matter. While ranking losses have been developed which consider many datapoints at a time (Sohn, 2016), the most popular ranking loss functions in the literature are based on *contrastive* (Koch, 2015b) and *triplet* (Schroff et al., 2015) losses.

2.4.1.1 Contrastive Loss

Let $\boldsymbol{x} \in \mathcal{X}$ be data with corresponding instance-level labels $l \in \{l_1, \ldots, l_N\}$. The superscripts $\boldsymbol{x}^a, \boldsymbol{x}^p, \boldsymbol{x}^n$ are used to denote *anchor*, *positive* and *negative* samples, respectively. The anchor point is the sample of interest, while the positive is a sample of the same class

(or deemed to be similar to the anchor) and the negative sample is a point of a different class (or dissimilar to the anchor). *Contrastive loss* functions take pairs of examples as input and learns and embedding function to predict whether two inputs are from the same class or not. Specifically, contrastive loss can be written as:

$$\begin{aligned}\mathcal{L}^{\alpha}_{\text{cont}}(\boldsymbol{x}_m, \boldsymbol{x}_n; f_\theta) =& \mathbf{1}\{l_m = l_n\}\mathcal{D}(f_\theta(\boldsymbol{x}_m), f_\theta(\boldsymbol{x}_n)) \\ &+\mathbf{1}\{l_m \neq l_n\}\max(0, \alpha - \mathcal{D}(f_\theta(\boldsymbol{x}_m), f_\theta(\boldsymbol{x}_n)))\end{aligned} \tag{2-122}$$

where α is a margin parameter imposing a distance between different classes to be larger than α. Analyzing Equation 2-122, it can be observed that this loss has two unique terms. If the label of $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ are the same (anchor and positive pair), only the first half of the loss is computed. This term is simply the dissimilarity between the samples in the embedding space. Ideally, this term should equate to zero, thus implying the terms are close to each other in the embedding space. If the labels of the samples are opposing (anchor and negative pair), the second term is computed. This equates to the maximum value between zero and the difference between the margin constraint and the dissimilarity between the samples. Thus, if the distance between the anchor and negative is greater than the margin α, the objective is met and the loss equates to zero. Otherwise, the loss is positive. Figure 2-11 demonstrates the idea of contrastive loss. If given a positive anchor pair, the transformation function should embed the samples closer in the embedding space. If given a negative pair, the opposite should occur. It is obvious that with each sample pair the embedding function is only updated in one way (pushing or pulling) (Sohn, 2016; Koch, 2015b).

2.4.1.2 Triplet Loss

Whereas contrastive loss considers pairs of samples, triplet loss jointly optimizes between three samples $\{\boldsymbol{x}^a, \boldsymbol{x}^p, \boldsymbol{x}^n\}$, the anchor, a positive and a negative (Hermans et al., 2017; Schroff et al., 2015; Deng et al., 2019). The fundamental idea behind triplet loss is that, for an input sample, we desire to shorten the distances between its embedded representation and

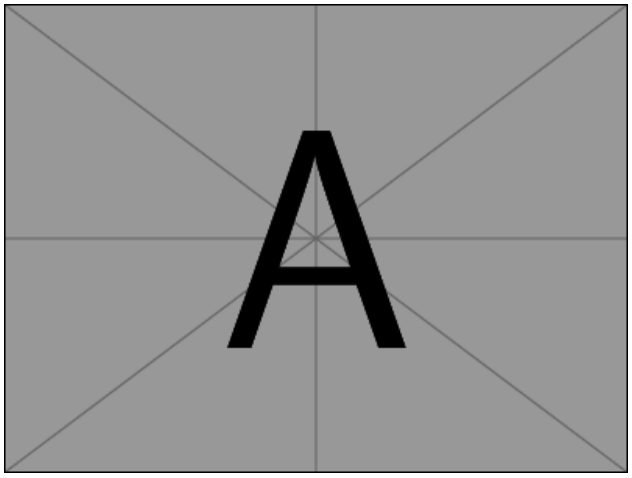


Figure 2-11

those of similar examples, while simultaneously enlarging the distances between dissimilar examples (Sohn, 2016). Triplet loss can be written as:

$$\mathcal{L}^{\alpha}_{\text{tri}}(\boldsymbol{x}^a, \boldsymbol{x}^p, \boldsymbol{x}^n; f_\theta) = \max(0, \mathcal{D}(f_\theta(\boldsymbol{x}^a), f_\theta(\boldsymbol{x}^p)) - \mathcal{D}(f_\theta(\boldsymbol{x}^a), f_\theta(\boldsymbol{x}^n)) + \alpha) \tag{2-123}$$

From Equation 2-123, it can be observed that the triplet loss is satisfied whenever the distance between the negative and anchor is greater than the distance between the corresponding positive sample and the anchor by a margin α. Whereas contrastive loss simply pushes or pulls, triplet loss does both, simultaneously. This is depicted visually by Figure 2-12.

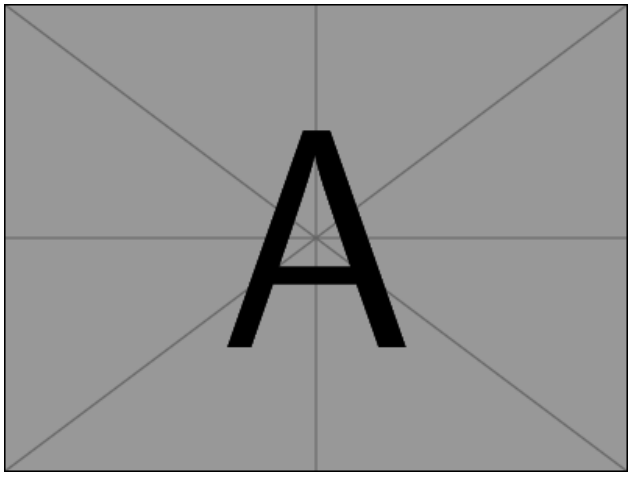


Figure 2-12

The advantage of this formulation is that, while all points of the same class will eventually form a cluster, they are not required to collapse to a single point; they just need to be close to each other than to any point from a different class (Hermans et al., 2017). A major drawback of triplet loss, however, is that as datasets grow larger, the possible number of triplets grows cubically. This can cause practical issues as many of the triplets will likely already satisfy the margin constraints and will not contribute toward learning. Therefore, Schroff et al. argue that *hard-mining* is often imperative for the success of triplet-based methods (Schroff et al., 2015). Essentially, three scenarios exists. *Easy triplets* are triplet sets which already satisfy the margin constraint, thus inducing zero loss. *Hard triplets* are sets where the negative sample is closer to the anchor than the positive. The loss is positive (and greater than α). *Semi-hard triplets* are sets where the negative point is more distant to the anchor than the positive, but it is not greater than the margin α, so the loss is still positive (but smaller than α). Therefore, careful consideration should be taken in triplet construction to maximize the number of hard and semi-hard triplets shown to the learning algorithm.

Alternative approaches such as the ones in (Sohn, 2016), (Deng et al., 2019) and (Xu et al., 2014) use modifications of contrastive or triplet loss to obtain feature representations which adhere to a desired metric embedding.

2.4.2 Large-Margin K-Nearest Neighbors (LMNN)

Weinberger and Saul (Weinberger and Saul, 2009) explored the topic of ranking loss with the explicit goal of performing k-nearest neighbor classification in the learned embedding space (Hermans et al., 2017). This approach is based on the *Large Margin Nearest Neighbor loss* for optimizing f_θ, defined as:

$$\mathcal{L}_{\text{LMNN}}(\theta) = (1-\mu)\mathcal{L}_{\text{pull}}(\theta) + \mu\mathcal{L}_{\text{push}}(\theta) \tag{2-124}$$

which is comprised of a *pull*-term that pulls data points toward its *target neighbors*, or its nearest neighbors from the same class, and a *push*-term that pushes data points from a different class. The term μ is a trade-off parameter used to control the priority of pushing or

pulling. The loss terms are defined as:

$$\mathcal{L}_{\text{pull}}(\theta) = \sum_{m,n} \eta_{mn} \mathcal{D}(f_\theta(\boldsymbol{x}_m), f_\theta(\boldsymbol{x}_n)) \tag{2-125}$$

$$\mathcal{L}_{\text{push}}(\theta) = \sum_{m,n,o} \eta_{mn}(1 - l_{m,o}) \max(0, \mathcal{D}(f_\theta(\boldsymbol{x}_m), f_\theta(\boldsymbol{x}_n)) - \mathcal{D}(f_\theta(\boldsymbol{x}_m), f_\theta(\boldsymbol{x}_o)) + \alpha) \tag{2-126}$$

where $\eta_{mn} \in \{0, 1\}$ is a binary indicator variable used to express whether input $\boldsymbol{x}_n$ is a target neighbor of input $\boldsymbol{x}_m$ and α is a margin enforcing distance between the classes. Analyzing Equations 2-125 and 2-126, it can be observed that the pull term only penalizes large distances between inputs and target neighbors (k-nearest neighbors with the same class label). The push term, however, is comparable to the triplet loss, which enforces that the dissimilarities between the anchor sample $\boldsymbol{x}_m$ and *all negative samples* is greater than the dissimilarities between the anchor and its corresponding target neighbors by at least a margin α. Figure 2-13 demonstrates the pushing and pulling idea captured by LMNN. In the rest of the algorithm, a matrix $\boldsymbol{z}$ is learned which captures an appropriate Mahalanobis metric to measure the dissimilarity between samples, optimized for the LMNN loss.

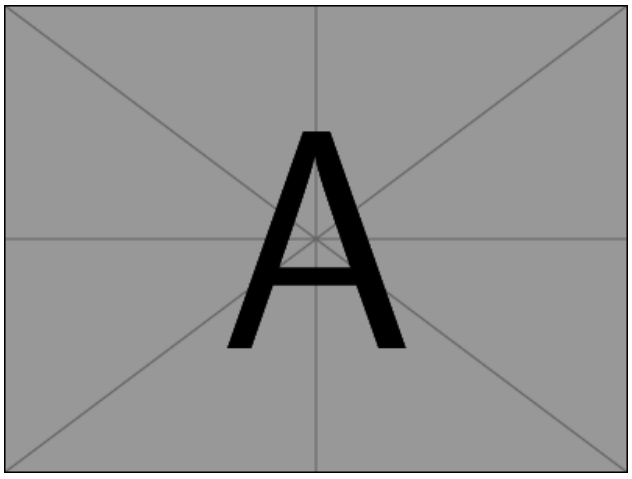
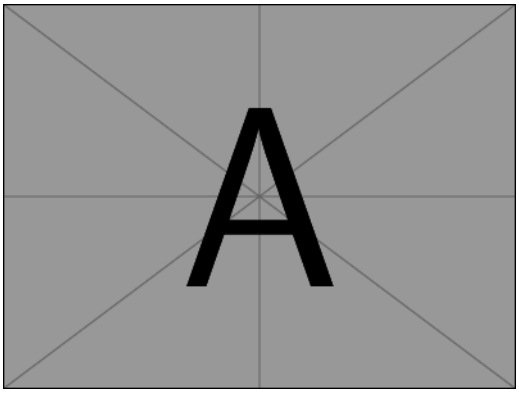


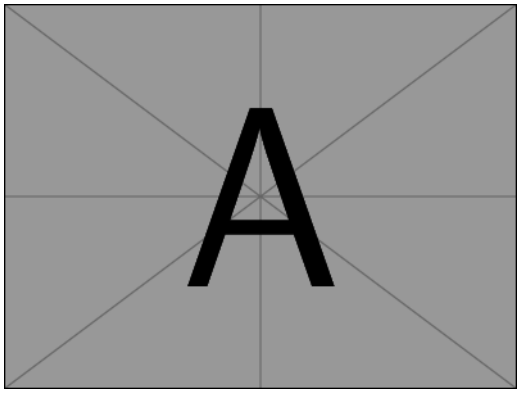
Figure 2-13

2.4.3 Siamese Neural Networks

Siamese Neural Networks are a unique neural architecture proposed by Bromley et al. used to rank similarity between inputs (Bromley et al., 1993). Essentially, Siamese networks use a base architecture to perform feature encoding. This base network is replicated to form twin networks which share parameters (Koch, 2015b,a). The two networks are tied at the output, where a distance metric is used to compare the embeddings of the inputs to each twin. Figure 2-14a demonstrates this idea. The base network can be any architecture (i.e. fully-connected, convolutional, recurrent) so long as it produces a fixed-sized embedding vector at its output. A common implementation is to embed sample pairs using a contrastive loss function (Koch, 2015b). In this scenario, pairs of samples (anchor and positive or anchor and negative) are passed into the twin networks. An embedded representation is produced for each sample, and a dissimilarity metric (typically L_2 or L_1 norm) is applied to compare the samples. Thus, the actual values of the features do not matter, and only the relative distance between them in the embedding space is compared. Following the idea of contrastive loss, pairs with the same label should be embedded close to each other, while contrasting pairs should be mapped further apart in the embedding space. While Siamese networks are traditionally applied as twins, they can be extended to take any number of inputs. Another common implementation is a *Siamese Triplet network* (Hoffer and Ailon, 2015). As the name suggests, the base neural architecture is duplicated not once, but twice to form three networks with shared weights. Triplet loss is used to update the network parameters. As demonstrated by Figure 2-14b, the network takes a triplet set of samples as input and an embedded representation is produced for each sample (through the same transformation because the weights are shared). A metric function then compares the dissimilarities between the anchor and positive samples, as well as the anchor and negative. A comparator is applied at the output to compute the triplet loss between the embedded representations. The idea is that samples from the same class or with high semantic similarity should have a lower dissimilarity in the embedding space than samples from opposing classes.



(a) Siamese network structured from contrastive loss.



(b) Siamese network structured from triplet loss.

Figure 2-14

The embeddings produced by Siamese networks have been shown to be powerful representations for discrimination tasks (Chen et al., 2020; Schroff et al., 2015; Koch, 2015b). In test, pairs or sets of images are passed through the contrastive network to simultaneously produce embeddings. The pairing with the lowest dissimilarity would assign the class label to the test point. For example, if classifying a test point into one of ten classes, ten pairs could be passed through the network (the test sample and a representative from each class), and the label of the test point would be assigned as the label of the representative from the best-ranking pair. Alternatively, after all training samples are embedded through the triplet network, an alternative classification scheme such as k-NN or a SVM could be trained on the embedded representations. Testing would simply consist of embedding a test sample through the network and using the test procedure of the external learner.

Chen et al. noted a few interesting observations regarding Siamese networks (Chen et al., 2020). First, data augmentation is often necessary when training Siamese networks to avoid overfitting. Second, representations often benefit from normalization. Finally, contrastive learning benefits from larger batch sizes and longer training times than its supervised counterpart.

Despite the nuances mentioned, Siamese networks have been applied successfully to applications in object recognition (Hoffer and Ailon, 2015), one-shot learning (Koch, 2015b), visual representation learning (Chen et al., 2020) and image retrieval (Hoffer and Ailon, 2015).

2.4.4 FaceNet

FaceNet is a convolutional neural network which learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity (Schroff et al., 2015). The method is based on learning a Euclidean embedding per image using a deep convolutional neural network. Once each sample has been embedded through the network, facial recognition can be achieved through the use of a simple k-NN classifier. FaceNet directly trains its outputs to be a compact 128-dimensional embedding using a triplet-based loss function based on LMNN. FaceNet is based on a Inception network architecture and optimizes a triplet loss formulated as:

$$\mathcal{L} = ||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha, \quad \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T} \tag{2-127}$$

where α is the margin enforced between positive and negative pairs and $\mathcal{T}$ is the set of all possible triplets in the training set. The idea is that the triplet loss does not only promote similar faces to be close to each other in the embedding space, but also enforces a margin between every other face in the dataset. To address the hard-mining problem, triplets can be mined online out of mini-batches. This provides a trade-off between speed and utility toward training. FaceNet is currently SOA for the facial recognition problem, and in response, similar networks have been explored in a variety of applications, including: metric learning, image classification and image retrieval (Hoffer and Ailon, 2015).

CHAPTER 3
TECHNICAL APPROACH

In this chapter, we present the proposed methods for discriminative dimensionality reduction using weakly-supervised learning. In the current state of the literature, few methods have been developed which address the problem of dimensionality reduction and manifold learning using weak labels under the multiple-instance learning framework. Among the proposed methods, none address the possibility of nonlinearity in the underlying manifold representing bags or instances. To this end, this proposal presents a plan of research to fill this gap in the literature. Specifically, we aim to provide a generalized framework for nonlinear dimensionality reduction under the multiple instance learning paradigm which promotes instance-level discriminability in the latent embedding space. This proposal only considers the binary classification problem, which is motivated by target detection in remote sensing. The approaches developed in this work will be directly compared to state-of-the-art approaches in the literature. The remainder of this chapter is divided into three sections. In the first section, we describe the initial work in investigating the efficacy of strictly supervised, nonlinear manifold learning methods for bag and instance-level classification. In the second section, we present high-level summaries of the proposed approaches. Finally, the third section presents the proposed datasets for applying and testing the methods developed.

3.1 Preliminary Work

In order to show potential for the proposed work to advance SOA in the literature, initial experiments were conducted to demonstrate proof of concept for nonlinear dimensionality reduction on bag and instance-level classification. Specifically, two algorithms, Supervised Laplacian Eigenmaps (S-LE) (Raducanu and Dornaika, 2012) and Supervised Enhanced Isomap (SE-Isomap) (Ribeiro et al., 2008), were evaluated across a range of test parameters. While S-LE and SE-Isomap were reviewed in sections 2.2.3.8 and 2.2.3.4, respectively, pseudo-code for the algorithms using Multiple Instance data is provided below.

3.1.1 S-LE Algorithm Description

The objective of S-LE (Raducanu and Dornaika, 2012) is to find low-dimensional representations $\boldsymbol{Z}$ of data $\boldsymbol{X}$, which promote class separability in the embedding space. Essentially, within-class and between class affinity matrices are defined by the nearest neighbors all of samples in the dataset. An objective is constructed using graph Laplacians which trades off between pulling samples of the same class close in the embedding space and pushing away samples of different classes. A detailed discussion of S-LE is given in Section 2.2.3.8 while pseudo-code is provided in Algorithm 9.

Algorithm 9 S-LE

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, parameters k, β, γ, d
Output: Low-dimensional data representations $\boldsymbol{Z}$

$\boldsymbol{X} \leftarrow$ vector representations of bags in $\boldsymbol{B}$
for $m, n \in N$ **do**
  if $\boldsymbol{x}_n$ in k-NN of $\boldsymbol{x}_m$ and $L_m = L_n$ **then**
    $\boldsymbol{W}_{w,mn} \leftarrow \exp\left(-\frac{||\boldsymbol{x}_n - \boldsymbol{x}_m||^2}{\beta}\right)$
    $\boldsymbol{W}_{b,mn} \leftarrow 0$
  else if $\boldsymbol{x}_n$ in k-NN of $\boldsymbol{x}_m$ and $L_m \neq L_n$ **then**
    $\boldsymbol{W}_{w,mn} \leftarrow 0$
    $\boldsymbol{W}_{b,mn} \leftarrow 1$
  else
    $\boldsymbol{W}_{w,mn} \leftarrow 0$
    $\boldsymbol{W}_{b,mn} \leftarrow 0$
  end if
end for
Compute affinity matrix $\boldsymbol{W} \leftarrow \boldsymbol{W}_w + \boldsymbol{W}_b$
Compute graph Laplacians $\boldsymbol{L}_w \leftarrow \boldsymbol{D}_w - \boldsymbol{W}_w$, $\boldsymbol{L}_b \leftarrow \boldsymbol{D}_b - \boldsymbol{W}_b$
Define $\boldsymbol{B} \leftarrow \gamma \boldsymbol{L}_b + (1 - \gamma)\boldsymbol{W}_w$
Solve the eigenvector problem $\boldsymbol{B}\boldsymbol{v} = \lambda \boldsymbol{D}_w \boldsymbol{v}$
Sort eigenvectors $\boldsymbol{v}$ and eigenvalues λ in decreasing order
$\boldsymbol{Z} \in \mathbb{R}^{N \times d} \leftarrow$ top d eigenvectors

3.1.2 SE-Isomap Algorithm Description

SE-Isomap (Ribeiro et al., 2008), as discussed in Section 2.2.3.4, uses a specialized measure of similarity which considers class information in order to project samples of the same

class closer and points in different classes away from each other in the embedding space. SE-Isomap first computes a distance matrix between all pairs of samples in the dataset. Samples from the same class have smaller pairwise distances than samples in different classes, as controlled by the parameter α. After computing the distance matrix $\boldsymbol{W}$, the shortest path distances between every pair of samples are found using Floyd's or Dijkstra's algorithms, and multidimensional scaling is applied to find the low-dimensional data representations $\boldsymbol{Z}$ of input data $\boldsymbol{X}$. Psuedo-code for both SE-Isomap and MDS are provided in Algorithms 10 and 11, respectively.

Algorithm 10 SE-Isomap

Input: Multiple-instance dataset $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K\}$, $\{L_1, \ldots, L_K\}$, $L_k \in \{-1, +1\}$, parameters k, β, α, d

Output: Low-dimensional data representations $\boldsymbol{Z}$

$\boldsymbol{X} \leftarrow$ vector representations of bags in $\boldsymbol{B}$
for $m, n \in N$ **do**
  if $\boldsymbol{x}_n$ in k-NN of $\boldsymbol{x}_m$ and $L_m = L_n$ **then**
    $\boldsymbol{W}(\boldsymbol{x}_m, \boldsymbol{x}_n) \leftarrow \sqrt{1 - \exp \frac{-||(\boldsymbol{x}_m, \boldsymbol{x}_n||^2}{\beta}}$
  else if $\boldsymbol{x}_n$ in k-NN of $\boldsymbol{x}_m$ and $L_m \neq L_n$ **then**
    $\boldsymbol{W}(\boldsymbol{x}_m, \boldsymbol{x}_n) \leftarrow \sqrt{\exp \frac{||(\boldsymbol{x}_m, \boldsymbol{x}_n||^2}{\beta} - \alpha}$
  else
    $\boldsymbol{W}(\boldsymbol{x}_m, \boldsymbol{x}_n) \leftarrow 0$
  end if
end for
$\boldsymbol{D} \leftarrow$ squares of the shortest distances between all points using Dijkstra's or Floyd's algorithm on $\boldsymbol{W}$
$\boldsymbol{Z} \in \mathbb{R}^{N \times d} \leftarrow \mathrm{MDS}(\boldsymbol{D})$

Algorithm 11 MDS

Input: Distance matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$, embedding space dimensionality d

Output: Low-dimensional data representations $\boldsymbol{Z}$

Calculate $\boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}$, where $\boldsymbol{H} = \mathbb{I}_N - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T$ is the centering matrix
Compute eigenvectors and eigenvalues $\boldsymbol{K}v = \lambda v$
$\boldsymbol{V}', \boldsymbol{\Lambda}' \leftarrow \textsc{SortDecreasing}(\boldsymbol{V}, \boldsymbol{\Lambda})$
$\boldsymbol{Z} \in \mathbb{R}^{N \times d} \leftarrow \boldsymbol{V}'\boldsymbol{\Lambda}'^{\frac{1}{2}}$

3.1.3 Instance-Level Classification

For initial bag-level classification experiments, S-LE and SE-Isomap were applied directly as mentioned. These methods were initially developed as strictly supervised approaches, where each sample is associated with a binary class label. Sample-level labels, however, are not available in MIL. In order to initially investigate nonlinear dimensionality reduction for instance level classification, each instance was given the label of its corresponding bag, as was done in CLFDA and mi-SVM (Saehoon Kim and Seungjin Choi, 2010; Andrews et al., 2002), and the instance-level labels were refined through learning. As is done in CLFDA, learning alternated between label refinement and computing the optimal embedding function. As mentioned previously, the label of a bag can be expressed succinctly as the maximum label of its instances, $L_k = \max_{n \in N_k} l_n$, or alternatively, as a set of linear constraints

$$\sum_{n \in N_k} \frac{l_n + 1}{2} \geq 1, \quad \forall k \ni L_k = 1, \quad \text{and} l_n = -1, \quad \forall k \ni L_k = -1 \tag{3-1}$$

These constraints say that if a bag's label is negative, every instance in the bag should have a negative label. Alternatively, the fraction in the first term will evaluate to 1 for an instance label of $+1$ and to 0 for a label of -1. Therefore, the inequality holds so long as at least one instance is labeled positive. Nonlinear dimensionality reduction of the instance-level was performed with S-LE and SE-Isomap as shown in Algorithm **??**. Essentially, each instance was initially given the label of its bag. Learning alternated between label refinement and learning an embedding function which optimized class separation in the embedding space.

3.2 Future Work

This work proposes the development of non-linear dimensionality reduction techniques that facilitate instance-level classification in the embedding space. Preliminary experiments used S-LE and SE-Isomap to embed data for bag and instance-level classification. However, to fully investigate this area, four specific tasks still need to be explored: instance selection, manifold learning, metric embedding and out-of-sample testing.

3.2.1 Instance Selection

Many feature learning approaches in the literature use supervised learning to obtain discriminative feature representations, or use supervised methods to fine-tune unsupervised feature extraction. However, this cannot be done directly in MIL because of the uncertainty on the labels (Carbonneau et al., 2016). Therefore, in order to adapt supervised nonlinear manifold learning to the MIL framework, techniques must be explored to propose the most-likely positive instances. While methods for this have been proposed in the literature (Saehoon Kim and Seungjin Choi, 2010; Maron and Lozano-Pérez, 1998; Bocinsky et al., 2019), I propose to develop an iterative technique to learn the most-likely positive points through training. A variety of techniques will be explored for this task, including: Multiple instance boosting, MI ranking/preference learning, maximum-likelihood, Diverse Density and clustering.

3.2.2 Manifold Learning

I will continue to explore the use of S-LE and SE-Isomap for use in instance-level classification. However, in order to use these methods for optimal instance embedding, notion of which instances are positive and negative is needed. To explore instance embedding, I will first implement DD as a method to select positive and negative prototypes which can be used to optimize the embeddings. Following, I will investigate incorporating a MI boosting or ranking procedure to iteratively adapt the samples which are selected as positive and negative. An iterative approach to training will be used which alternates between instance selection and providing the optimal embeddings for instance-level classification.

3.2.3 Metric Embedding

Metric embedding is often realized through weakly-supervised learning, where instead of labels, the data is accompanied with sets of preferences (Hermans et al., 2017; Koch, 2015b; Schroff et al., 2015). While this embedding is straightforward to implement at the bag level, enforcing instance embeddings is difficult due to the uncertainty on the labels. However, one could (potentially) use instance ranking as discussed in Section 2.1.5 to provide estimated

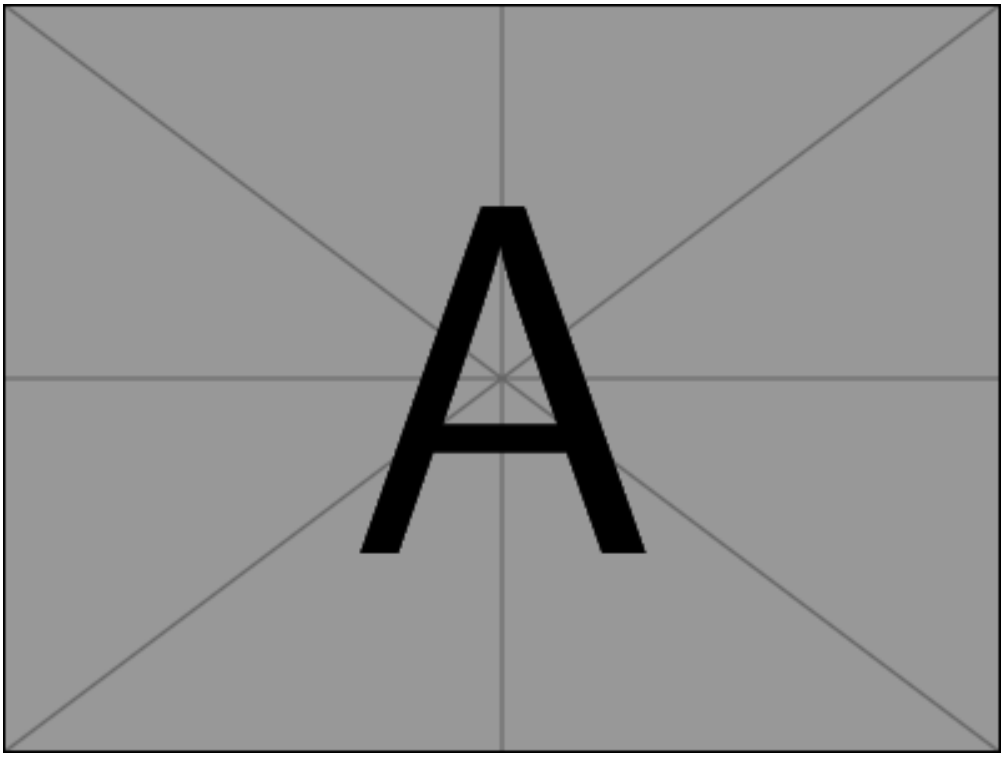


Figure 3-1: Image showing the mapping of instances from bags to a manifold, where the instances are well-separated on the manifold.

preference sets on the instances for use in embedding. In the proposed work, I will investigate MI ranking/preference learning as a way to discriminate between positive and negative instances. Specifically, Siamese networks or ranking SVMs will be trained with contrastive and triplet loss in order to learn low-dimensional features that promote class separation in the embedding space. Bag-level information will be utilized to ensure similarity between instances in negative bags, while enforcing dissimilarity between the instances in positive bags and negative bags. Given a test bag, the algorithm should be able to project negative instances close to other negative instances and far from positive instances in the embedding space (and vice versa) or correctly rank instances as being more likely to be positive or negative.

3.2.4 Out-of-Sample Testing

Preliminary work used simple linear reconstructions of samples' input space neighbors to embed out-of-sample points for testing. However, this approach is likely sub-optimal as it does

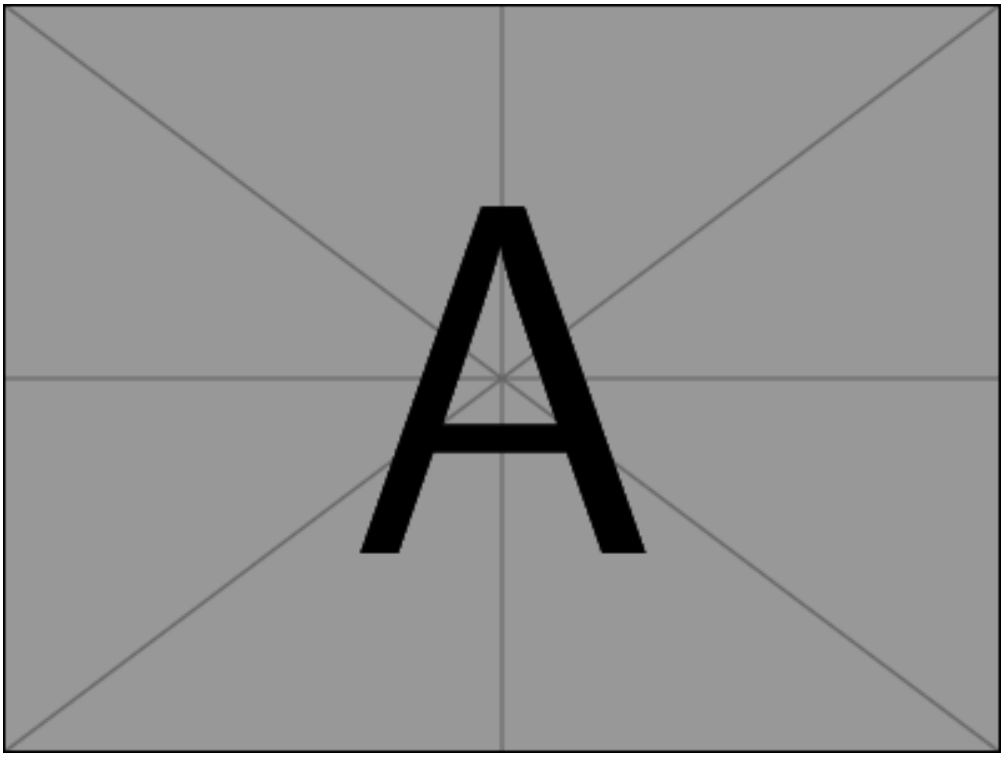


Figure 3-2: Image to show how instances are ranked and are easily separable in a learned metric space.

not utilize training class label information for a non-linear embedding. For this reason, out-of-sample embedding approaches will be explored to project test points into the embedded space. Specifically, I will explore the use of neural networks to learn nonlinear mapping functions between the input and embedding spaces. I will also explore approaches in the literature that consider class information of the training data, such as the technique in (Vural and Guillemot, 2016).

3.2.5 Comparison to SOA

The methods developed through the proposed work will be directly compared to SOA MIL manifold learning and dimensionality reduction methods in the literature. Specifically, bag-level classification will be compared against MIDR (Sun et al., 2010) and MidLABS (Ping et al., 2010), which are inherently suited for bag-level classification. Instance-level classification with be compared against MIDA (Chai et al., 2014), CLFDA (Saehoon Kim and Seungjin Choi, 2010) and MI-FEAR (Latham, 2015). Each approach will be evaluated in terms of

classification performance with k-NN, SVM and a feed-forward neural network on a variety of datasets, including: synthetic manifold learning data, ATR infrared vehicle detection data, hyperspectral target detection data, and people and object classification data. Classification and detection performance will be evaluated with simple classification accuracy and ROC curves (if the classifier produces a continuous classification score).

3.3 Datasets

In this section, a table is given to provide a summary of the datasets proposed to test the developed approaches. The table provides a brief summary of the datasets as well as their intended uses in testing.

Table 3-1: Summary of datasets.

Datasets		
Dataset	**Summary**	**Justification/Use**
Quadratic Surfaces Synthetic	Two sets of synthetic datasets representing 2D quadratic surfaces embedded in 3D. In one set, two quadratic surfaces are separable in the input space. In the other scenario, the surfaces intersect, and are thus inherently non-separable.	Test classification at both the bag and instance level.
DSIAC MS-0003-DB	Publicly available MWIR videos of military vehicles.	Test classification at both the bag and instance level.
MUUFL Gulfport	Hyperspectral, LiDAR and GPS data of scenes taken over University of Mississippi Gulfport.	Test detection and classification at the bag-level.

LFW Faces in the Wild	Images of 5749 different faces labeled at the image level.	Classification and ranking.
PASCAL VOC	2913 RGB images labeled at the pixel-level.	Test instance-level classification.

CHAPTER 4
EXPERIMENTAL DESIGN

In this chapter, a summary is provided of initial experiments used to show proof of concept for the proposed methods for discriminative manifold learning and dimensionality reduction.

4.0.1 Overview

Two approaches were compared in terms of classification accuracy for nonlinear discriminative dimensionality reduction with weak labels. Experiments compared Supervised Laplacian Eigenmaps and Supervised Enhanced Isomap across three parameters of variation: 1.) dimensionality of the embedding space and 2.) feature representation and 3.) amount of label imprecision. Previous work has explored linear methods for dimensionality reduction under the multiple instance learning paradigm (Sun et al., 2010; Ping et al., 2010; Saehoon Kim and Seungjin Choi, 2010). However, no MIL methods have employed a nonlinear manifold learning approach. The objective of this work was to demonstrate proof of concept for discriminative nonlinear dimensionality reduction using weak labels. To prove this point, tests were conducted to explore instance level classification on a set of synthetic 2-D quadratic surfaces embedded in 3-D and to embed image chips abstracted from a real-world remote sensing task of vehicle detection in infrared imagery. Each chip (bag) was provided a positive label if at least one pixel in the image belonged to a vehicle. A negative label was provided if the chip contained only background pixels. The objective was to determine if classification accuracy with a K-nearest neighbor classifier was improved in the embedding space, as compared to the high-dimensional input feature space. The classification performance of each algorithm was tested across a range of embedding dimensionalities using traditional gradient-based features and convolutional neural network features. These tests were repeated for increasing amounts of label imprecision. Experiments are detailed in the following.

4.0.2 Description of Data

4.0.2.1 Quadratic Surfaces

4.0.2.2 DSIAC MS-003-DB

The DSIAC MS-003-DB Algorithm Development Database (DSIAC, 2014) contains a publicly-available collection of mid-wave infrared MWIR imagery collected by the US Army Night Vision and Electronic Sensors Directorate (NVESD). The system for this collection was the L3 Cincinnati Electronics Night Conqueror MWIR imager. The system used a fixed field of view (FOV) 300mm lens, resulting in a 3.4x2.6 FOV. Thus, frames consisted of 640x480 pixels The objective of the data collection was to capture a set of targets at a minimum of 72 unique aspect angles in range steps of 500 meters from 500 to 5000 meters during both day and night. This was achieved by marking a circle with a diameter of approximately 100 meters at each range. Vehicle drivers tracked the circle at around 10 mph. Imagery was taken in real-time for one minute using the MWIR camera at a frame rate of 30 HZ, resulting in approximately 1800 still frames for each target of interest at each range and lighting scenario. The collection used in this work consisted of MWIR video segments of three military and civilian vehicle classes, namely, pickup, SUV and BTR70, at ranges of 1000, 1500 and 2000 meters. To account for sensor noise and lighting variations, each frame was clipped at 0.5% and 98% before applying MAD normalization.

Initial experimentation was performed for bag-level classification. In this work, bags were represented as image chips extracted from the training frames. A chip was given a positive label if it contained at least one pixel on target, and a chip was given a negative label if it contained only background pixels. Figure 4-1 demonstrates the process used for image chip extraction. The level of imprecision was measured by the amount of non-target pixels contained in a bag. Essentially, four levels of imprecision were tested. As seen in Figure 4-2, an uncertainty ratio of 0 denotes that positive bags are constructed from canonical bounding boxes (the tightest possible bounding box on the target), meaning the majority of pixels in the bag lie on target. This scenario can be considered as supervised learning, and is thus used

to provide baseline performance on the dataset. For the remaining levels of label imprecision, chips were extracted in ratios of 51x121 pixels, which corresponds to the largest canonical bounding box present in the dataset across all ranges. Besides increasing the number of background pixels allowed in each bag, the boxes were allowed to shift, so long as at least half of the target was contained in the box. This effectively allowed for potential decrease in the number of pixels on target. Each chip was up-sampled with bicubic interpolation to 510x720 to maintain consistency amongst the dataset. To construct the dataset, 10 background chips and 10 chips containing a portion of the target were extracted from every 50th frame in the videos of interest.

4.0.3 Feature Extraction

Two types of features were explored for high-dimensional bag representation in this work, namely, hand-crafted Histogram of Oriented Gradients (HOG) and features pulled from a convolutional neural network (CNN).

4.0.3.1 HOG Features

Histogram of Oriented Gradients (HOG) features are popular object representations in the computer vision literature. HOG features are essentially descriptors of local object appearance and shape that are characterized by intensity gradients and edge directions (Dalal and Triggs, 2005). In practice, HOG descriptors are extracted by dividing an image into small spatial windows and accumulating local 1-D histograms of gradient directions or edge orientations over the pixels in the windows. The combined histogram representations over all the windows form the combined feature vector for the image. Figure 4-3 demonstrates the HOG gradient information presented in a target and background image chips. It can be observed that the descriptor cues on edge information, which is likely an important feature for the MWIR target detection problem. HOG descriptors were extracted for each chip in the dataset, thus providing 11160-dimensional input features for each bag.

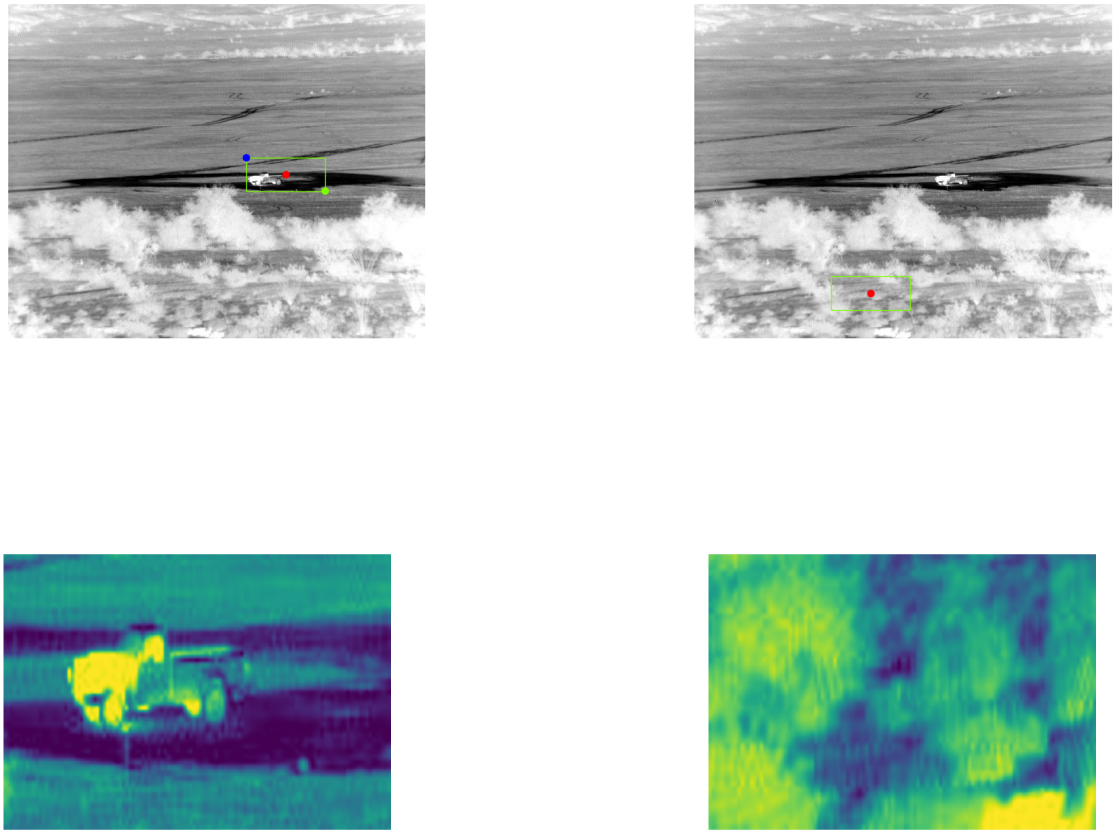
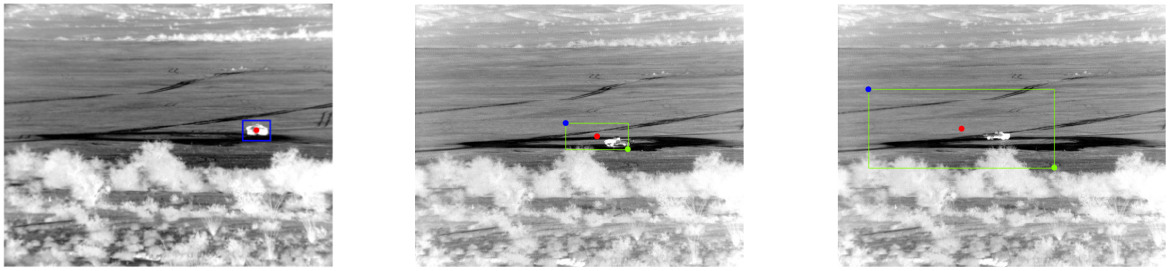


Figure 4-1: The top row shows proposed target (left) and background (right) 51x121 image chips for a label uncertainty ratio of 1. The bottom row shows the corresponding chips after being up-sampled with bicubic interpolation to 510x720. The image containing the truck would be labeled as a positive bag, while the image of foliage would be considered a negative bag.

4.0.3.2 CNN Features

As mentioned in Section **??**, deep learning has recently been explored under the MIL framework. The primary assumption is that networks can extract useful feature representations using only bag-level label information (Ghaffarzadegan, 2018). This is, typically, at the cost of large quantities of data. However, methods for representation learning with deep learning typically result in high levels of classification accuracy (Bengio et al., 2012). Therefore, we tested the dimensionality reduction frameworks with CNN extracted features. To extract



(a) Uncertainty 0 (b) Uncertainty 1 (c) Uncertainty 3

Figure 4-2: Proposed positive image chips as denoted by the green and blue boxes. The left image corresponds to a label uncertainty of 0, which is the canonical bounding box around the target. The middle represents a random target chip of base size 51x121 and the right image is a proposed chip of label uncertainty 3, or 3 times the base chip.

features, we fine-tuned ResNet18 (He et al., 2015) pre-trained on ImageNet (Deng et al., 2009) in PyTorch for 2000 epochs in mini-batches of 20 chips. Image chips were input to the network, which was tasked with predicting the corresponding bag-level labels. After training, the entire dataset was fed through the network and the corresponding convolutional feature maps were extracted directly before the fully connected layers in order to provide every chip a corresponding 512-dimensional feature vector. It should be noted that, while feature vectors were extracted to represent full image chips directly before the fully connected layers, it is possible to extract features from other parts of the network (to potentially represent each pixel). Alternative methods for feature extraction from neural networks will be investigated during the scope of the proposed work.

4.0.4 Algorithm Parameters

Algorithm hyper-parameters were selected by 5-fold cross validation. It was found that for S-LE,

The parameters of SE-Isomap were set at

A K-NN classifier was trained for each test scenario with a fixed $k = 3$ for simplicity.

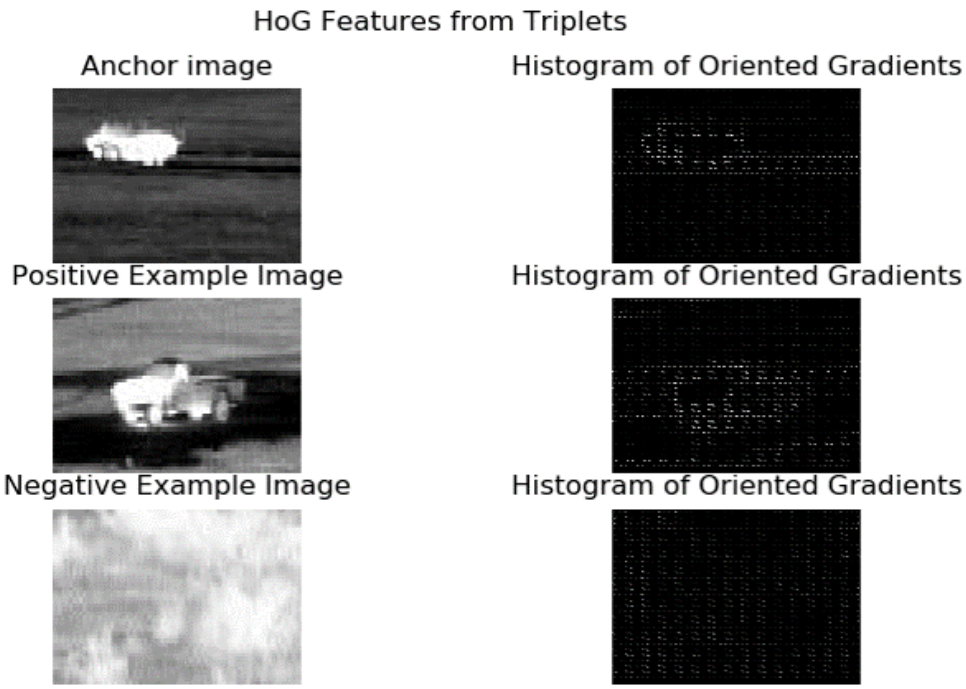


Figure 4-3: Example image chips and corresponding HOG gradient magnitude and direction features over the chip. In the left column, the chips labeled as "Anchor" and "Positive" correspond to positive bags, while th image labeled "Negative" represents a negative bag because it only contains background pixels. The images in the right column demonstrate edge direction and magnitude calculated over 16x16 windows across the images.

4.0.5 Training and Testing Procedures

Training. The data set for each level of uncertainty was split into training and validation folds, where target chips contained vehicles taken at 1000m and 2000m. Each algorithm was tested on a hold-out test set where targets were captured at 1500m. For each level of label imprecision and embedding dimensionality, the nonlinear manifold learning methods were trained in one-shot. Meaning a manifold was constructed from the entire training set in one pass. Hyper-parameters were chosen based on validation performance.

Out-of-Sample Testing. A known difficulty with nonlinear manifold learning methods is the embedding of *out-of-sample* datapoints. These points are samples that were not included in the training dataset. While methods have been developed to perform this embedding which consider class-label information of the training data or learn the embedding function through the universal function approximation abilities of a neural network (Vural and Guillemot, 2016;

Quispe et al., 2016), this work adopts a simple linear reconstruction method for out of sample embedding. Essentially, a low-dimensional coordinate for a test point is found as a linear combination of the low-dimensional representations of its $k = 5$ nearest neighbors in the input feature space. A bias term is also added to improve reconstruction ability.

Evaluation Metrics. Total classification accuracy, or the number of correctly labeled bags divided by the total number of bags, was used to measure the performance of each method during this initial work. In future investigation, however, *Receiver-Operating Characteristic Curves* (ROC) will be utilized if the classifier produces a classification score. A ROC curve is a tool used in binary classification, where the x-axis shows the false-positive rate for detection and the y-axis shows the true-positive rate. True positive rate (TPR) is the measure of true positives over the number of true positive plus false negatives, and false positive rate (FPR) is the number of false negatives over the number of false negatives plus the number of true positives, or one minus TPR. A perfect ROC curve would be a vertical line, where 100% of true positive bags or instances are detected with zero false alarms.

CHAPTER 5
PRELIMINARY RESULTS

In this chapter, a summary of results is provided for the initial experiments using the proposed method for discriminative manifold learning and dimensionality reduction.

CHAPTER 6
FUTURE TASKS

This chapter provides a summary and table of the planned tasks and experiments for the duration of the work proposed.

6.0.1 Overview of Tasks:

This proposal calls for the investigation of manifold learning and dimensionality reduction methods which promote class discriminability and are simultaneously capable of addressing groundtruth imprecision. Two approaches will be explored during this work: Weakly-Supervised Manifold Learning and Weakly-Supervised Metric Embedding. In both areas of investigation, label uncertainty will adhere to the constraints of Multiple Instance Learning. Individual tasks fall under the following categories: 1.) Multiple Instance Manifold Learning, 2.) Multiple Instance Metric Embedding and 3.) Comparison to the literature. General tasks and estimated dates of completion are listed in Section 6.0.2.

6.0.2 Tasks and Approximate Dates of Completion:

Table 6-1: List of research tasks and their corresponding estimated dates for completion.

Tasks	**Approximate Date of Completion**
Multiple Instance Manifold Learning	
Continued on next page	

Table 6-1 – continued from previous page

Tasks	**Approximate Date of Completion**
Develop discriminative MI manifold learning approach for bag and instance-level classification. (**Completed**)	
1. Supervised Nonlinear DR method for bag-level classification	2020-04
(a) Compare S-LE and SE-Isomap with k-NN	
i. Evaluate over a range of embedding dimensions	
ii. Evaluate over varying levels of label uncertainty	
iii. Compare performance on hand-crafted and CNN-based features	
2. Explore methods for out-of-sample embedding	2020-05
(a) Linear Reconstruction (**Completed**)	
(b) Neural Networks	
(c) Methods that consider class information of training data	
3. Investigate methods for selecting positive instances for instance-level embedding/classification	2020-08
(a) Clustering	
(b) Diverse Density-based	
(c) Maximum Likelihood/ KDE of negative instances	
(d) Citation k-NN	
(e) Multiple Instance Boosting	
(f) MI Ranking	
4. Repeat experiments in 1 for instance-level classification	2020-10

Continued on next page

Table 6-1 – continued from previous page

Tasks	**Approximate Date of Completion**
Multiple Instance Metric Embedding	
Develop discriminative MI metric embedding approach for bag and instance-level classification.	
1. Investigate metric embedding approach with Siamese networks.	2021-03
(a) Contrastive loss	
(b) Triplet loss	
(c) Investigate architectures/ training times	
2. Compare performance for both loss functions	2021-05
3. Investigate methods for constructing pairs/triplets from both bags and instances	2021-06
Departmental Requirements	
Status Meeting	2021-04
Initial Dissertation Submission	2022-02-01
Dissertation Defense	2022-03-01
Final Dissertation Submission	2022-03-12

Tasks	**Approximate Date of Completion**
Comparison to SOA	
Continued on next page	

Table 6-2 – continued from previous page

Tasks	**Approximate Date of Completion**
Compare proposed methods to state of the art.	
1. Code SOA MI dimensionality reduction methods, namely: MIDR, MidLABS, MIDA, CLFDA and MI-FEAR	2021-09
2. Compare to SOA MI dimensionality reduction methods in terms of bag-level and instance-level classification performance (with k-NN, SVM and feed-forward neural network), optimal embedding dimensionality and computational complexity.	2021-11
3. Investigate performance on different datasets: DSIAC, Gulfport, LFW, instance-level MIL dataset	2022-01

REFERENCES

A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11 – 26, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.038.

Oludare Abiodun, Aman Jantan, Oludare Omolara, Kemi Dada, Nachaat Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4:e00938, 11 2018. doi: 10.1016/j.heliyon.2018.e00938.

Fabio Aiolli and Alessandro Sperduti. Learning preferences for multiclass problems. 01 2004.

Md. Zahangir Alom, Tarek Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Nasrin, Mahmudul Hasan, Brian Essen, Abdul Awwal, and Vijayan Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8:292, 03 2019. doi: 10.3390/electronics8030292.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81 – 105, 2013. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2013.06.003.

Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James Voss. The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures. *Journal of Machine Learning Research*, 35, 11 2013.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, 15:561–568, 01 2002.

Amina Asif, Wajid Arshad Abbasi, Farzeen Munir, Asa Ben-Hur, and Fayyaz ul Amir Afsar Minhas. pylemmings: Large margin multiple instance classification and ranking for bioinformatics applications, 2017.

Boris Babenko, Zhuowen Tu, and Serge J. Belongie. Simultaneous learning and alignment: Multi-instance and multi-pose learning. 2008.

Boris Babenko, Nakul Verma, Piotr Dollr, and Serge Belongie. Multiple instance learning with manifold bags. pages 81–88, 01 2011.

Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical report, 2005.

John E. Ball, Derek T. Anderson, and Chee Seng Chan Sr. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 11(4):1 – 54, 2017. doi: 10.1117/1.JRS.11.042609.

G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000. doi: 10.1162/089976600300014980.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1):209–239, Jul 2004. ISSN 1573-0565. doi: 10.1023/B:MACH.0000033120.25363.1e.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248632.

Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, Dec. 2002.

C. Bergeron, G. Moore, J. Zaretzki, C. M. Breneman, and K. P. Bennett. Fast bundle algorithm for multiple-instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1068–1079, June 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2011.194.

Charles Bergeron, Jed Zaretzki, Curt Breneman, and Kristin P. Bennett. Multiple instance ranking. In *Proceedings of the 25th International Conference on Machine Learning*, ICML 08, page 4855, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390163. URL https://doi.org/10.1145/1390156.1390163.

C. Bishop, M. Svensn, and C. K. I. Williams. Gtm: The generative topographic mapping. 10: 215–234, January 1998. URL https://www.microsoft.com/en-us/research/publication/gtm-the-generative-topographic-mapping/.

J. Bocinsky. Learning multiple target concepts from uncertain, ambiguous data using the adaptive cosine estimator and spectral match filter. Master's thesis, Univ. of Florida, Gainesville, FL, May 2019.

J. Bocinsky, C. H. McCurley, D. Shats, and A. Zare. Investigation of initialization strategies for the multiple instance adaptive cosine estimator. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV, 110120N*, volume 11012 of *Proc.SPIE*, May 2019. doi: 10.1117/12.2519463.

Matthew Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15*, pages 961–968. MIT Press, 2003.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199 – 231.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS93, page 737744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1053-5888. doi: 10.1109/MSP.2017.2693418.

L. Cao, F. Luo, L. Chen, S. Yihan, H. Wang, C. Wang, and R. Ji. Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning. *Pattern Recognition*, 64, 12 2016. doi: 10.1016/j.patcog.2016.10.033.

M. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *CoRR*, abs/1612.03365, 2016. URL http://arxiv.org/abs/1612.03365.

Jing Chai, Xinghao Ding, Hongtao Chen, and Tingyu Li. Multiple-instance discriminant analysis. *Pattern Recognition*, 47(7):2517 – 2531, 2014. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2014.02.002". URL http://www.sciencedirect.com/science/article/pii/S0031320314000387.

G. Chao, Y. Luo, and W. Ding. Recent advances in supervised dimension reduction: A survey. *Machine Learning and Knowledge Extraction*, 1(1):341–358, 2019. ISSN 2504-4990. doi: 10.3390/make1010020.

B. Chaudhuri and S. Parui. Target detection: Remote sensing techniques for defence applications. *Defence Science Journal*, 45:285–291, 04 1995. doi: 10.14429/dsj.45.4135.

Junfen Chen, Bojun Xie, Hui Zhang, and Junhai Zhai. *Deep Autoencoders in Pattern Recognition: A Survey*, chapter Chapter 9, pages 229–255. 2019. doi: 10.1142/9789813143180_0009.

M. Chen, J. Wang, X. Li, and X. Sun. Robust semi-supervised manifold learning algorithm for classification. *Mathematical Problems in Engineering*, 2018:1–8, 02 2018. doi: 10.1155/2018/2382803.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

V. Cheplygina, M. Bruijne, and J. P. W. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280 – 296, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.03.009.

J. Chiang and P. D. Gader. Hybrid fuzzy-neural systems in handwritten word recognition. *IEEE Transactions on Fuzzy Systems*, 5(4):497–510, Nov 1997. ISSN 1063-6706. doi: 10.1109/91.649901.

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML 05, page 137144, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi:

10.1145/1102351.1102369.

Chun-Guang Li and Jun Guo. Supervised isomap with explicit mapping. In *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, volume 3, pages 345–348, Aug 2006. doi: 10.1109/ICICIC.2006.530.

Ronald R. Coifman and Stphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2006.04.006. URL http://www.sciencedirect.com/science/article/pii/S1063520306000546. Special Issue: Diffusion Maps and Wavelets.

M. Cook. Task driven extended functions of multiple instances (td-efumi). Master's thesis, Univ. of Missouri, Columbia, MO, 2015.

M. Cook, A. Zare, and D. K. C. Ho. Buried object detection using handheld wemi with task-driven extended functions of multiple instances. In *Proc. SPIE 9823, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXI, 98230A*, volume 9823 of *Proc. SPIE*, pages 9823 – 9823 – 9, Apr. 2016. doi: 10.1117/12.2223349.

B. Dai, Y. Wang, J. Aston, G. Hua, and D. Wipf. Hidden talents of the variational autoencoder, 2017.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) - Volume 1 - Volume 01*, CVPR 05, page 886893, USA, 2005. IEEE Computer Society. ISBN 0769523722. doi: 10.1109/CVPR.2005.177. URL https://doi.org/10.1109/CVPR.2005.177.

Ofer Dekel, Yoram Singer, and Christopher D Manning. Log-linear models for label ranking. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 497–504. MIT Press, 2004.

J. Delaporte, B. M. Herbst, W. Hereman, and S. Van der Walt. An introduction to diffusion maps. 2008.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2019.

Thomas Deselaers and Vittorio Ferrari. A conditional random field for multiple-instance learning. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML10, page 287294, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Anamika Dhillon and Gyanendra Verma. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 12 2019. doi: 10.1007/s13748-019-00203-0.

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Prez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31 – 71, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(96)00034-3.

David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10): 5591–5596, 2003. doi: 10.1073/pnas.1031596100.

Defense Systems Information Analysis Center DSIAC. DSIAC MS-003-DB Algorithm Development Database. https://www.dsiac.org/resources/research-materials/cds-dvds-databases-digital-files/atr-algorithm-development-image, 2014.

X. Du. *Multiple Instance Choquet Integral For MultiResolution Sensor Fusion*. PhD thesis, Univ. of Missouri, Columbia, MO, Dec. 2017.

X. Du and A. Zare. Technical report: Scene label ground truth map for muufl gulfport data set. Technical Report 417, University of Florida, Gainesville, FL, April 2017. URL http://ufdc.ufl.edu/IR00009711/00001.

H. Frigui and P. Gader. Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k-nearest neighbor classifier. *IEEE Transactions on Fuzzy Systems*, 17(1), Feb 2009. ISSN 1063-6706. doi: 10.1109/TFUZZ.2008.2005249.

B. Fritzke. A growing neural gas network learns topologies. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, pages 625–632, Cambridge, MA, USA, 1994. MIT Press. URL http://dl.acm.org/citation.cfm?id=2998687.2998765.

K. Fukunaga and J. M. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(6):671–678, Nov 1983. ISSN 1939-3539. doi: 10.1109/TPAMI.1983.4767461.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, *Machine Learning: ECML 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell. Muufl gulfport hyperspectral and lidar airborne data set. Technical Report 570, University of Florida, Gainesville, FL, October 2013.

M. Gao, A. Li, R. Yu, V. I. Morariu, and L. S. Davis. C-WSL: count-guided weakly supervised localization. *CoRR*, abs/1711.05282, 2017. URL http://arxiv.org/abs/1711.05282.

X. Gao, X. Wang, D. Tao, and X. Li. Supervised gaussian process latent variable model for dimensionality reduction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(2):425–434, April 2011. ISSN 1083-4419. doi: 10.1109/TSMCB.2010.2057422.

Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alexander J. Smola. Multi-instance kernels. In *ICML*, 2002.

U. Gaur and B. S. Manjunath. Weakly supervised manifold learning for dense semantic object correspondence. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1744–1752, Oct 2017. doi: 10.1109/ICCV.2017.192.

Xiurui Geng, Luyan Ji, and Yongchao Zhao. The basic equation for target detection in remote sensing, 2017.

Shabnam Ghaffarzadegan. Deep multiple instance feature learning via variational autoencoder. In *AAAI Workshops*, 2018.

Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Fisher and kernel fisher discriminant analysis: Tutorial. *CoRR*, abs/1906.09436, 2019. URL https://arxiv.org/abs/1906.09436.

Nicolas Gillis. The why and how of nonnegative matrix factorization, 2014.

T. Glenn, A. Zare, P. Gader, and D. Dranishnikov. Bullwinkle: Scoring code for sub-pixel targets. URL https://github.com/GatorSense/MUUFLGulfport/.

M. Gnen. Bayesian supervised dimensionality reduction. *IEEE Transactions on Cybernetics*, 43 (6):2179–2189, Dec 2013. ISSN 2168-2267. doi: 10.1109/TCYB.2013.2245321.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

A. N. Gorban and I. Y. Tyukin. Blessing of dimensionality: mathematical foundations of the statistical physics of data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2118):20170237, 2018. doi: 10.1098/rsta.2017.0237.

A. N. Gorban and A. Y. Zinovyev. Elastic maps and nets for approximating principal manifolds and their application to microarray data visualization. In A. N. Gorban, B. Kégl, D. C. Wunsch, and A. Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 96–130, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-73750-6.

Bruce Hajek. *Random Processes for Engineers*. Cambridge University Press, 2015. doi: 10.1017/CBO9781316164600.

H. Hajimirsadeghi and G. Mori. Multi-instance classification by max-margin training of cardinality-based markov networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1839–1852, Sep. 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2613865.

S. S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Xiaofei He and Partha Niyogi. Locality preserving projections. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS03, page 153160, Cambridge, MA, USA, 2003. MIT Press.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. URL http://arxiv.org/abs/1703.07737.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *Lecture Notes in Computer Science*, page 8492, 2015. ISSN 1611-3349. doi: 10.1007/978-3-319-24261-3_7. URL http://dx.doi.org/10.1007/978-3-319-24261-3_7.

D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu. Learnable manifold alignment (lema): A semi-supervised cross-modality learning framework for land cover and land use classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:193 – 205, 2019. ISSN 0924-2716.

Yang Hu, Mingjing Li, and Nenghai Yu. Multiple-instance ranking: Learning to rank images for image retrieval. pages 1–8, 07 2008. ISBN 978-1-4244-2242-5. doi: 10.1109/CVPR.2008.4587352.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

W. Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K. Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. *CoRR*, abs/1906.03291, 2019. URL http://arxiv.org/abs/1906.03291.

A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411 – 430, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/

S0893-6080(00)00026-5.

M. Ilse, Jakub M. Tomczak, and M. Welling. Attention-based deep multiple instance learning. *CoRR*, abs/1802.04712, 2018. URL http://arxiv.org/abs/1802.04712.

C. Jiao. *Target Concept Learning From Ambiguously Labeled Data*. PhD thesis, Univ. of Missouri, Columbia, MO, Dec. 2017.

Changzhe Jiao, Chao Chen, Ronald G. McGarvey, Stephanie Bohlman, Licheng Jiao, and Alina Zare. Multiple instance hybrid estimator for hyperspectral target characterization and sub-pixel target detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:235 – 250, 2018. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2018.08.012.

Balzs Kegl, Donald Wunsch, and Andrei Zinovyev. *Principal Manifolds for Data Visualisation and Dimension Reduction, LNCSE 58*. 01 2008. ISBN 978-3-540-73750-6.

Gregory Koch. *Siamese Neural Networks for One-Shot Image Recognition*. PhD thesis, University of Toronto, Toronto, Canada, 2015a.

Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015b.

T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, Sep. 1990. ISSN 0018-9219. doi: 10.1109/5.58325.

Teuvo Kohonen. *Learning Vector Quantization*, pages 175–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2143–2156, 2007.

Adam R. Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E. Hinton. Stacked capsule autoencoders, 2019.

Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991. doi: 10.1002/aic.690370209. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690370209.

A. C. Latham. Multiple-instance feature ranking. Master's thesis, Case Western Reserve University, Cleveland, OH, August 2015.

N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816, dec 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1194904.

Christian Leistner, Amir Saffari, and Horst Bischof. Miforests: Multiple-instance learning with randomized trees. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 29–42, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

H. Li, D. Liu, and D. Wang. Approximate policy iteration with unsupervised feature learning based on manifold regularization. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, July 2015.

Hongyu Li, Wenbin Chen, and I-Fan Shen. Supervised local tangent space alignment for classification. pages 1620–1621, 01 2005.

Shaohua Li, Yong Liu, Xiuchao Sui, Cheng Chen, Gabriel Tjio, Daniel Shu Wei Ting, and Rick Siow Mong Goh. Multi-instance multi-scale cnn for medical image classification. *Medical Image Computing and Computer Assisted Intervention MICCAI 2019*, page 531539, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-32251-9_58. URL http://dx.doi.org/10.1007/978-3-030-32251-9_58.

Z. Li, W. Shi, X. Shi, and Z. Zhong. A supervised manifold learning method. *Comput. Sci. Inf. Syst.*, 6:205–215, 12 2009.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Constrained graph variational autoencoders for molecule design. *CoRR*, abs/1805.09076, 2018. URL http://arxiv.org/abs/1805.09076.

Weifeng Liu, Jose C. Principe, and Simon Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley Publishing, 1st edition, 2010. ISBN 0470447532.

L. Livi and A. Rizzi. The graph matching problem. *Pattern Anal. Appl.*, 16(3):253–283, Aug 2013. ISSN 1433-7541. doi: 10.1007/s10044-012-0284-8.

E. Lopez-Rubio and E. J. Palomo. Growing hierarchical probabilistic self-organizing graphs. *IEEE Transactions on Neural Networks*, 22(7):997–1008, July 2011. ISSN 1045-9227. doi: 10.1109/TNN.2011.2138159.

L. Ma, M. M. Crawford, and J. W. Tian. Generalised supervised local tangent space alignment for hyperspectral image classification. *Electronics Letters*, 46(7):497 –498, April 2010.

O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pages 570–576, Cambridge, MA, USA, 1998. MIT Press. ISBN 0-262-10076-2.

O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 341–349, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

C. H. McCurley, J. Bocinsky, and A. Zare. Comparison of hand-held wemi target detection algorithms. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV, 110120U*, volume 11012 of *Proc.SPIE*, May 2019. doi: 10.1117/12.2519454.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

M. Meng and X. Zhan. Zero-shot learning via low-rank-representation based manifold regularization. *IEEE Signal Processing Letters*, 25(9):1379–1383, Sep. 2018. ISSN 1070-9908. doi: 10.1109/LSP.2018.2857201.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020, 9780262018029.

R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408976.

M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6338–6347. Curran Associates, Inc., 2017.

E. J. Palomo and E. Lopez-Rubio. Learning topologies with the growing neural forest. *International Journal of Neural Systems*, 26(04):1650019, 2016. doi: 10.1142/S0129065716500192. URL https://doi.org/10.1142/S0129065716500192. PMID: 27121995.

E. J. Palomo and E. Lopez-Rubio. The growing hierarchical neural gas self-organizing neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 28(9):2000–2009, Sep. 2017. ISSN 2162-237X. doi: 10.1109/TNNLS.2016.2570124.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder. *CoRR*, abs/1802.04407, 2018. URL http://arxiv.org/abs/1802.04407.

Yanwei Pang, Lei Zhang, Zhengkai Liu, Nenghai Yu, and Houqiang Li. Neighborhood preserving projections (npp): A novel linear dimension reduction method. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 117–125, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

N. Patwari and A. O. Hero. Manifold learning algorithms for localization in wireless sensor networks. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–857, May 2004.

Marian Peña, Wesam Ashour, and Colin Fyfe. *Topology-Preserving Mappings for Data Visualisation*, volume 58, pages 131–150. 09 2007. doi: 10.1007/978-3-540-73750-6_5.

Wei Ping, Ye Xu, Ren Kexin, Chi-Hung Chi, and Shen Furao. Non-i.i.d. multi-instance dimensionality reduction by learning a maximum bag margin subspace. volume 1, 01 2010.

Jose C. Principe, Neil R. Euliano, and W. Curt Lefebvre. *Neural and Adaptive Systems: Fundamentals through Simulations with CD-ROM*. John Wiley and Sons, Inc., USA, 1st edition, 1999. ISBN 0471351679.

Arturo Mendoza Quispe, Caroline Petitjean, and Laurent Heutte. Extreme learning machine for out-of-sample extension in laplacian eigenmaps. *Pattern Recognition Letters*, 74:68 – 73, 2016. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2016.01.024. URL http://www.sciencedirect.com/science/article/pii/S0167865516000362.

B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432 – 2444, 2012. ISSN 0031-3203.

F. Ratle, G. Camps-Valls, and J. Weston. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48 (5):2271–2282, May 2010. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2037898.

A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, 13(6): 1331–1341, Nov 2002. ISSN 1045-9227. doi: 10.1109/TNN.2002.804221.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

B. Ren, B. Hou, J. Zhao, and L. Jiao. Unsupervised classification of polarimetirc sar image via improved manifold regularized low-rank representation with multiple features. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2):580–595, Feb 2017. ISSN 1939-1404. doi: 10.1109/JSTARS.2016.2573380.

Bernardete Ribeiro, Armando Vieira, and João Carvalho das Neves. Supervised isomap with dissimilarity measures in embedding learning. In José Ruiz-Shulcloper and Walter G. Kropatsch, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 389–396, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

I. Rish, G. Grabarnik, G. A. Cecchi, F. Pereira, and G. J. Gordon. Closed-form supervised dimensionality reduction with generalized linear models. pages 832–839, 01 2008. doi: 10.1145/1390156.1390261.

R. Rojas. Associative networks. In *Neural Networks - A Systematic Introduction*, chapter 12, pages 311–336. Springer-Verlag, Berlin, New-York, 1st edition, 1996.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. ISSN 0036-8075. doi: 10.1126/science.290.5500.2323. URL https://science.sciencemag.org/content/290/5500/2323.

José Francisco Ruiz-Muñoz, Mauricio Orozco-Alzate, and Germán Castellanos-Domínguez. Multiple instance learning-based birdsong classification using unsupervised recording segmentation. In *IJCAI*, 2015.

D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9 (1):75–112, 1985. doi: 10.1207/s15516709cog0901_5. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0901_5.

Saehoon Kim and Seungjin Choi. Local dimensionality reduction for multiple instance learning. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 13–18, Aug 2010. doi: 10.1109/MLSP.2010.5589175.

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. *Kernel Principal Component Analysis*, page 327352. MIT Press, Cambridge, MA, USA, 1999. ISBN 0262194163.

Matthias Scholz, Martin Fraunholz, and Joachim Selbig. Nonlinear principal component analysis: Neural network models and applications. In Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 44–67, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. URL http://arxiv.org/abs/1503.03832.

Nida Shahid, Tim Rappon, and Whitney Berta. Applications of artificial neural networks in health care organizational decision-making: A scoping review. *PLOS ONE*, 14(2):1–22, 02 2019. doi: 10.1371/journal.pone.0212356. URL https://doi.org/10.1371/journal.pone.

0212356.

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.

Vin Silva and Joshua Tenenbaum. Unsupervised learning of curved manifolds. 01 2002. doi: 10.1007/978-0-387-21579-2_31.

Vin D. Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 721–728. MIT Press, 2003. URL http://papers.nips.cc/paper/2141-global-versus-local-methods-in-nonlinear-dimensionality-reduction.pdf.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques, 2014.

J. S. Stanley III, G. Wolf, and S. Krishnaswamy. Manifold alignment with feature correspondence. *CoRR*, abs/1810.00386, 2018. URL http://arxiv.org/abs/1810.00386.

Q. Sun, H. Liu, and T. Harada. Online growing neural gas for anomaly detection in changing surveillance scenes. *Pattern Recognition*, 64:187 – 201, 2017. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2016.09.016. URL http://www.sciencedirect.com/science/article/pii/S0031320316302771.

Yu-Yin Sun, Michael K. Ng, and Zhi-Hua Shou. Multi-instance dimensionality reduction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 587–592. AAAI Press, 2010. URL http://dl.acm.org/citation.cfm?id=2898607.2898702.

R. Talmon, S. Mallat, H. Zaveri, and R. R. Coifman. Manifold learning for latent variable inference in dynamical systems. *IEEE Transactions on Signal Processing*, 63(15):3843–3856,

Aug 2015. ISSN 1053-587X. doi: 10.1109/TSP.2015.2432731.

Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualization large-scale and high-dimensional data. *CoRR*, abs/1602.00370, 2016.

J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. ISSN 0036-8075. doi: 10.1126/science.290.5500.2319. URL https://science.sciencemag.org/content/290/5500/2319.

Alaa Tharwat. Independent component analysis: An introduction. *Applied Computing and Informatics*, 2018. ISSN 2210-8327. doi: https://doi.org/10.1016/j.aci.2018.08.006.

S. Theodoridis and K. Koutroumbas. Kernel pca. In *Pattern Recognition, Fourth Edition*, chapter 6, pages 351–353. Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008. ISBN 1597492728, 9781597492720.

N. Thorstensen. *Manifold learning and applications to shape and image processing*. PhD thesis, Ecole Nationale des Ponts et Chaussees, Paris, France, Nov. 2009.

M. E. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, January 1999. URL https://www.microsoft.com/en-us/research/publication/probabilistic-principal-component-analysis/.

I. W. Tsang and J. T. Kwok. Large-scale sparsified manifold regularization. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1401–1408. MIT Press, 2007. URL http://papers.nips.cc/paper/3005-large-scale-sparsified-manifold-regularization.pdf.

Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning, 2018.

Ming Tu, Jing Huang, Xiaodong He, and Bowen Zhou. Multiple instance learning with graph neural networks, 06 2019.

D. Tuia and G. Camps-Valls. Kernel manifold alignment. 04 2015.

L. van der Maaten, E. Postma, and H. Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research - JMLR*, 10, 01 2007.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. 2008.

Michail Vlachos, Carlotta Domeniconi, Dimitrios Gunopulos, George Kollios, and Nick Koudas. Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 02, page 645651, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775143. URL https://doi.org/10.1145/775047.775143.

E. Vural and C. Guillemot. Out-of-sample generalizations for supervised manifold learning for classification. *IEEE Transactions on Image Processing*, 25(3):1410–1424, March 2016. ISSN 1057-7149. doi: 10.1109/TIP.2016.2520368.

E. Vural and C. Guillemot. A study of the classification of low-dimensional data with supervised manifold learning. *CoRR*, abs/1507.05880, 2018. URL http://arxiv.org/abs/1507.05880.

C. Wang and S. Mahadevan. Multiscale manifold alignment. 09 2010.

C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1541–1546. AAAI Press, 2011. ISBN 978-1-57735-514-4. doi: 10.5591/978-1-57735-516-8/IJCAI11-259. URL http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-259.

Quan Wang. Kernel principal component analysis and its applications in face recognition and active shape models. *CoRR*, abs/1207.3538, 2012. URL http://arxiv.org/abs/1207.3538.

S. Wang, W. Chen, S.M. Xie, G. Azzari, and D.B. Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sens.*, 12:207, 2020.

Yong Wang, Jian-Bin Xie, and Yi Wu. Orthogonal discriminant analysis revisited. *Pattern Recognition Letters*, 84:149 – 155, 2016. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2016.09.010.

Andrew Webb. A kernel approach to metric multidimensional scaling. pages 613–629, 02 2002.

Andrew R Webb and David Lowe. The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis. *Neural Networks*, 3(4):367 – 375, 1990. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(90)90019-H.

David Weinberger. Ai outside in: Machine learning's triangle of error. URL https://accelerate.withgoogle.com/stories/ai-outside.

Kilian Weinberger, Fei Sha, and Lawrence Saul. Learning a kernel matrix for nonlinear dimensionality reduction. 07 2004. doi: 10.1145/1015330.1015345.

Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207244, June 2009. ISSN 1532-4435.

Daniela M. Witten and Robert Tibshirani. Supervised multidimensional scaling for visualization, classification, and bipartite ranking. *Computational Statistics and Data Analysis*, 55(1):789 – 801, 2011. ISSN 0167-9473.

Hui Wu. *Weakly supervised learning on image manifolds*. PhD thesis, University of North Carolina at Charlotte, Charlotte, NC, 2015.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. URL http://arxiv.org/abs/1901.00596.

X. Geng, D. Zhan, and Z. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, Dec 2005. ISSN 1083-4419.

Y. Xiao, B. Liu, and Z. Hao. A sphere-description-based approach for multiple-instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):242–257, Feb 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2539952.

C. Xu, D. Tao, and Y. Rui. Large-margin weakly supervised dimensionality reduction. *31st International Conference on Machine Learning, ICML 2014*, 3:2472–2482, 01 2014.

Y. Xu, W. Ping, and A. T. Campbell.

S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 29(1):40–51, Jan 2007.

Hujun Yin. *Learning Nonlinear Principal Manifolds by Self-Organising Maps*, volume 60, pages 68–95. 09 2007. doi: 10.1007/978-3-540-73750-6_3.

Yixin Chen, Jinbo Bi, and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12): 1931–1947, Dec 2006.

F.-N Yuan, L. Zhang, J.-T Shi, X. Xia, and G. Li. Theories and applications of auto-encoder neural networks: A literature survey. *Jisuanji Xuebao/Chinese Journal of Computers*, 42: 203–230, 01 2019. doi: 10.11897/SP.J.1016.2019.00203.

S. E. Yuksel, J. Bolton, and P. D. Gader. Landmine detection with multiple instance hidden markov models. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, Sep. 2012. doi: 10.1109/MLSP.2012.6349734.

S. E. Yuksel, J. Bolton, and P. Gader. Multiple-instance hidden markov models with applications to landmine detection. *IEEE Transactions on Geoscience and Remote Sensing*, 53(12): 6766–6775, Dec 2015. ISSN 1558-0644. doi: 10.1109/TGRS.2015.2447576.

Z. Zhang, H. Zha, and M. Zhang. Spectral methods for semi-supervised manifold learning. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2008. doi: 10.1109/CVPR.2008.4587381.

A. Zare. *Hyperspectral Endmember Detection and Band Selection Using Bayesian Methods*. PhD thesis, Univ. of Florida, Gainesville, FL, 2008.

A. Zare, M. Cook, B. Alvey, and D. K. Ho. Multiple instance dictionary learning for subsurface object detection using handheld emi. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX, 94540G*, Proc. SPIE, May 2015. doi: 10.1117/12.2179177. URL https://doi.org/10.1117/12.2179177.

Alina Zare, Changzhe Jiao, and Taylor Glenn. Discriminative multiple instance hyperspectral target characterization. *IEEE Trans. Pattern Anal. Mach. Inteli.*, 40(10):2342–2354, Oct. 2018. doi: 10.1109/TPAMI.2017.2756632.

Cha Zhang, John C. Platt, and Paul A. Viola. Multiple instance boosting for object detection. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1417–1424. MIT Press, 2006.

Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1073–1080. MIT Press, 2002.

Tianhao Zhang, Jie Yang, Deli Zhao, and Xinliang Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7):1547 – 1553, 2007. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2006.11.007. URL http://www.sciencedirect.com/science/article/pii/S0925231206004577. Advances in Computational Intelligence and Learning.

Y. Zhang, X. Zheng, G. Liu, X. Sun, H. Wang, and K. Fu. Semi-supervised manifold learning based multigraph fusion for high-resolution remote sensing image classification. *IEEE Geoscience and Remote Sensing Letters*, 11(2):464–468, Feb 2014. ISSN 1545-598X. doi: 10.1109/LGRS.2013.2267091.

Yan Zhang, Zhao Zhang, Jie Qin, Li Zhang, Bing Li, and Fanzhang Li. Semi-supervised local multi-manifold isomap by linear embedding for feature extraction. *Pattern Recognition*, 76: 662 – 678, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.09.043. URL http://www.sciencedirect.com/science/article/pii/S0031320317303977.

Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *CoRR*, cs.LG/0212008, 2002.

Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML 07, page 11671174, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273643. URL https://doi.org/10.1145/1273496.1273643.

Zhi-Hua Zhou and Min-Ling Zhang. Ensembles of multi-instance learners. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, *Machine Learning: ECML 2003*, pages 492–502, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 09, page 12491256, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553534. URL https://doi.org/10.1145/1553374.1553534.

Hong Zhu, Li-Zhi liao, and Michael K. Ng. Multi-instance dimensionality reduction via sparsity and orthogonality. *Neural Comput.*, 30(12):3281–3308, dec 2018. ISSN 0899-7667. doi: 10.1162/neco_a_01140. URL https://doi.org/10.1162/neco_a_01140.