Semi-Supervised Deep Representation Learning

by

VAHID NOROOZI
B.S., Shiraz University, 2008
M.S., Amirkabir University of Technology, 2011

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:

Philip S. Yu, Chair and Advisor
Ajay D. Kshemkalyani
Chris Kanich
Joelle Hallak, Department of Ophthalmology and Visual Sciences
Sihong Xie, Lehigh University

To my parents, my brothers, and my wife, Sara.

ACKNOWLEDGMENTS

First I would like to express my appreciation and gratitude to my advisor Prof. Philip S. Yu, for the guidance and mentorship he provided to me during the past five years. It has been an honor to pursue my Ph.D. degree with him. I am thankful that he gave me the chance to explore a broad range of ideas freely. I appreciate all his contributions of time and help to make my Ph.D. experience productive.

I would like to thank Prof. Joelle Hallak, Prof. Chris Kanich, Prof. Sihong Xie, and Prof. Ajay D. Kshemkalyani for their support and taking their valuable time to serve on my dissertation committee. Additionally, I would also like to thank all my colleagues and friends that I met at the University of Illinois at Chicago, specially Sara, Mehrdad, and Lei. I was really lucky to meet these people and truly appreciate all the great and enjoyable moments that we have had together.

I would like to thank my family. I am truly indebted to my mom and dad for their continued love and support during all of my life. I deeply appreciate my beloved wife, Sara, for her unconditional love and encouragement. We had a wonderful journey together in the last couple of years with all the ups and downs of life. This dissertation would not have been possible without her.

VN

CONTRIBUTION OF AUTHORS

Chapter 2 presents a published manuscript [79] for which I was the primary author. Sara Bahaadini contributed to the revising and drafting some parts of the manuscript. Lei Zheng, Sihong Xie, and Philip S. Yu contributed to the discussions with respect to the work and revising the manuscript.

Chapter 3 presents a published manuscript [78] for which I was the primary author. Sara Bahaadini contributed to the revising and drafting some parts of the manuscript. Lei Zheng, Sihong Xie, and Philip S. Yu contributed to the discussions with respect to the work and revising the manuscript.

Chapter 4 presents a published manuscript [77] for which I was the primary author. Sara Bahaadini contributed to the revising and drafting some parts of the manuscript. Weixiang Shao contributed to data preprocessing. Lei Zheng, Sihong Xie, and Philip S. Yu contributed to the discussions with respect to the work and revising the manuscript.

Chapter 5 presents a work for which I was the primary researcher. Sara Bahaadini contributed to drafting some parts of the manuscript. Nooshin Mojab, Samira Sheikhi, and Philip S. Yu contributed to the discussions with respect to the work and revising its manuscript.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ADULT	UCI Adult Income Dataset
CCA	Canonical Correlation Analysis
ConvNet	Convolutional Neural Network
DCCA	Deep Canonical Correlation Analysis
DDML	Discriminative Deep Metric Learning
DNN	Deep Neural Network
FLDA	Fishers Linear Discriminant Analysis
GAN	Generative Adversarial Networks
KCCA	Kernel Canonical Correlation Analysis
KL-divergence	Kullback-Leibler divergence
KLDA	Kernel Linear Discriminant Analysis
LDA	Linear Discriminant Analysis
LFW	Labeled Faces in the Wild Dataset
MDNN	Multi-view Discriminative Neural Network
MLP	Multilayer Perception
NReLU	Noisy Rectified Linear Units
PCA	Principle Component Analysis

LIST OF ABBREVIATIONS (Continued)

RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Units
RMSProp	Root Mean Square Propagation
SEVEN	Deep SEmi-supervised VErification Networks
SGD	Stochastic Gradient Descent
SONOF	BiosecurID-SONOF Dataset
SVM	Support Vector Machine
TF-IDF	Term FrequencyInverse Document Frequency
t-SNE	t-Distributed Stochastic Neighbor Embedding
USPS	US Postal Service Dataset
VAT	Virtual Adversarial Training
WebKB	Web Knowledge Base Dataset

SUMMARY

In recent years, deep learning and neural networks have shown very promising results and successes in many applications. Generally, deep neural networks need a lot of data to show their full potential in modeling and solving problems. But there are a handful of real-world applications where labeling data is expensive or not feasible while abundant unlabeled data is available. Semi-supervised learning has shown to be a successful solution for such scenarios. Recently semi-supervised and unsupervised learning for neural networks have attracted a lot of attention [13,81,106]. Most of these works are focused on traditional classification problems. In this thesis, we would explore and study semi-supervised learning for neural networks to tackle other problems and applications. We propose semi-supervised algorithms for three categories of machine learning problems: (i) verification problem, (ii) multi-view learning, (iii) fairness.

First, we propose two semi-supervised algorithms for verification problem. One of the proposed models benefits from auto-encoders and the other one benefits from adversarial training to exploit the unlabeled data to improve the accuracy in verification tasks. Then, we present a multi-view learning algorithm which is capable of benefiting from cross-view correlation between views to exploit the structural information exists in unlabelled data. In the last work, we study the effectiveness of semi-supervised learning in exploiting unlabeled data to improve the fairness of neural network classifiers.

CHAPTER 1

INTRODUCTION

1.1 Thesis Outline

Over the last recent years, deep learning has shown impressive results in various research areas including computer vision, speech recognition, natural language processing, and healthcare [5, 25, 59]. However, deep neural networks require a large amount of labeled training data to work effectively, and such data are not easily acquired for many problems due to its high costs and unavailability. On the contrary, semi-supervised learning resolves this problem by exploiting unlabeled data when a small subset of data has labels. In many applications, there are plenty of unlabeled data available which are usually inexpensive to obtain and can improve the secrecy of labeled data [19]. Exploiting unlabeled data can be helpful in increasing the generalization of supervised learning algorithms [81].

Motivated by the impressive performance brought by deep networks to many machine learning tasks along with their need for large amount of data have lead to the recent increase of the attention toward semi-supervised learning approaches based on deep neural networks [7, 13, 23, 46, 72, 81, 99, 100, 106]. However, most of these works are focused on the traditional single view classification tasks. In this thesis, we attempt to explore other aspects and applications of semi-supervised learning. We worked on three applications of semi-supervised

learning: (i) Verification Problem; (ii) Multi-view Problems; (iii): Fairness. Four different research directions are presented in this dissertation to tackle these three group of problems.

1.2 Semi-supervised Learning for Verification Problems

(Part of the section was previously published in [78, 79].)

The goal in verification problems is to determine the similarity of two samples or verifies if they belong to the same category or not. It has important applications in the face and fingerprint verification, where thousands or millions of categories are present, but each category has scarce labeled examples, presenting two major challenges for existing deep learning models. In such applications, it is also necessary to handle new classes without the need to train the model from scratch. Most of the traditional classification techniques have difficulties to address these challenges. Motivated by the impressive performance brought by deep networks to many machine learning tasks, we present two deep learning models to improve the existing verification models. However, deep networks require a large amount of labeled data for each class, which are not readily available in verification. Therefore, we propose two semi-supervised algorithms based on embedding learning for verification problems in Chapters 2 and 3.

In Chapter 2, we introduce a deep semi-supervised model, named SEmi-supervised VErification Network (`SEVEN`), to exploit the unlabeled data for verification tasks [79]. The proposed model uses autoencoders to benefits from unlabeled data. `SEVEN` consists of two complementary components. The autoencoder component addresses the lack of supervision within each category by learning general salient structures from a large amount of data across categories. The discriminative component exploits the learned general features to mitigate the lack of supervi-

sion within categories, and also directs the autoencoder component to find more informative structures of the whole data manifold. The two components are tied together in `SEVEN` to allow an end-to-end training of the two components.

In Chapter 3, we propose another semi-supervised embedding technique for verification tasks using deep neural networks [78]. The proposed model, named `VerVAT`, exploits the unlabeled data by making the model robust to the perturbation of the input with Virtual Adversarial Training (`VAT`) [72]. `VAT` is inspired by the adversarial training [36] technique originally proposed for increasing the robustness of neural networks toward adversarial examples. `VAT` has shown promising performance for semi-supervised classification tasks [81] where the distributions of the train and test data are similar. But to the best of our knowledge, it has never been applied to embedding learning problems where classes of the training and test data can be different. We are the first to adopt this idea and propose a semi-supervised learning model for verification tasks through the introduction of an objective function based on adversarial training. The proposed objective function is a combination of a discriminative part which imposes separation between various classes and an adversarial part which exploits the underlying structure of the unlabeled data. Adversarial training part of the model increases the generalization of the embedding function and prevents overfitting which is crucial for verification tasks.

1.3 Semi-supervised Learning for Multi-view Problems

(Part of the section was previously published in [77].)

In many real-world problems, more than one set of features, referred to as views or modalities of the data, are available. For example, a web page can be represented by text data, images,

and meta-data. Multiple views can help to improve the performance of many learning tasks because each view can provide information complementary to others, and learning using all views can maximally exploit the information available.

While neural networks for learning representation of multi-view data have been previously proposed as one of the state-of-the-art multi-view dimension reduction techniques, how to make the representation discriminative with only a small amount of labeled data is not well-studied. Most of the representation learning algorithms for multi-view problems ignore the labels of the data and thus not learn a sufficiently discriminative space for end tasks such as classification. Discriminative multi-view dimension reductions can help learn representations that can not only unify different views by dimensionality reduction but also discriminate different classes. However, as the neural networks become deeper, more parameters need to be learned and a larger amount of labeled data are required which are not readily available in many applications. The high cost of obtaining labeled data along with the growing size of unlabeled data has driven the development of semi-supervised learning that combines labeled and unlabeled data to mitigate the issue. However, there is still a lack of a semi-supervised deep discriminative method for multi-view dimension reduction.

In Chapter 4, we introduce a semi-supervised neural network model, named Multi-view Discriminative Neural Network (`MDNN`) [77], for multi-view problems. `MDNN` finds nonlinear view-specific mappings by projecting samples to a common feature space using multiple coupled deep networks. It is capable of leveraging both labeled and unlabeled data to project multi-view data so that samples from different classes are separated and those from the same class are

clustered together. It also uses the inter-view correlation between views to exploit the available information in both the labeled and unlabeled data.

To the best of our knowledge, `MDNN` is the first deep semi-supervised representation learning method in multi-view problems, which has all of the following properties in a single unified model: (i) yielding a discriminative feature representation, (ii) using the complementary information of other views to exploit the information in unlabeled data, and (iii) achieving the above properties using a large amount of unlabeled data to help learning with only a small amount labeled data.

1.4 Semi-supervised Learning for Fairness

In recent years, many decision-makings are being made automated by machine learning approaches. It has been shown that these models which are designed to help the process of decision-making are not immune to social biases [10, 22]. Machine learning algorithms are used currently or going to be used in the future for many sensitive applications like credit approvals, loan applications, criminal risk assessment, university admissions, or online advertisement. Therefore, it is important to consider other metrics and aspects of a machine learning algorithm other than just accuracy. Recently fairness in machine learning has become an important concern to build a socially responsible and inclusive system.

The naive approach of just removing or ignoring protected attributes like sex, gender, age in building machine learning systems does not work in many real applications [65] because 1) there exists already some degree of bias in the training data, and 2) there can exist some proxy features or correlation between features and protected attributes which may reveal those

protected attributes. Many learning algorithms have been proposed to address this problem and make the predictions of the learning algorithms fairer [2, 16, 40, 63, 111].

Unlabeled data do not contain label information which can be a significant source of bias in training machine learning systems. Additionally, in some real-world problems, not enough labeled data is available or labeling is expensive and time-consuming. In such scenarios, semi-supervised learning has shown to be an effective way of exploiting unlabeled data to increase accuracy. It has been shown that semi-supervised learning techniques can benefit from unlabeled data to improve the performance of a classifier in terms of accuracy, but to the best of our knowledge, there is no study on the effect of unlabeled data on the process of learning a fair classifier with neural networks. In Chapter 5, we propose a semi-supervised algorithm using neural networks, called `SSFair`, which benefits unlabeled data to not just improve the accuracy but also improving the fairness of the decision-making process. The proposed model exploits the information in the unlabeled data by using Pseudo-labeling [60] to mitigate the bias in the training data.

There exist different criteria to measure fairness in machine learning. Our proposed model is built with neural networks and can support any fairness measurement which can be defined or approximated as a differentiable function.

CHAPTER 2

AUTOENCODERS FOR SEMI-SUPERVISED VERIFICATION PROBLEM

(This chapter was previously published as "**SEVEN: Deep Semi-supervised Verification Networks**", in the Proceedings of 2017 International Joint Conference on Artificial Intelligence (IJCAI), 2017 [79].)

2.1 Introduction

Different from traditional classification tasks, the goal of verification tasks is to determine whether two samples belong to the same class or not, without predicting the class directly [21]. Verification tasks arise from applications where thousands or millions of classes are present with very few samples within each category (in some cases just one). For example, in face and signature verification, faces and signatures of a person are considered to belong to a class. While there can be millions of persons in the database, very few examples for each person are available. In such applications, it is also necessary to handle new classes without the need to train the model from the scratch. It is not trivial to address such challenges with traditional classification techniques.

Motivated by the impressive performance brought by deep networks to many machine learning tasks [8, 59, 116], we pursue a deep learning model to improve existing verification models. However, deep networks require a large amount of labeled data for each class, which are not

readily available in verification. There are semi-supervised training methods for deep network to tap on the large amount of unlabeled data. These semi-supervised methods usually have separate learning stages [74, 96]. They first pre-train a model using unlabeled data and then fine-tune the model with labeled data to fit the target tasks. Such two-phase methods are not suitable for verification. First, the large number of classes and the lack of data (be it labeled or unlabeled) within each category prohibit us from any form of within class pre-training and fine-tuning. Second, if we pool data from all categories for pre-training, the learned features are general but not specific towards each category, and the later fine-tuning within each category may not be able to correct such bias due to the lack of labeled data.

To address such challenges, we propose Deep SEmi-supervised VErification Networks (`SEVEN`) that consists of a generative and a discriminative component to learn general and category specific representations from both unlabeled and labeled data simultaneously. We cross the category barrier and pool unlabeled data from all categories to learn salient structures of the data manifold. The hope is that by tapping on the large amount of unlabeled data, the structures that are shared by all categories can be learned for verification.

Additionally, the proposed model adapts the general structures to each category by attaching the generative component to the discriminative component that uses the labeled data to learn category-specific features. In this sense, the generative component works as a regularizer for the discriminative component, and aids in exploiting the information hidden in the unlabeled data. On the other hand, as the discriminative component depends on the structures learned by the generative component, it is desirable to inform the generative component about the

subspace that is beneficial to the final verification tasks. Towards this end, instead of training the two components separately or sequentially, SEVEN chooses to train the two components simultaneously and allow the generative component to learn more informative general features.

We evaluate SEVEN on four datasets and compare it to four state-of-the-art semi-supervised and supervised algorithms. Experimental results demonstrate that SEVEN outperforms all the baselines in terms of accuracy. Furthermore, it has shown that by using very small amount of labeled examples, SEVEN reaches competitive performance with the supervised baselines trained on a significantly larger set of labeled data.

The rest of this chapter is organized as follows. In Section 2.2 we give an overview of the related works. In Section 2.3 we present SEVEN in detail. Section 2.4 gives the experimental evaluation and analysis of the proposed model.

2.2 Related Work

SEVEN can serve as a metric learning algorithm that is commonly employed in verification. The goal of metric learning is to learn a distance metric such that samples in any negative pair are far away and those in any positive pair are close. Many of the existing approaches [42, 80, 96, 112] learn a linear or nonlinear transformation that maps the data to a new space where the distance metric satisfies the above requirements. However, these methods do not address the large number of categories with scarce supervision information.

One of the earliest works in neural network-based verification is proposed by Bromley et al. for signature verification [15]. The proposed architecture, named Siamese networks, uses a contrastive objective function to learn a distance metric with ConvNets. Similar approaches

are employed for many other tasks such as face verification or re-identification [56, 97, 104]. It is worthy to mention that all these works are supervised and do not exploit unlabeled data.

Great interest in deep semi-supervised learning has emerged in applications where unlabeled data are abundant but obtaining labeled data is expensive or not feasible [41, 54, 60, 61, 87]. However, most of such approaches are designed for classification. To the best of our knowledge, there exists no deep semi-supervised learning to address the above two challenges in verification.

A key difference between `SEVEN` and most of the previous semi-supervised deep networks lies in the way that unlabeled and labeled data are exploited. Lee [60] has presented a semi-supervised approach for classification tasks called Pseudo-Label based on self-training scheme. It predicts the labels of unlabeled samples by training the model with the available labeled samples. Then they bootstrap the model with the highly confident labeled samples. This approach is prone to error because it may reinforce wrong predictions especially in problems with low confident estimation.

A more common semi-supervised approach is to pre-train a model with unlabeled samples and then the learned model is fine-tuned using the labeled samples. For example, [74] have pre-trained a Restricted Boltzmann Machine (`RBM`) with Noisy Rectified Linear Units (`NReLU`) in the hidden layers, then they used the learned weights to initialize and train a `Siamese` network [21] in a supervised way. The problem with pre-training based approaches is that the supervised part of the algorithm can ignore or lose what the model has learned in the unsupervised step. Another problem with pre-training based approaches is that they still need enough labeled examples for the fine-tunning step.

Recently, some works have tried to alleviate such problems by performing the learning process from all the labeled and unlabeled data in a joint manner for *classification* tasks [41, 61, 66, 87]. They make the unsupervised model involved in the learning as a regularizer for the supervised model. It should be considered that all such techniques are designed for classification tasks and can not handle the cases mentioned in the introduction such as the few samples per each class and the high number of classes.

Another line of work that handles a large number of categories is extreme multi-label learning [107]. The most popular assumption is that all classes have sufficient amount of labeled data, and this is clearly different from our problem setting. Recently, there are methods focusing on predicting the tail labels [48], but they are proposed for traditional classification task and can not handle new classes in the test data.

2.3 Proposed Model

2.3.1 Problem Formulation

The training set is represented as $\mathcal{X} = \left\{(x_1^i, x_2^i)\right\}_{i=1}^{N}$, where (x_1^i, x_2^i) is a pair of training samples $x_j^i \in \mathbb{R}^m$, $N = L + U$ is the total number of training pairs consisting of L labeled and U unlabeled pairs. The label set denoted by $\mathcal{Y} = \{y^i | y^i \in \{pos, neg\}\}_{i=1}^{L}$ specifies the relation between the samples of each pair. A positive relation indicates that two samples of the pair belong to the same class and a negative relation indicates the opposite. The relations for the unlabeled pairs are unknown.

Our goal is to learn a nonlinear function $r_{\theta_e}(x_1, x_2)\colon \mathbb{R}^m \times \mathbb{R}^m \rightarrow \{pos, neg\}$ parameterized by θ_e that predicts the relation between the two data samples x_1 and x_2. In other words, function $r_{\theta_e}(x_1, x_2)$ verifies if two samples are similar or not.

We define $r_{\theta_e}(.,.)$ based on the distance of x_1 and x_2 estimated by a metric distance function as:

$$r_{\theta_e}(x_1, x_2) = \begin{cases} neg & \text{if} \quad d_{\theta_e}(x_1, x_2) > \tau \\ pos & \text{if} \quad d_{\theta_e}(x_1, x_2) \leq \tau \end{cases} \tag{2.1}$$

where $d_{\theta_e}(.,.)$ is the metric distance function and threshold τ specifies the maximum distance that samples of a class are allowed to have. We define a nonlinear embedding function $f_{\theta_e}(.)$ that projects data to a new feature space and $d_{\theta_e}(x_1, x_2) = \|f_{\theta_e}(x_1) - f_{\theta_e}(x_2)\|_2$ is the Euclidean distance between x_1 and x_2 in the new space. An arbitrary distance function can be also used instead of the Euclidean distance.

2.3.2 Model Description

Our proposed model consists of discriminative and generative components. The model learns a non-linear function for each component. For the discriminative component, the nonlinear embedding function $f_{\theta_e}()$ is learned to yield "discriminative" and "informative" representation. In a discriminative feature space, similar samples are mapped close to each other while dissimilar pairs are far from each other. Such property is crucial for a good metric function. The generative component of the model is designed to exploit the information hidden in the unlabeled data. The desired representation should keep the salient structures shared by all categories as much

as possible. We define a probabilistic framework of the problem along with the discriminative and generative modelings of our algorithm.

The conditional probability distribution of the relation variable y given the i^{th} pair can be estimated as:

$$p(y^i|x_1^i, x_2^i) = 1 - tanh(d_{\theta_e}(x_1, x_2)) \tag{2.2}$$

which can be written as the following.

$$p(y^i|x_1^i, x_2^i) = \frac{2}{1 + \exp(2d_{\theta_e}(x_1, x_2))} \tag{2.3}$$

Here we use a *tanh* function to map the distance between samples to $[0, 1]$. However, any monotonic increasing function $u(.)$ which gives $u(0) = 1$ and $u(\text{inf}) = 0$ can be also used for this purpose.

We define $\tilde{p}$ as the ground truth distribution to be approximated by p (in Equation 2.3) as $\tilde{p}(y^i|x_1^i, x_2^i) = 1$ if $y_i = pos$, and $\tilde{p}(y^i|x_1^i, x_2^i) = 0$ otherwise. For the rest of the chapter, the conditional distributions $p(y^i|x_1^i, x_2^i)$ and $\tilde{p}(y^i|x_i^1, x_i^2)$ are denoted by p_i and $\tilde{p}_i$, respectively. Due to the probabilistic nature of such distributions, we approximate $\tilde{p}$ with p by minimizing the Kullback-Leibler divergence between them and introduce the following discriminative loss function $\mathcal{L}_{\mathcal{D}}(\mathcal{X}, \mathcal{Y}; \theta_e)$ defined over all the labeled pairs as:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{X}, \mathcal{Y}; \theta_e) = \sum_{i=1}^{L} l_d(x_1^i, x_2^i; \theta_e) = \sum_{i=1}^{L} KL(\tilde{p}_i \| p_i) \tag{2.4}$$

where $l_d(x_1^i, x_2^i; \theta_e)$ denotes the discriminative loss for the i^{th} pair, and $KL(\tilde{p_i} \| p_i)$ denotes the `KL-divergence` between $\tilde{p_i}$ and p_i. $KL(\tilde{p_i} \| p_i)$ can be substituted by $H(p_i, \tilde{p_i}) - H(\tilde{p_i})$ where $H(p_i)$ specifies the entropy of p_i, and $H(p_i, \tilde{p_i})$ defines the cross entropy between p_i and $\tilde{p_i}$. Considering that the loss function is optimized with a gradient based optimization approach and $H(\tilde{p_i})$ is a constant with respect to the the parameters, we simplify the discriminative loss function as:

$$l_d(x_1^i, x_2^i; \theta_e) = -I\{y_i = pos\} log(p_i) - I\{y_i = neg\} log(1 - p_i) \tag{2.5}$$

where $I\{.\}$ is the identity function. The loss function becomes equivalent to the the cross entropy over p_i and $\tilde{p_i}$. It penalizes large distance (similarity) between samples from the same (different) class to make the new space discriminative. $\mathcal{L}_{\mathcal{D}}$ attains its minimum when $p_i = \tilde{p_i}$ over all the labeled pairs.

To alleviate the insufficiency of the unlabeled data for verification task, through generative modeling, we encourage the embedding function $f_{\theta_e}(.)$ to learn the salient structures shared by all categories. We define a nonlinear function $g_{\theta_d}(.)$ parametrized by θ_d to project back the samples from new representation obtained from $f_{\theta_e}(.)$ to the original feature space.

The generative loss for the i^{th} pair (x_1^i, x_2^i) is defined as the reconstruction error between the original input and the corresponding reconstructed output as:

$$l_g(x_1^i, x_2^i; \theta_e, \theta_d) = \| g_{\theta_d}(f_{\theta_e}(x_1^i)) - x_1^i \|_2 + \| g_{\theta_d}(f_{\theta_e}(x_2^i)) - x_2^i \|_2 \tag{2.6}$$

where $g_{\theta_d}(f_{\theta_e}(x_j^i))$ indicates the reconstruction of the input of the x_j^i and is denoted by $\hat{x}_j^i$. The generative loss function $\mathcal{L}_\mathcal{G}$, over all pairs including labeled and unlabeled, is defined as:

$$\mathcal{L}_\mathcal{G}(\mathcal{X}; \theta_e, \theta_d) = \sum_{i=1}^{L+U} l_g(x_1^i, x_2^i; \theta_e, \theta_d) \tag{2.7}$$

We combine the generative and discriminative components into a unified objective function and write the optimization problem of `SEVEN` as:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}; \theta_e, \theta_d) = \sum_{i=1}^{L} l_d(x_1^i, x_2^i; \theta_e) + \alpha \sum_{i=1}^{L+U} l_g(x_1^i, x_2^i; \theta_e, \theta_d) + \beta(\|\theta_e\|_2 + \|\theta_d\|_2) \tag{2.8}$$

where $\|\theta_e\|$ and $\|\theta_d\|$ are the regularization terms on the parameters of the functions $f_{\theta_e}(.)$ and $g_{\theta_d}(.)$. The parameter β controls the effect of this regularization, parameter α controls the trade off between the discriminative and generative objectives.

2.3.3 Model Architecture and Optimization

We choose deep neural networks for modeling $f_{\theta_e}(.)$ and $g_{\theta_d}(.)$. The schematic representation of `SEVEN` is illustrated in Figure 1. The input pair is given to two neural networks denoted by F_1 and F_2 with shared parameters θ_e. They represent the discriminative component of the `SEVEN` (nonlinear embedding function $f_{\theta_e}(.)$). They project the input samples to the discriminative feature space. A layer, denoted by d, is added on top of the networks F_1 and F_2 that estimates the distance between the two samples of the input pair in the discriminative space.

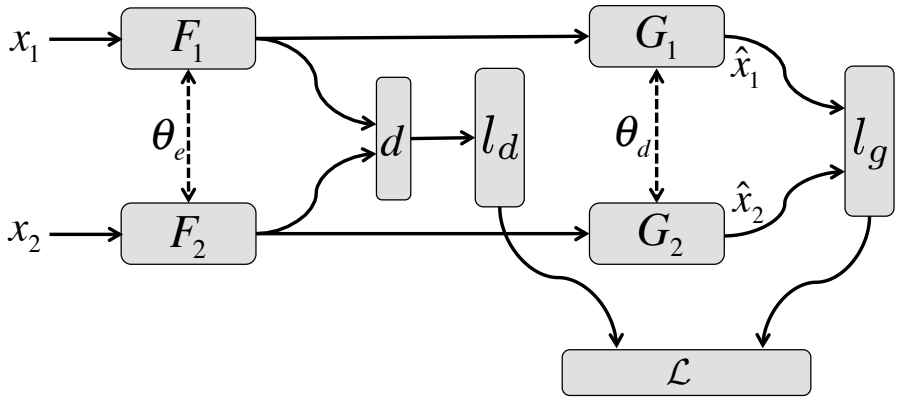


Figure 1: The overall architecture of `SEVEN`.

It can be considered as the metric distance function $d_{\theta_e}(.,.)$ which networks F_1 and F_2 are supposed to learn. The final layers of F_1 and F_2 are connected to two other subnetworks denoted by G_1 and G_2 in Figure 3 with shared parameters θ_d. They model the generative component of `SEVEN` ($g_{\theta_d}(.)$). They project back the samples to the original space. In other words, they can be considered as decoders for the encoders F_1 and F_2. The outputs of G_1 and G_2 shown as $\hat{x}_1$ and $\hat{x}_2$ are the reconstructions of the corresponding inputs x_1 and x_2.

Subnetworks F_1 and F_2 are `ConvNets` built with convolutional and max-pooling layers. G_1 and G_2 are made with transposed convolutional and upsampling layers which perform the reverse operations of convolutional and max-pooling layers, respectively. More detail of the transposed convolutional layer can be found in [29]. The complete specifications of the models for all the datasets are presented in Table I and Table II.

The whole model is trained using backpropagation with respect to the objective function in Equation 2.8. Given a set of N pairs, we optimize the model through an adaptive version of gradient descent called `RMSProp` [24] over shuffled mini-batches.

We employ $l2$-regularization and dropout [92] strategy to the convolutional and fully connected layers of the subnetworks to prevent overfitting. Batch normalization [45] technique is also applied after each convolutional layer to normalize the output of each layer. It can improve the performance in some cases. The training procedure of `SEVEN` is illustrated in Algorithm 1.

2.4 Experiments

2.4.1 Datasets

We evaluate the proposed algorithm on the following four datasets.

MNIST [26]: It is a dataset of 70000 grayscale images of handwritten digits from 0 to 9. We use the original split of 60000/10000 for the training and test sets. A uniform random noise of $[0, 1]$ is added to each pixel to make it noisy and more challenging.

US Postal Service (`USPS`) [44]: It is a dataset of 9298 handwritten digits automatically scanned from envelopes by the US Postal Service. All images are normalized to 16×16 grayscale images. We selected randomly 85% of the images for the training set.

Labeled Faces in the Wild (`LFW`) [43]: It is a database of face images that contains 1100 positive and 1100 negative pairs in the training set, and 500 positive and 500 negative pairs in the test set. All images are resized to 64×48.

BiosecurID-SONOF (`SONOF`) [34]: We use a subset of this dataset comprising signatures collected from 132 users, each user has 16 signatures. Signature images are normalized and

Procedure 1: Training procedure of `SEVEN`

Input: Training set: $\mathcal{X} = \left\{(x_1^i, x_2^i)\right\}_{i=1}^{N}$, label set $\mathcal{Y} = \{y^i\}_{i=1}^{L}$, number of iterations T, and batch size m.
Output: Model's parameters: Θ
$B = \frac{|\mathcal{X}|}{m}$;
// number of batches
Randomly split the training set $\mathcal{X}$ into B batches;
for $t = 1, 2, \cdots, T$ **do**
    for $b = 1, 2, \cdots, B$ **do**
        Feedforward propagation of the b^{th} batch;
        Calculate $\mathcal{L}^b$ according to Equation 2.8;
        Estimate gradients $\frac{\partial \mathcal{L}^b}{\partial \Theta_t}$ by backpropagation;
        Calculate Θ_{t+1} using `RMSProp`;
    end
end
return Θ_T;

converted to 80×80 grayscale images. We divided the users randomly into 100/32 for the training and test purposes.

In `SONOF` and `LFW` datasets, classes in the training and test samples are disjoint, while in `MNIST` and `USPS` classes are common between test and train sets. The samples of `LFW` are already in the form of pairs. For other datasets, we create the pairs by first splitting samples into two distinct sets for the training and test. We split the train set randomly into labeled and unlabeled samples. Then, each sample gets paired with two other samples randomly. One sample is selected from the same class to form a positive pair, and another one from a different class to form a negative pair.

2.4.2 Baselines

We compare the performance of SEVEN with the following baselines. It should be considered that we can not compare SEVEN with classification techniques because they are not usually designed to handle new classes in the test data which happens in verification applications. Since there are no other deep semi-supervised works for verification tasks, we adopt the common deep semi-supervised techniques to verification networks as our baselines.

Discriminative Deep Metric Learning (DDML) [42]: They developed a deep neural network that learns a set of hierarchical transformations to project pairs into a common space by using a contrastive loss function. It is a supervised approach and can not use unlabeled data.

Pseudo-Label [60]: It is a semi-supervised approach for training deep neural networks. It initially trains a supervised model with the labeled samples. Then it labels the unlabeled samples with the current trained model before each iteration, and use the high confidence ones along with the labeled samples for training in the next iteration. We followed the same approach for training a Siamese network [11] to extend their approach to the verification tasks.

Convolutional Autoencoder + Siamese Network (PreConvSia): We pre-train a Siamese network [11, 21] with an convolutional autoencoder model [70]. Then we fine-tune the network with labeled pairs. The network uses ConvNets as the underlying network for the modeling.

Autoencoder + Siamese Network (`PreAutoSia`): It is similar to `PreConvSia`, but uses MLP as the underlying network for the modeling. It is significantly faster in training compared to `PreConvSia`.

Principle Component Analysis (`PCA`): We use `PCA` as an unsupervised feature learning technique. The distance between samples in the new space learned by `PCA` indicates their relations. The threshold on the distance is selected for each dataset separately based on the performance on the training data.

2.4.3 Experimental Settings

The architectures of `SEVEN` for all datasets are presented in Table I and Table II. All the parameters of `SEVEN` and also other baselines are selected based on a validation on a randomly selected 20% subset of the training data. The $l2$-regularization parameter β is selected from $\{0.0001, 0.001, 0.01, 0.1\}$ for each dataset separately. The parameter α that controls the trade-off between generative and discriminative objectives is selected from $\{0.01, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$. It is set to 0.05, 0.1, 0.05, and 0.2 for `MNIST`, `LFW`, `USPS` and `SONOF`, respectively. Parameter τ is set to 0.5 for all the four datasets.

All the neural network models are trained for 150 epochs. The pre-training is also performed for 150 epochs for the baselines which require pre-training. `RMSProp` optimizer is used for the training of all the neural networks with the default value $\lambda = 0.001$ recommended in the original paper.

TABLE I: Specifications of the Neural Networks for `MNIST` and `USPS`. BN: batch normalization, ReLU: Rectified Linear Unit, Conv: convolutional layer, TransConv: transposed convolutional layer, Upsampling: Upsampling layer, Dense Layer: fully connected layer, and Max-pooling: max-pooling layer.

`MNIST` and `USPS`

Network F_i	**Network G_i**
`MNIST`: 28×28 `USPS`: 16×16	Input 128×1
3×3 Conv (8)	Dense Layer
ReLU	ReLU
2×2 Max-pooling	Reshape layer
Dropout (0.5)	2×2 Upsampling
5×5 Conv (8)	5×5 TransConv (8)
ReLU	ReLU
2×2 Max-pooling	Dropout (0.5)
Dropout (0.5)	2×2 Upsampling
Dense Layer (128×1)	3×3 TransConv (1)
ReLU	Sigmoid
	Dropout (0.5)

TABLE II: Specifications of the Neural Networks for `LFW` and `SONOF`. BN: batch normalization, ReLU: Rectified Linear Unit, Conv: convolutional layer, TransConv: transposed convolutional layer, Upsampling: Upsampling layer, Dense Layer: fully connected layer, and Max-pooling: max-pooling layer.

`LFW` and `SONOF`

Network F_i	**Network G_i**
`LFW`: 64×48 `SONOF`: 100×100	Input 128×1
4×4 Conv (32)	Dense Layer
BN-ReLU	ReLU
Dropout (0.5)	Reshape layer
2×2 Max-pooling	3×3 TransConv (64)
Dropout (0.5)	BN-ReLU
3×3 Conv (64)	Dropout (0.5)
BN-ReLU	2×2 Upsampling
2×2 Max-pooling	3×3 TransConv (32)
Dropout (0.5)	BN-ReLU
3×3 Conv (128)	Dropout (0.5)
BN-ReLU	2×2 Upsampling
Dropout (0.5)	3×3 TransConv (1)
Dense Layer (128×1)	BN-Sigmoid
ReLU	Dropout (0.5)

TABLE III: Performance of different methods on LFW and SONOF in terms of accuracy.

Dataset	LFW						SONOF					
# of labeled pairs	110	440	880	1320	1760	*All*	160	320	640	960	1280	*All*
SEVEN	**61.2**	**64.1**	**65.7**	**66.3**	**67.0**	68.7	**72.7**	**74.6**	**79.3**	**83.1**	**84.1**	85.3
PCA	-	-	-	-	-	64.5	-	-	-	-	-	67.61
DDML	51.5	54.2	61.9	63.8	64.8	**71.1**	58.5	67.7	72.5	78.4	82.9	**86.1**
Pseudo-Label	52.0	52.2	53.9	57.4	57.9	70.1	53.8	59.9	63.2	71.0	80.5	84.5
PreConvSia	55.1	62.3	63.5	63.2	64.2	66.0	61.9	67.1	70.4	71.5	78.8	82.1
PreAutoSia	51.1	62.9	63.0	63.5	64.2	66.1	57.2	62.7	66.4	70.1	73.1	79.0

TABLE IV: Performance of different methods on MNIST and USPS in terms of accuracy.

Dataset	MNIST						USPS					
# of labeled pairs	30	60	120	600	2400	*All*	40	80	160	300	800	*All*
SEVEN	**75.5**	**76.9**	**79.8**	**84.8**	90.7	96.8	**76.2**	**77.3**	**80.2**	**80.7**	82.8	93.1
PCA	-	-	-	-	-	65.84	-	-	-	-	-	70.96
DDML	61.1	65.9	75.7	84.0	90.4	96.8	69.0	71.8	75.7	75.9	80.8	92.7
Pseudo-Label	59.8	67.9	76.8	83.2	89.3	95.2	70.1	57.4	57.9	77.2	78.3	**93.3**
PreConvSia	64.4	73.0	77.2	82.7	**90.8**	**97.2**	72.2	78.2	77.6	78.1	**82.9**	93.0
PreAutoSia	61.5	68.0	71.9	78.9	84.7	93.1	70.6	73.9	69.0	75.0	82.0	90.2

2.4.4 Performance Evaluation

We report the performance in terms of accuracy which is the number of pairs in the test set verified correctly divided by the total number of pairs in the test set. The performance of SEVEN and all baselines are presented in Table III and Table IV. The results are reported for different number of labeled pairs and the best accuracy for each case is depicted in bold. The last column indicates the case where all labeled training pairs are used. PCA is a fully unsupervised method, thus one performance is reported for each dataset.

As can be seen from the tables, SEVEN outperforms other baselines in cases where a limited number of labeled pairs are used and the differences in performance are more significant where the number of labeled pairs is lower, and thus SEVEN can address the scarcity of labeled data better.

Algorithm DDML can give good performance when we have enough labeled data but its performance is significantly lower compared to SEVEN in cases with few labeled samples. DDML does not use the unlabeled data while other baselines benefit from the information hidden in the unlabeled data. By increasing the number of labeled pairs, the difference in accuracy decreases.

SEVEN outperforms all the semi-supervised baselines. One of the main advantages of SEVEN over other semi-supervised methods is that they perform supervised step after pre-training with unlabeled data is finished. This may cancel out some of the learned information from unlabeled data through a supervised process. There is no guarantee that the supervised process can benefit from the unsupervised learning [87]. Among the semi-supervised baselines, Pseudo-Label not only gives worse results compared to SEVEN, but also it shows lower performance than PreConvSia and PreAutoSia in many cases. It can be related to the noise and error in estimating the labels for unlabeled pairs.

2.4.5 Model Analysis

We perform some experiments to analyze the effect of the different components of SEVEN. The performances of different variants of SEVEN are given in Table V. The number of labeled pairs for each dataset is indicated in front of the name of the dataset. DisSEVEN indicates SEVEN with $\alpha = 0$ in Equation 2.8 which disables the G_i networks and the generative aspect of the

TABLE V: Performance of the variants of SEVEN.

Method	MNIST (120)	LFW (440)	USPS (80)	SONOF (320)
DisSEVEN	75.7	54.2	73.9	70.3
GenSEVEN	73.0	58.2	60.0	62.5
MLPSEVEN	73.1	60.0	77.3	70.9
SEVEN	**79.8**	**64.1**	**77.3**	**74.6**

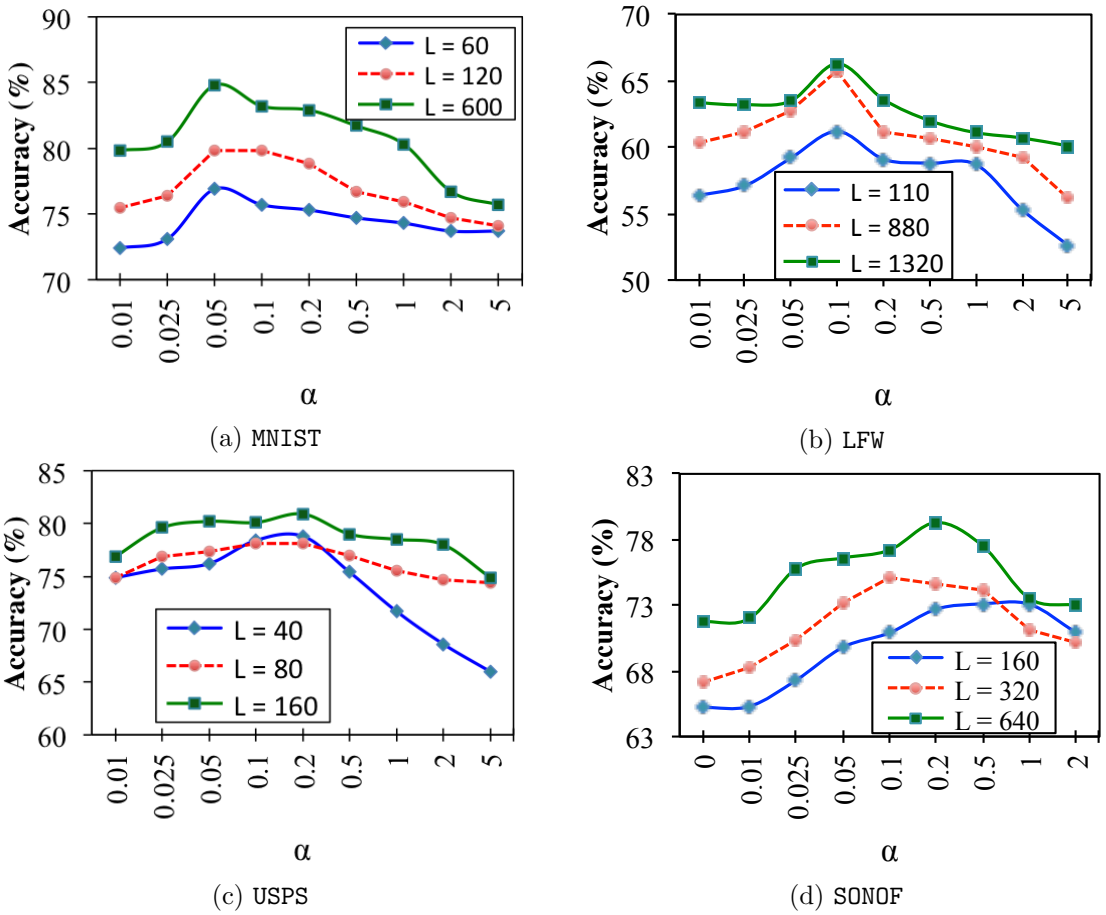
model. This variant does not consider the unlabeled data during the learning. GenGenSEVEN corresponds to a model that does not have the discriminate component. In other words, it does not have the contrastive layer and does not use the label information. SEVEN indicates the full variant of SEVEN with both generative and discriminative components. The variant MLPSEVEN is similar to the regular SEVEN, except that it uses fully connected layers instead of convolutional and transposed convolutional layers.

Among all the different variants, full SEVEN gives the best performance. It shows the effectiveness of both the generative and discriminative components. It also verifies the effectiveness of using the information hidden in the unlabeled data. The results show that the discriminative component has the broader impact compared to the generative component. MLPSEVEN gives weaker performance compared to SEVEN. It is mainly because of the capabilities of convolutional layers in modeling image data as it has also been shown by ConvNets in image processing applications.

2.4.6 Parameter Sensitivity

We analyze the effect of the parameter α in Equation 2.8 on the performance of SEVEN on all the four datasets. Parameter α of SEVEN controls the trade-off between the generative and discriminative aspects of the model. In Figure 2 the performance of SEVEN for different values of α is plotted. For each dataset, the performance is plotted for three different values of L (number of labeled pairs).

There exists a trade-off between the two generative and discriminative aspects of SEVEN on all of the four datasets. As it can be seen, the optimum value of this parameter is dependent to the dataset and also to the ratio of labeled data to some extent.



(a) MNIST (b) LFW

(c) USPS (d) SONOF

Figure 2: The accuracy of SEVEN for different values of parameter α for (a) MNIST, (b) LFW, (c) USPS and (d) SONOF.

CHAPTER 3

VIRTUAL ADVERSARIAL TRAINING FOR SEMI-SUPERVISED VERIFICATION PROBLEM

(This chapter was previously published as "**Virtual Adversarial Training for Semi-supervised Verification Tasks**", in the Proceedings of the 27th European Signal Processing Conference (EUSIPCO), 2019, IEEE [78].)

3.1 Introduction

The task of estimating the similarity of two objects is called verification task. It has important applications such as face verification [42], signature verification [15], and learning sentence similarity [73]. Most of the suggested models for verification tasks are based on embedding learning techniques. It makes them applicable for some other tasks such as classification problems in scenarios with a large number of classes and limited or skewed number of samples for each class [73, 75].

Deep learning models have shown great and promising performance in many applications recently [84] but most of the successes are in supervised tasks where a large amount of labeled data is available. Labeling data can be very expensive or not feasible in some cases while unlabeled data are abundant for many problems. In such applications, semi-supervised learning can be an effective solution. While some works have been done on training neural networks in semi-supervised setting for classification problems, to the best of our knowledge, limited

works have been done on deep semi-supervised verification tasks. In [79], a semi-supervised model, called SEVEN, is proposed which combines a supervised loss with an unsupervised one to handle unlabeled data. It showed promising results compared to the baselines. However the unsupervised part of SEVEN is based on auto-encoders. One of the drawbacks of auto-encoding approach is that the decoder part doubles the size of the network.

In this work, we propose a semi-supervised embedding model for verification tasks. The proposed algorithm, named VerVAT, benefits from Virtual Adversarial Training (VAT) [72] to exploit the unlabeled data. Adversarial training may refer to different categories of machine learning algorithms. These algorithms are used for a variety of problems such as Generative Adversarial Networks (GAN) [35], adversarial examples [36], and adversarial loss optimization [32]. VAT is adopted from the adversarial training [36] technique originally proposed for increasing the robustness of neural networks toward adversarial examples. VAT has shown promising performance for semi-supervised classification tasks [81] where the distributions of train and test data are similar. But to the best of our knowledge, it has never been applied to embedding learning problems where classes of the training and test data can be different. We are the first to adopt this idea and propose a semi-supervised learning model for verification tasks through the introduction of an objective function based on Virtual Adversarial Training. The proposed objective function is a combination of a discriminative part which imposes separation between various classes and a VAT based part which exploits the underlying structure of the unlabeled data. Virtual adversarial loss also helps the model to avoid overfitting and to have a smoother embedding function.

The proposed model can also be used in other tasks such as extreme classification where there exists a large number of classes in the order of thousands or millions. One common example of such tasks is face recognition where there may exist millions of classes with few samples for each class. In such settings, traditional neural networks for classification suffer from long tail problem and overfitting [89].

We have evaluated `VerVAT` on three different verification tasks. In two of them, the training and test samples are drawn from disjoint classes which can not be handled easily by most of the traditional classification techniques for neural networks. In all of the experiments, the proposed algorithm achieves better results in terms of accuracy compared to the baselines. It shows the effectiveness of Virtual Adversarial Training for semi-supervised embedding learning.

3.2 Problem Formulation

We define the training data as a set of pairs consisting of two samples. The items of a pair can belong to the same class to form a positive pair or belong to different classes to from a negative pair. If the class information for at least one of the items of a pair is missing or not available, the pair's label is considered as unknown. The training set is represented as $\mathcal{D} = \left\{(x_1^i, x_2^i)\right\}_{i=1}^{N}$, where (x_1^i, x_2^i) is a pair of training samples. $x_1^i \in \mathbb{R}^m$ is the first item of the i^{th} pair and $x_2^i \in \mathbb{R}^m$ is the second item. The total number of pairs is indicated by N. The label set is defined as $\mathcal{L} = \{y^i | y^i \in \{p, n, u\}\}_{i=1}^{L}$ where p, n, and u denote the positive, negative and unknown label, respectively.

We want to learn a parametric and highly nonlinear function that can verify whether two samples are similar or not. To be more specific, the goal of the model is to learn function

$v(x_1, x_2; \Theta)$ to predict the relation between x_1 and x_2. It is defined based on the distance of x_1 and x_2 as

$$v(x_1, x_2; \Theta) = \begin{cases} p & \text{if} \quad d(f(x_1; \Theta), f(x_2; \Theta)) \leq \tau \\ n & \text{if} \quad d(f(x_1; \Theta), f(x_2; \Theta)) > \tau \end{cases} \tag{3.1}$$

where $d(.,.)$ is an arbitrary distance function. Function $f(x; \Theta)$ is a highly nonlinear function parameterized by Θ which maps a sample to a new space where distances can get estimated by a simple distance function like Euclidean or cosine distance. The threshold τ specifies the maximum distance that samples of a positive pair are allowed to have from each other. Samples farther than this threshold are considered to be from different classes with negative relation.

3.3 Proposed Algorithm

3.3.1 Model Architecture

The overall architecture of the proposed model is illustrated in Figure 3. The input pair is given to two neural networks denoted as F_1 and F_2 with shared weights and parameters Θ like `Siamese` networks [15]. `Siamese` networks are widely used in similarity learning [11, 73, 75], embedding learning [3, 9], verification [15, 21, 56], and retrieval [85].

They should project the input samples to a new discriminative space where samples with positive relation are close to each other and samples with negative relation are far from each other. As the weights of F_1 and F_2 are shared, both subnetworks define the same nonlinear mapping function, denoted by $f(.; \Theta)$. To make the new representation to have such a discriminative property, a layer is added at the top of the networks F_1 and F_2 that calculates the distance between the two input samples in the new space denoted by $d(.,.)$. Function d can be

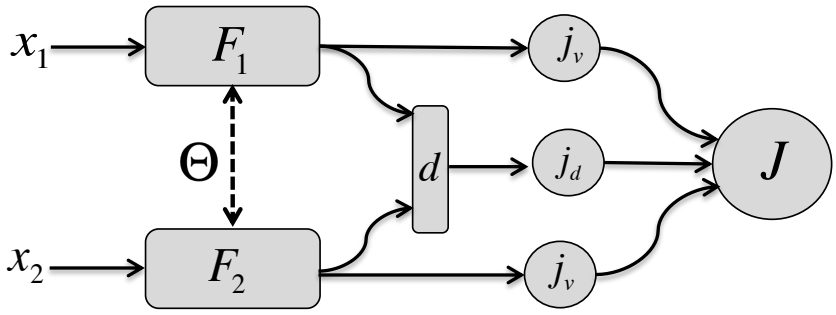


Figure 3: The schematic representation of `VerVAT`. F_1 and F_2 are the neural networks with shared weights, and the circle shapes denote the loss functions.

an arbitrary distance metric such as Euclidean or cosine distance in the new subspace. This function can be considered as a metric distance function which networks F_1 and F_2 are supposed to learn it. These networks are `ConvNets` built with convolutional layers, max-pooling, and a fully connected layer as the last layer.

3.3.2 Loss Function

We propose to impose two main characteristics on the new subspace to be learned by networks F_1 and F_2. First of all, the new subspace obtained from these two networks should be discriminative so that samples from different classes are separable easily. Samples from the same class should be close to each other, and samples from different classes should be far from each other. This property makes the similarity prediction performed by function $d(.,.)$ easier. However, the discriminative property is not enough for semi-supervised settings where the relation of some pairs are not available.

To fully exploit the information of all data, we impose the unsupervised constraint. Another challenge in training neural networks is overfitting especially when the distribution of the classes in the test and train are different [81]. To address this problem, we propose to adopt the idea of Virtual Adversarial Training (`VAT`) to regularize the training process (will be explained in detail in Section 3.3.2.2). Both properties are imposed by a unified loss function as

$$\mathcal{J}(X, Y; \Theta) = (1 - \alpha)\mathcal{J}_{\mathcal{D}}(X, Y; \Theta) + \alpha\mathcal{J}_{\mathcal{V}}(X; \Theta) + \beta \left\|\Theta\right\|_2 \tag{3.2}$$

where $\mathcal{J}_{\mathcal{D}}(X, Y; \Theta)$ indicates the supervised loss for labeled data, and $\mathcal{J}_{\mathcal{V}}(X; \Theta)$ is the unsupervised loss for all data which imposes the adversarial training loss on the learned function. Parameter β controls the regularization term $\|\Theta\|$ which is imposed on all the weight parameters of the network. Regularization to prevent overfitting is important especially in cases where the distribution of train and test data are not similar. Parameter α is the weighting parameter that controls the trade-off between the supervised and unsupervised part of the loss.

3.3.2.1 Discriminative Space

The discriminative part of the loss function, $\mathcal{J}_{\mathcal{D}}(X, Y; \Theta)$ is estimated for the L labeled pairs as:

$$\mathcal{J}_{\mathcal{D}}(X, Y; \Theta) = \sum_{1 \leq i \leq L} j_d(x_1^i, x_2^i; \Theta) \tag{3.3}$$

where $j_d(x_1, x_2)$ indicates the discriminative loss for the pair (x_1, x_2) in the new subspace. It can be defined with a contrastive loss function as:

$$j_d(x_1^i, x_2^i; \Theta) = I\{y^i = p\} d(f(x_1; \Theta), f(x_2; \Theta))^2 + I\{y^i = n\} max\{0, m - d(f(x_1; \Theta), f(x_2; \Theta))\}^2 \tag{3.4}$$

where $I\{.\}$ is the identity function. Function d measures the distance of two samples in the new space. We used Euclidean distance as the distance function. It penalizes the distance between positive samples and also the similarity between negative ones. This loss pushes the positive samples close to each other in the space while pushes negative samples far from each other. It makes the new representation space discriminative. Parameter m specifies a margin which prevents the loss function to push negative pairs further than m.

3.3.2.2 Virtual Adversarial Training

In order to exploit the information in the unlabeled data we adopt Virtual Adversarial Training (`VAT`) [72] to our embedding learning model. `VAT` is inspired by the defense techniques which are used to increase the robustness of neural networks toward adversarial attacks. It tries to minimize the change in the output of a neural network when its input is perturbed locally. It regularizes the embedding space and increases the generalization of the learned subspace while there exists limited labeled data. `VAT` has shown to be effective for semi-supervised learning [81].

The loss $\mathcal{J}(X, Y; \Theta)$ for all the pairs including unlabeled and labeled samples is defined as:

$$J_v(X; \theta) = \sum_{1 \leq i \leq N} \sum_{j=1}^{2} j_v(x_j^i; \theta) \tag{3.5}$$

where $j_v(x_j^i;\theta)$ estimates the `VAT` loss for sample x_j^i. It is defined as the following to minimize the greatest change in the embedding space for sample x_j^i.

$$j_v(x_j^i;\theta) = g(f(x_j^i;\theta), f(x_j^i + r_{adv};\theta))$$

$$r_{adv} = \underset{r;\|r\|_2<\varepsilon}{\arg\max}\, g(\, f(x_j^i;\theta), f(x_j^i + r;\theta)) \tag{3.6}$$

where g is a non-negative function which measures the distance between its two inputs, and ε is a small positive number. We selected Euclidean distance function as the distance function g. Vector r_{adv} is the adversarial perturbation which specifies the direction in the input space which produces the maximum difference in the embedding space. By minimizing this loss function, the sensitivity of the output embedding space to the input perturbation is minimized. There exists no closed form to calculate the vector r_{adv}, but it can get approximated by

$$r_{adv} = \varepsilon \frac{g}{\|g\|} \tag{3.7}$$

where

$$g = \nabla_r d(f(x_j^i;\theta), f(x_j^i + r;\theta))$$

$$r \sim N(0, \frac{\varepsilon}{\sqrt{D_x}} I) \tag{3.8}$$

Vector r is a random noise vector added to the input of the neural networks to create the perturbation. It is drawn from a normal distribution N. D_x is the dimension size of the inputs, and I is an identity matrix with the dimension of D_x. The gradient vector g can get computed by back-propagation on the network. More details on this approximation can be found in [72].

The whole model is trained using backpropagation with respect to the loss function in Equation 3.2. Given a set of N pairs, we optimize the model by `Adam` [53] optimization technique over shuffled mini-batches. Batch normalization [45] technique is also applied after each convolutional layer to normalize the output of each layer.

3.4 Experiments

3.4.1 Datasets

We evaluate the proposed algorithm on the following datasets:

Labeled Faces in the Wild (`LFW`) [43]: It is a database of face photographs designed for evaluating face verification or recognition tasks. It contains 2200 pairs of face images consisting of 1100 positive and 1100 negative pairs for verification tasks. Positive pairs are images from the same person, while negative pairs are from different persons. There are 500 positive and 500 negative pairs in the test set. Due to the small size of the training data, we use 5-fold validation in the validation process for estimating the best parameters.

BiosecurID-SONOF (`SONOF`) [34]: We use a subset of this dataset comprising signatures collected from 132 users. It contains 16 signatures for each user. All images are normalized and resized to 80×80. Users are randomly divided into two groups of 100 and 32 for the training and test purposes.

US Postal Service (USPS) [44]: This dataset contains 9298 handwritten digits automatically scanned from envelopes by the US Postal Service. It has 10 classes. All images are normalized 16 x 16 grayscale. We divided the samples randomly into 7900/1398 for training and test. After the pairing process, we will have 7900 and 1398 pairs for training, and test. We used 5-fold cross-validation for estimating the best values for the parameters. All images are resized to 64×48.

Dataset LFW is originally built for verification tasks, and its train and test samples are already in the form of positive and negative pairs, but the rest are mostly used for image classification tasks. We make these datasets in pairs so that they can be used for verification. The pairing process is as follows. First, we split the training data randomly into labeled and unlabeled sets with the specified ratio. Then, each sample gets paired with another sample randomly. The other sample is selected from the same class with the probability of 0.5, otherwise from a different class to have equal number of positive and negative pairs. The pairs are selected from their own corresponding set, labeled or unlabeled. Test or validation samples are not divided into labeled and unlabeled sets like training set, but they just get paired with a similar process. The classes in the training and test samples are disjoint in SONOF and LFW datasets, while in USPS dataset, classes are common between the test and train.

3.4.2 Baselines

Handling new classes in the test data is a common case in verification tasks, while it is a great challenge for most of the traditional classification techniques based on neural networks.

Therefore, we adopted some of the deep semi-supervised techniques to verification networks to be used as our baselines.

Principle Component Analysis (`PCA`): It as an unsupervised feature learning technique which does not need any label information. The distance between samples after applying the `PCA` transformation is considered as the similarity of two samples. A threshold is selected for each dataset based on the performance on the training data to find the relation between two samples.

Pseudo-Label [60]: It is a semi-supervised approach for training deep neural networks. It initially trains a supervised model with the labeled data. Then in each epoch, it predicts the labels of the unlabeled samples with the trained model, and then adds the ones with high confidence to the labeled samples to continue training. The model was proposed and evaluated for classification tasks. We followed the same approach to train a `Siamese` network [15].

Discriminative Deep Metric Learning (`DDML`) [42]: It uses the architecture of `Siamese` networks [15] with a modified version of the contrastive loss function. It is a supervised approach and does not use unlabeled pairs.

Autoencoder-Siamese: It pre-trains an autoencoder in an unsupervised manner. Then, its encoder part is fine-tuned with labeled pairs in a `Siamese` network [15] structure.

SEVEN [79]: It is a model based on neural networks specifically proposed for semi-supervised verification tasks. This model used auto-encoding and generative models to handle unlabeled data and prevent overfitting problem while our algorithm benefits Virtual Adversarial Training to exploit the information in the unlabeled data.

TABLE VI: Performance of different methods on LFW, SONOF, and USPS in terms of accuracy.

Dataset	LFW				SONOF				USPS			
# of labeled pairs	110	880	1760	All	160	640	1280	All	40	160	800	All
PCA	-	-	-	64.5	-	-	-	67.6	-	-	-	70.9
DDML [42]	51.5	61.9	64.8	71.1	58.5	72.5	82.9	86.1	69.0	75.7	80.8	92.7
Pseudo-Label [60]	52.0	53.9	57.9	70.1	53.8	63.2	80.5	84.5	70.1	57.9	78.3	**93.3**
Autoencoder-Siamese	55.1	63.5	64.2	66.0	61.9	70.4	78.8	82.1	72.2	77.6	82.9	93.0
SEVEN [79]	61.2	65.7	67.0	68.7	72.7	79.3	84.1	85.3	76.2	80.2	82.8	93.1
VerVAT	**61.6**	**68.6**	**72.6**	**73.5**	**82.9**	**83.45**	**85.6**	**87.7**	**78.2**	**84.5**	**84.9**	93.0

3.4.3 Performance Evaluation

The performance of VerVAT and all baselines are presented in Table VI. The results are reported for a different number of labeled pairs and the best accuracy for each case is depicted in bold. The performance is reported in terms of accuracy which is the number of pairs in the test set verified correctly divided by the total number of pairs in the test data. The last column of each section indicates the case where all the training pairs have label information. As PCA is a fully unsupervised method, no label information is used for this baseline, and just one performance is reported for each dataset.

Most of the parameters of baselines are selected based on the accuracy metric using cross-validation. USPS is divided into training and validation sets because is has enough samples, but LFW and SONOF are validated with 5-fold validation. After finding the best values for parameters with 5-fold validation, the whole training data is used for training.

All the neural networks are trained for 250 epochs with Adam [53] optimizer and the best model with the lowest loss is selected as the final model. The pre-training phase of training for

both `Pseudo-Label` and `Autoencoder-Siamese` is performed for 150 epochs. The batch size is set to 512 for all the experiments. Margin parameter m is set to 1.0.

As can be seen, `VerVAT` outperforms other baselines in terms of accuracy in cases with limited number of labeled pairs. The difference in performance compared to other baselines is more significant for the lower number of labeled pairs. It verifies empirically the effectiveness of the proposed approach of addressing the problem of limited labeled data.

One of the drawbacks of `SEVEN` and `Autoencoder-Siamese` is that they use autoencoders. Their encoders should incorporate most of the unnecessary detail of the image data into the hidden representations so that the decoder can reconstruct the original input. Such representations contain unnecessary information for the goal task and can affect the performance of the verification task while `VerVAT` benefits `VAT` to exploit the unlabeled data and does not have this limitation.

CHAPTER 4

SEMI-SUPERVISED DEEP REPRESENTATION LEARNING FOR MULTI-VIEW PROBLEMS

(This chapter was previously published as "**Semi-supervised Deep Representation Learning for Multi-View Problems**", in the Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), 2018, IEEE [78].)

4.1 Introduction

In many real-world problems, more than one set of features, referred to as views of the data, are available. For example, a web page can be represented by text data, images, and meta-data. Multiple views can help improve the performance of many learning tasks because each view can provide information complementary to others, and learning using all views can maximally exploit the information available. In particular, multi-view dimension reduction has been proven effective for learning from high dimensional multi-view data [109] such as image and text in image processing [20], speech and video in speech processing [6], and multilingual texts in language processing [31].

Compared to traditional multi-view dimension reduction, multi-view dimension reduction using deep networks has shown the state-of-the-art results. Projections are learned to map all views to a common feature space where view information is retained and fused. The new space can enable or improve learning algorithms that are not applicable or inferior in multiple high-

dimensional spaces. Low-dimensional representations learned in such a way are without labeled data and thus not sufficiently discriminative for end tasks such as classification. Discriminative multi-view dimension reductions based on CCA [102], topic models [117], and information bottleneck [108] can help learn representations that can not only unify different views by dimensionality reduction but also discriminate different classes. However, as the networks become deeper, more parameters need to be learned and a larger amount of labeled data are required which are not readily available in many applications. The high cost of obtaining labeled data along with the growing size of unlabeled data has driven the development of semi-supervised learning that combines labeled and unlabeled data to mitigate the issue. However, there is still a lack of a semi-supervised deep discriminative method for multi-view dimension reduction.

We propose MDNN (Multi-view Discriminative Neural Network) for the above purpose, using only a small amount of labeled data and a large amount of unlabeled data. MDNN maximizes between-class separations and minimizes within-class variations while leveraging label information for discriminativeness. MDNN consists of a pair of parallel neural networks coupled by a shared layer on the top of the last layers (see Figure 4). The model is trained in a joint manner to find view-specific nonlinear transformations. The learned transformations are further used to project samples to the common space. MDNN not only projects paired instances from different views to the same space (maximal correlation) but also projects instances from different classes far from each other (inter-class separation) while instances with the same class label are close to each other (intra-class variation).

To the best of our knowledge, MDNN is the first deep semi-supervised representation learning method in multi-view problems, which has all of the following properties in a single unified model: (i) yielding a discriminative feature representation, (ii) using the complementary information of other views to exploit the information in unlabeled data, and (iii) achieving the above properties using a large amount of unlabeled data to help learning with only a small amount labeled data. We evaluate MDNN on four multi-view datasets, namely Noisy MNIST, WebKB, FOX, and CNN, and compare it to the state-of-the-art baselines. Experimental results demonstrate that the proposed algorithm outperforms all the baselines in terms of accuracy especially when a limited number of labeled samples are available. The reminder of this chapter is organized as follows. In Section 4.2, an overview of the related previous works is given. The proposed algorithm is discussed in detail in Section 4.3 and experiments are presented in Section 4.4.

4.2 Previous Works

In Table VII, capabilities of different multi-view dimension reduction models are compared. The proposed algorithm (MDNN) is the only one that enjoys all the capabilities. CCA is a well-known dimension reduction technique for data with two views [6,20,27,33,39]. It finds two linear transformations to project the views to a common feature space, so that correlation between views is maximized. However, CCA suffers from the lack of nonlinearity in its transformations to model nonlinear data. Kernel CCA (KCCA) extends CCA to find nonlinear projections [71] for both views. KCCA requires training data during testing and does not easily scale to large datasets.

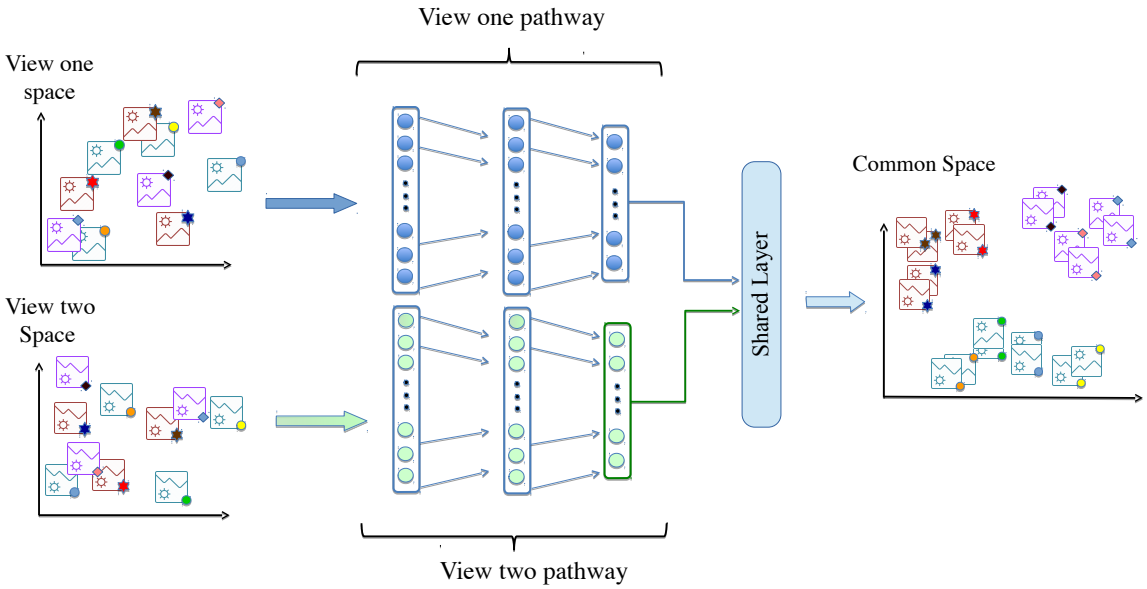


Figure 4: The schematic structure of MDNN for two views. In the left side of the figure, instances are shown in their original input space for both views. After passing them through MDNN, the common feature space is obtained. It is depicted on the right side of the figure. In this example, samples belong to three classes. The color indicates the class of an instance. Corresponding views are marked with small shapes at the corner of each instance.

More recently, several deep neural network (DNN)-based algorithms have been proposed for nonlinear feature representation learning on multi-view problems [76,88]. A deep model for CCA estimation, referred to as Deep CCA (DCCA), has also been proposed [4,12]. Like CCA, DCCA is a parametric approach and is scalable to large datasets, and like KCCA, it can model nonlinearity in the data.

Nonetheless, Linear CCA, KCCA and Deep CCA are unsupervised feature learning techniques. They cannot exploit labels available (if any) during feature representation learning. The learned low-dimensional representations thus lack class discriminativeness that is critical to the success

of end tasks such as classification and clustering. While discriminative representation learning from one-view using deep network or topic model has also been explored in [108,117]. Learning a discriminative representation from multi-view data can require more labeled data as the number of views and network layers increase. While semi-supervised techniques for deep learning has been explored in [82,115] to use a large amount of unlabeled data to mitigate the lack of labeled data, semi-supervised discriminative multi-view learning has not been studied and is the focus of this work.

Maximizing between-class separations while minimizing within-class variations have been widely used in many learning algorithms, such as Fisher's Linear Discriminant Analysis (`FLDA`) [95]. However, `FLDA` is a linear technique and although kernel-based versions of `LDA` have been proposed (`KLDA`) [71], they suffer from similar drawbacks of `CCA` and `KCCA`, such as scalability and fixed kernels. Recently, a deep version of `LDA` has been introduced [28]. However, all these studies work with only a single view and do not benefit from the noise-robustness of `CCA`-based techniques, which is the result of maximizing the correlation between views.

They are some scattered works in multi-view learning such as [17, 90] but they are clearly different from our problem setting.

4.3 The Proposed Algorithm

The schematic representation of the proposed model for two views is shown in Figure 4. The `MDNN` comprises of two deep neural networks (one for each view) coupled in a shared layer (interview-layer). More networks can get coupled to handle more views. Both networks are trained jointly to find view-specific nonlinear transformations to map the input views to a

TABLE VII: Comparison of different techniques with MDNN on various aspects.

Method	**Nonlinearity**	**Scalability**	**Discriminativity**	**Multi-view**
CCA				✓
KCCA	✓ (limited)			✓
DCCA	✓	✓		✓
LDA		✓	✓	
KLDA	✓ (limited)		✓	
Deep LDA	✓	✓	✓	
MDNN	✓	✓	✓	✓

common feature space. The inter-view layer encourages inter-view correlation between views and is responsible for exploiting the information in both views of both labeled and unlabeled. All views of a single instance are projected as near as possible to each other. Moreover, two objectives are imposed on the output layer of each view independently to make the new space discriminative. It is achieved by maximizing intra-view discrimination using the labeled data: instances of the same class in one view are mapped closed together, whereas instances of different classes are mapped distant apart. Such properties make all the views of each instance to be *highly correlated*, and instances of different classes are *easily separable*.

These two parts of the model work in a joint manner to learn the desired representation from all the labeled and unlabeled data, and each can be considered as a regularizer for the other during the subspace learning. We train our model with backpropagation to learn two nonlinear transformations through optimizing the introduced objective functions. After training, the network is employed to map multiple views of data to a common low-dimensional space, where classifiers can be trained.

The purpose of using an independent network for each view is to learn low-level view-specific representations according to the properties of each view. Thus, the architecture of each network, such as the type or number of layers, can get adjusted according to the view's properties. In addition, representations obtained from higher levels of the networks are more likely to reveal the views' statistical properties compared to the original inputs [93].

4.3.1 Deep Model Definition

For a two-view problem, the training set is represented as $\mathcal{X} = \big\{(x_1^i, x_2^i)|x_1^i \in X_1, x_2^i \in X_2, 1 < i < N\big\}$, where (x_1^i, x_2^i) is a training sample with views $x_1^i \in \mathbb{R}^p$ and $x_2^i \in \mathbb{R}^q$ with dimension p and q, respectively. $N = L + U$ is the total number of training pairs consisting of L labeled and U unlabeled pairs. The label set for labeled samples is denoted by $\mathcal{Y} = \{y^i|1 \leq i \leq L\}$.

We aim to learn two nonlinear view-specific functions $f_1(x; \Theta_1) : X_1 \rightarrow Z_1$ and $f_2(x; \Theta_2) : X_2 \rightarrow Z_2$ that map the given paired views to the embedding spaces Z_1 and Z_2. Slightly abusing the notation, inputs to the first layers of the networks for the two views are batches of samples denoted by X_1 and X_2, and the hidden representations output by the last layers right before the shared layer are denoted by Z_1 and Z_2. Parameters Θ_1 and Θ_2 are the parameters of the two networks, respectively (see Figure 5).

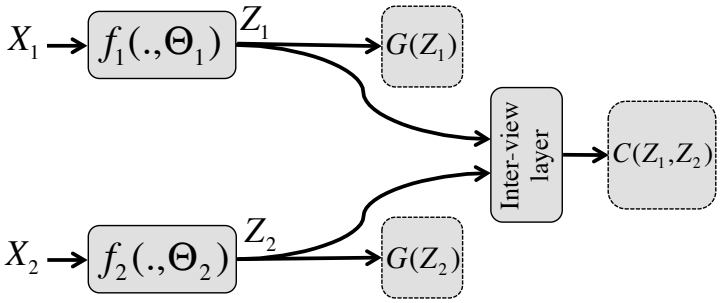


Figure 5: The overall model of `MDNN`. A batch of instances are denoted for view one and two with X_1 and X_2, respectively. They are passed through non-linear view-specific functions f_1 and f_2. Discriminativity is imposed through the objective function $G(.)$ over the outputs Z_1 and Z_2. The maximization of inter-view correlation is imposed through the objective $C(Z_1, Z_2)$.

4.3.2 Objective Function

To learn a discriminative representation for more effective classification, we define the objective function

$$L(Z_1, Z_2; \theta_1, \theta_2) = C(Z_1, Z_2) + \lambda(G(Z_1) + G(Z_2)) - \alpha(\|\theta_1\|^2 + \|\theta_2\|^2) \tag{4.1}$$

where the function $C\,(Z_1, Z_2)$ maximizes the inter-view correlation between the samples in the new space, and functions $G(Z_i)$ encourages discriminative subspaces. The term $\|\Theta_1\|^2 + \|\Theta_2\|^2$ with regularization parameter α is added to regularize the networks. Parameter λ specifies the trade-off between the importance of the inter-view correlation and intra-view discrimination properties in the new space.

We define the function C based on `CCA` by following the approach that is proposed in `Deep CCA` [4] for neural networks. `CCA` maps multiple views of samples into a new space where paired

views of each sample are highly correlated using a linear transformation matrix. It has been shown that the orthogonality of the learned dimensions is critical to effective representations of the multi-views [102].

Considering the outputs of the two branches of `MDNN` as two sets of variables, also denoted by Z_1 and Z_2, `CCA` maximizes their correlation

$$(v_1^*, v_2^*) = \underset{v_1, v_2}{\arg\max}\, \text{corr}(v_1^T Z_1, v_2^T Z_2) = \underset{v_1, v_2}{\arg\max} \frac{v_1^T{}_1 \Sigma_{12} v_2}{\sqrt{v_1^T \Sigma_{11} v_1 v_2^T \Sigma_{22} v_2}} \tag{4.2}$$

where Σ_{ij} is the covariance matrix of Z_i and Z_j:

$$\Sigma_{ij} = \frac{1}{N-1} \bar{Z}_i \bar{Z}_j^T, \tag{4.3}$$

where $\bar{Z}_i$ and $\bar{Z}_j$ are the centered matrices of Z_i and Z_j, respectively.

Vectors v_1^* and v_2^* are the two linear transformation vectors that map Z_1 and Z_2 to a maximally correlated new space. Since such correlation function is invariant to scaling of transformation vectors v_1 and v_2, the objective function can be written as a constraint optimization problem as follows:

$$(v_1^*, v_2^*) = \underset{v_1^T \Sigma_{11} v_1 = v_2^T \Sigma_{22} v_2 = 1}{\arg\max} v_1^T \Sigma_{12} v_2^T \tag{4.4}$$

We need to find other transformation vectors which produce projections uncorrelated with previous ones. The constrained problem to find all transformation vectors is

$$(V_1^*, V_2^*) = \underset{V_1^T \Sigma_{11} V_1 = V_2^T \Sigma_{22} V_2 = I}{\arg\max} trace(V_1^T \Sigma_{12} V_2) \tag{4.5}$$

where matrices V_1 and V_2 contain transformation vectors as columns. Note that there are several ways to solve such optimization problems. It is shown in [38] that the sum of the largest singular values of $R = \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$ gives the maximal value of (Equation 4.5), and the corresponding eigenvectors are the optimal projection directions. The sum of all singular values can be estimated by the Frobenius matrix norm of R

$$C(Z_1, Z_2) = \|R\|_F = \sqrt{trace(R^T R)} \tag{4.6}$$

All covariance matrices in (Equation 4.5) are regularized by a small positive number r to ensure that the matrices are positive definite

$$\Sigma_{ij} = \frac{1}{N-1} \bar{Z}_i \bar{Z}_j^T + rI. \tag{4.7}$$

We define the function G based on the two criteria of inter-class separation and intra-class variation to learn transformations that lead to a discriminative feature space. Inter-class separation measures how close instances from different classes are to each other. Intra-class variation measures how close instances from the same class are to each other. Generally, intra-

class variation should be minimized while inter-class separation should be maximized to obtain a discriminative feature space.

The intra-class criterion S_W (also referred to as a within-class scatter matrix) for a set of L labelled samples from view v, $Z_v = \{z_v^1, z_v^2, ..., z_v^L\}$, is defined as

$$S_W(Z_v) = \frac{1}{L} \sum_{i=1}^{|C|} \sum_{z_v^j \in C_i} (z_v^j - m_v^i)(z_v^j - m_v^i)^T \tag{4.8}$$

where $|C|$ is the number of classes, and $z_v^j = f(d_v^j)$ is the j_{th} instance of view v in the new space. Variable m_v^i denotes the mean of the samples from class i for view v.

The inter-class criterion S_B (also referred to as the between-class scatter matrix) for the same set of samples Z_v can be defined as follows

$$S_B(Z_v) = \frac{1}{2L^2} \sum_{i,j=1}^{|C|} L_i L_j (m_v^i - m_v^j)(m_v^i - m_v^j)^T$$

where L_i is the number of labelled samples from class i. These two criteria can be merged into a single optimization problem as:

$$G(Z_v) = Tr\{(S_W(Z_v) + S_B(Z_v) + rI)^{-1} S_B(Z_v)\}$$

where $G(Z_v)$ measures the discriminiveness of the learned space for labelled samples of view v. The parameter r is a regularization parameter to increase the stability of the inverse op-

eration. Maximizing the function $G(Z_v)$ leads to maximizing $S_B(Z_v)$ and minimizing $S_W(Z_v)$ simultaneously to obtain a discriminative feature space.

4.3.3 Optimization

To optimize the objective function L, we find the optimal values of all parameters for both networks (Θ_1 and Θ_2) using Stochastic Gradient Descent (`SGD`). To use `SGD`, we split the samples into some labeled and unlabeled mini-batches. In labeled batches, labeled samples from each class are present proportional to their ratio in the whole data.

We estimate the gradient of L with respect to the outputs of networks Z_1 and Z_2 to use the backpropagation technique. The backpropagation algorithm estimates other gradients to update the networks' parameters Θ_1 and Θ_2.

If the singular value decomposition of matrix R in function C is $R = UDV$, then the gradient of function C with respect to Z_1 can be estimated as follows:

$$\frac{\partial C(Z_1, Z_2)}{\partial Z_1} = \frac{1}{N-1}(2\nabla_{11}\bar{Z}_1 + \nabla_{12}\bar{Z}_2) \tag{4.9}$$

where

$$\nabla_{12} = \Sigma_{11}^{-\frac{1}{2}} U V^T \Sigma_{22}^{-\frac{1}{2}}$$

$$\nabla_{11} = \frac{-1}{2}\Sigma_{11}^{-\frac{1}{2}} U U^T \Sigma_{11}^{-\frac{1}{2}}$$

N denotes the total number of samples in the batch. Similar expressions hold for the gradient with respect to Z_2. More detail on calculating this gradient can be found in [4].

Calculating the gradient of the $G(Z_i)$ is not trivial. Similar variants of $G(Z_i)$ have been already investigated in other works [28, 105], but in most cases they tackled them by formu-

lating the optimization problem as a general eigen decomposition problem. We avoided such reformulation as we found out in our experiments that it increases the training instability of the neural networks. Therefore, we optimize $G(Z_i)$ without any reformulation by following [94].

We denote $S_W(Z_v)$ and $S_B(Z_v)$ as S_W^v and S_B^v respectively in the following derivations. If sample $z_v^n \in C_k$, then gradient of scatter matrices S_W^v and S_B^v for view v are defined as (Equation 4.10) and (Equation 4.11).

$$\frac{\partial S_W^v[i,j]}{\partial Z_v[n,p]} = \frac{1}{L}\begin{cases} 0 & \text{if} \quad \text{i} \neq n \quad \text{and} \quad \text{j} \neq n \\ Z_v[j,p] - m_v^k[j] & \text{if} \quad \text{i} = \text{n} \quad \text{and} \quad \text{j} \neq n \\ Z_v[i,p] - m_v^k[i] & \text{if} \quad \text{i} \neq n \quad \text{and} \quad \text{j} = \text{n} \\ 2(Z_v[n,p] - m_v^k[n]) & \text{if} \quad \text{i} = \text{n} \quad \text{and} \quad \text{j} = \text{n} \end{cases} \tag{4.10}$$

$$\frac{\partial S_B^v[i,j]}{\partial Z_v[n,p]} = \frac{1}{L^2}\sum_{s=1}^{|C|} L_s \begin{cases} 0 & \text{if} \quad \text{i} \neq n \quad \text{and} \quad \text{j} \neq \text{n} \\ m_k[j] - m_v^s[j] & \text{if} \quad \text{i} = \text{n} \quad \text{and} \quad \text{j} \neq \text{n} \\ m_k[i] - m_v^s[i] & \text{if} \quad \text{i} \neq n \quad \text{and} \quad \text{j} = \text{n} \\ 2(m_k[n] - m_v^s[n]) & \text{if} \quad \text{i} = \text{n} \quad \text{and} \quad \text{j} = \text{n} \end{cases} \tag{4.11}$$

Defining $S_T = S_B + S_W$, then the gradient of the discriminative objective function $G(Z_v)$ is estimated as

$$\begin{aligned} \frac{\partial G(Z_v)}{\partial Z_v[n,p]} &= Tr\{(S_T^v)^{-1} S_B^v\} \\ &= \sum_{s=1}^{d} \frac{\partial (S_T^v)^{-1}[s,s]}{\partial Z_v[n,p]} S_B^v[s,s] + (S_T^v)^{-1}[s,s] \frac{\partial S_B^v[s,s]}{\partial Z_v[n,p]} \end{aligned} \tag{4.12}$$

As we can have the following:

$$\frac{\partial (S_T^v)^{-1} S_B^v}{\partial Z_v} = -(S_T^v)^{-1} \frac{\partial S_B^v}{\partial Z_v} (S_T^v)^{-1} \tag{4.13}$$

We rewrite the gradient $\frac{\partial G(Z_v)}{\partial Z_v}$ by using (Equation 4.10) and (Equation 4.11) as

$$\begin{aligned} \frac{\partial G(Z_v)}{\partial Z_v} =& \frac{2}{L^2}[S_T^{-1}(Z_v) S_B(Z_v) - I] S_T^{-1}(Z_v) (\sum_{j=1}^{|C|} M_v^j) \\ & - \frac{2}{L} S_T^{-1}(Z_v) S_B(Z_v) S_T^{-1}(Z_v)(Z_v - M_v) \end{aligned} \tag{4.14}$$

where

$$\begin{aligned} M_v =& (m_v^q . 1^T)_{1 \le q \le |C|} \\ M_v^j =& (L_j . (m_v^j - m_v^q) . 1^T)_{1 \le q \le |C|} \end{aligned} \tag{4.15}$$

The detail of the gradients for the discriminative loss function can be found in [94]. The gradient of the total objective function is used to train both the networks simultaneously with the backpropagation algorithm. It is necessary to train the model with mini-batches because the objective function is defined on the properties of whole space, not just a single instance. Therefore at each step, we need a batch of sample to optimize the objective function.

4.4 Experiments

In this section, we present the experimental evaluation and analysis of `MDNN`. All experiments are performed in cross-view classification setting, however, it can be extended to other tasks such as cross-modal image and text retrieval [50].

4.4.1 Datasets

We evaluate the proposed algorithm on the following four datasets. A summary of the datasets is presented in Table VIII.

`Noisy MNIST`: It is a noisy version of the well-known `MNIST` dataset that contains images of handwritten digits. Following [102], a two-view version of `MNIST` for evaluating multi-view problem has been created. This was accomplished by rotating and adding random noise to the images of the dataset. Each image was rotated by a randomly sampled angle from a uniform distribution between $-pi/2$ and $+pi/2$. The resulting images were used as the first view. For each image, another image from the same class was selected randomly as the second view. Uniform noise samples in the range $[0, 1]$ were also added to each pixel of the images in the second view. The `Noisy MNIST` dataset contains 70K grayscale images of digits 0 to 9. The split of $50,000/10,000/10,000$ is used in the experiments for train/validation/test. Two examples of this dataset are shown in Figure 6.

Web Knowledge Base (`WebKB`)[1]: It is a collection of 1,051 web documents crawled from four universities [91]. The data has two classes: course or non-course web pages. Each document has two views: 1) the textual content of the web page and 2) the anchor text on the links pointing to the web page.

`CNN` and `FOX`: These two datasets were crawled from `CNN` and `FOX` web news [86]. The category information extracted from their RSS feeds are considered as their class label. Each

---

[1] **http://vikas.sindhwani.org/manifoldregularization.html**

TABLE VIII: Summary of the datasets: Noisy MNIST, WebKB, FOX, CNN.

Dataset	# Instance	# Feature	# Class
Noisy MNIST	70000	$784 + 784$	10
WebKB	1051	$3000 + 1840$	2
FOX	1523	$1143 + 996$	4
CNN	2707	$7989 + 996$	7

instance is represented in two views: the text view and image view. Titles, abstracts, and text body contents are considered as the text view data (view 1), and the image associated with the article is the image view (view 2). All text is stemmed by Porter stemmer, and l2-normalized TF-IDF is used as text features. Processed data samples in CNN and FOX datasets have $1,143$ and $7,980$ features respectively. Also, seven groups of color features and five textural features are used for image features [86], which results in 996 features for both datasets.

4.4.2 Baselines

We compare the performance of MDNN with the state-of-the-art of multi-view representation learning techniques. We use the same cross-view classification setting as [102, 103]. From methods that do not use deep neural network, we compare MDNN to linear Canonical Correlation Analysis (CCA) and Kernel CCA (KCCA) as the most commonly used techniques for representation learning in multi-view problems [102,109]. Although CCA finds linear transformations, it is still widely used because of its speed and simplicity. As traditional Kernel CCA is not scalable, we use FKCCA [62] method which is an approximation of the real Kernel CCA definition.

Figure 6: Two examples from the multi-view `Noisy MNIST`. Left images are from view one, and right images are their corresponding samples from the second view.

Among all the DNN based approaches, `Deep CCA` (DCCA) [4] is used in the experiments because of the better performance than other `DNN`-based algorithms [102]. None of these `CCA` based techniques use the label information, and all are categorized as unsupervised feature reduction techniques.

Also, two approaches which consider label information are also selected as baselines, Linear Discriminant Analysis (`LDA`) [47] and its neural network variant: `Deep LDA` [28]. These approaches are not designed for multi-view problems, but they are selected because they use labeled data to learn the new representation. Therefore they are applied on each view independently, and cannot use inter-view relation between views.

4.4.3 Experimental Settings

We perform all the experiments in cross-view learning, that is used in [102,109] for evaluating representation learning techniques in multi-view problems. In this setting, all views are available during the representation learning but one is missing during the testing process. All the methods in the experiments use both primary and complementary views in the training process to learn a common feature space. After learning representation, primary view is mapped to the new learned space. Then a linear Support Vector Machine (`SVM`) classifier [98] is trained on the new representation to evaluate it in a classification task. We would like to emphasize that the aim of this work is to present a new representation for multi-view setting. Therefore, we selected `linear SVM` as the classifier instead of a more complicated method for classification. In this way, we can evaluate the effectiveness of the representation learning more accurately.

All the parameters are selected to obtain the best performance in cross-validation process. All neural network based models are trained for 150 epochs. All samples are distributed randomly over the batches proportionally to their class size.

Regularization parameter α is selected from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for all datasets. Trade-off parameter λ is selected from $\{10^{-1}, 1, 10^{2}, 10^{3}, 10^{4}\}$ for each dataset separately. Regularization r is set to 10^{-4}. Representation size are selected as the number of classes for each dataset except for `WebKB` which is 10. These sizes may not give the best performance possible for `MDNN`, but they are set as the number of classes for all models to have a fair comparison among all techniques. Parameter C of the `SVM` is also selected from $\{10, 1, 10^{-1}, 10^{-2}, 10^{-3}\}$ by cross-validation.

Networks with the same architecture consisting of 3 hidden layers with the same number of hidden nodes are used for both views. The only exception is the network for WebKB which has 2 hidden layers instead of 3. Number of hidden nodes is selected as 1024, 128, 512, 512 for Noisy MNIST, WebKB, FOX, and CNN datasets, respectively. We use a variant of SGD, called Adam [53], to optimize the neural networks. All the parameters of Adam are set as the its paper recommends. The architecture of the networks for all the neural network based models including Deep CCA, Deep LDA, and MDNN are defined the same to have fair comparisons.

4.4.4 Performance Evaluation

We evaluate the effectiveness of the new representations learned by MDNN on cross-view classification tasks. All the results are reported for the primary view which is the only available view during the test.

The classification accuracy of different baselines on datasets Noisy MNIST, WebKB, FOX, and CNN are reported in Tables Table IX and Table X. Results are reported for different numbers of labeled samples to show the effectiveness of the proposed algorithm in semi-supervised settings. The column labeled as 'All' indicates the case where the label information for all samples is available. The best performance for each case is shown in bold. As it can be observed, MDNN outperforms all the other baselines in most cases.

The differences of MDNN's accuracies are more significant comparing to others in cases with fewer labeled samples. It shows the effectiveness of the proposed algorithm in exploiting labeled information which helps the model in semi-supervised settings. It should be considered that none of the current approaches can exploit both the labeled and unlabeled data together.

TABLE IX: Performance of different methods trained with various number of labeled examples on Noisy MNIST and WebKB in terms of accuracy.

Dataset	Noisy MNIST				WebKB			
# of labeled samples	200	400	600	*All*	50	100	150	*All*
MDNN	75.00	79.64	80.32	**97.34**	**86.58**	**93.29**	**94.46**	**96.50**
Deep CCA	80.17	85.85	77.91	84.83	80.17	80.46	93.00	81.92
Deep LDA	61.42	73.29	78.61	96.83	83.38	86.88	93.00	95.62
Linear CCA	69.02	72.70	73.99	76.13	70.17	71.62	77.35	83.50
LDA	40.05	40.27	25.65	76.79	57.43	61.51	62.23	94.75
Kernel CCA	**75.81**	**87.25**	**93.78**	94.21	77.46	79.59	80.17	82.79

TABLE X: Performance of different methods trained with various number of labeled examples on FOX and CNN in terms of accuracy.

Dataset	FOX				CNN			
# of labeled samples	125	250	375	*All*	250	500	750	*All*
MDNN	**73.62**	**84.25**	**90.94**	**97.63**	**77.07**	**81.02**	**81.94**	**82.34**
Deep CCA	72.73	79.03	80.67	88.82	41.47	45.73	57.50	59.30
Deep LDA	74.42	80.07	84.73	93.07	50.57	63.49	71.54	70.48
Linear CCA	71.75	77.16	78.34	82.57	37.02	41.23	40.18	45.45
LDA	70.66	78.74	81.69	87.09	67.85	69.43	73.91	75.86
Kernel CCA	71.25	72.63	71.06	78.13	38.20	39.92	40.84	49.84

The experiments also demonstrate that MDNN can also show superior results even for supervised settings where all data are labeled. It shows that the idea of combining inter-view correlation and intra-view discrimination can be effective even when label information is available for all samples.

MDNN demonstrates better accuracy compared to Deep CCA because it considers both label information and cross-view correlation when finding the projections; while deep CCA ignores

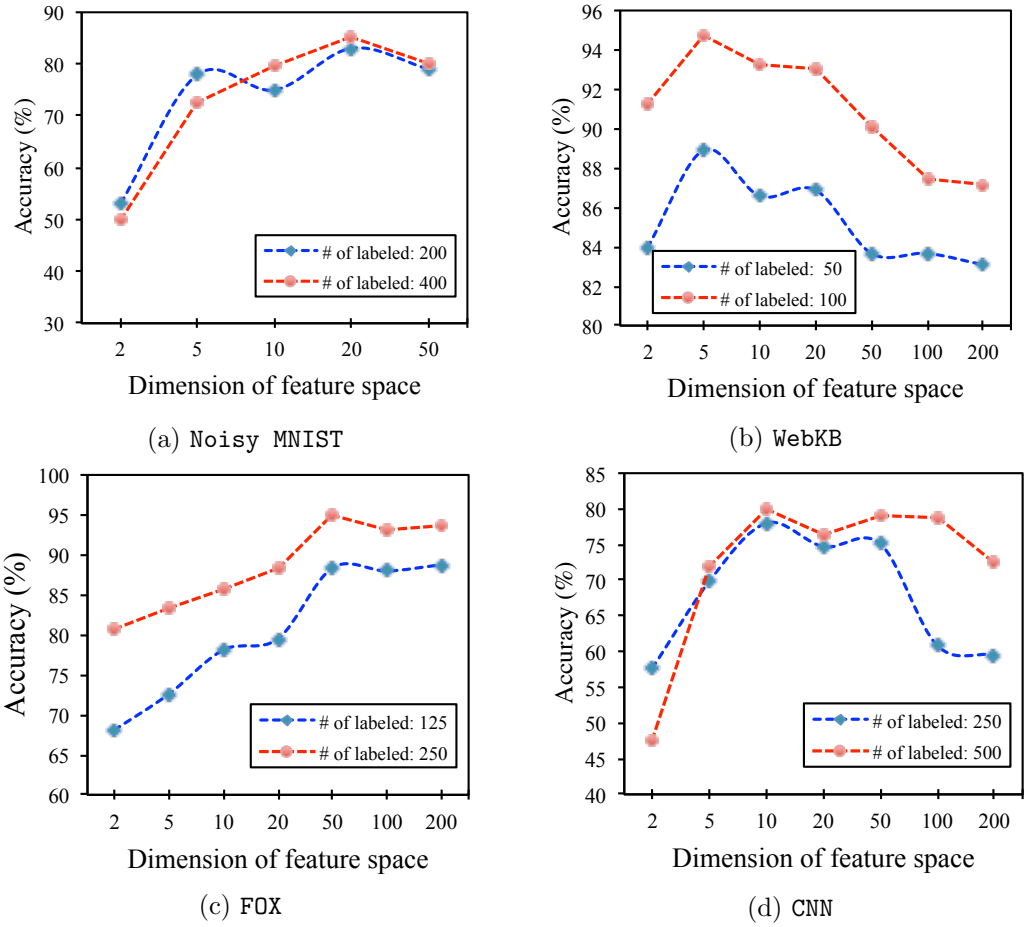
the available label information. The proposed MDNN attempts to produce more discriminative feature sets by leveraging label information into the mapping learning process. Simultaneous optimization of inter-class separation, intra-class variation, and cross-view correlation make the new representations more discriminative; therefore, prediction is easier.

Kernel CCA shows better results than MDNN in some cases of Noisy MNIST. It can be due to the simplicity of Noisy MNIST dataset. As it can be seen, just a few labeled samples are enough to get good results on this dataset.

4.4.5 Model Analysis

We investigate and explore the influence of the main parameter of MDNN, the size of new representation, on the classification task. In Figure 7, the accuracies of MDNN on all datasets are plotted for various sizes of space. As it can be seen, good results can get achieved with a small size of representation, and there is no need to learn a high dimensional space. A simple classification algorithm such as linear SVM can classify the samples in the new space efficiently. It shows the representation learning power of MDNN. Representation learning can make it feasible to work on high dimensional data for the algorithms which are not able to handle high dimensional data efficiently.

Additionally, it can be seen that having unnecessary large sizes for the output dimension can affect the performance. For most datasets, hidden output size close to the number of classes can be a good choice. Unnecessary large embedding size may reduce the performance. It can be the result of producing noisy information in higher dimensional space.



(a) Noisy MNIST (b) WebKB

(c) FOX (d) CNN

Figure 7: Accuracy of MDNN trained with two different numbers of labeled samples and various feature space size on (a) Noisy MNIST, (b) WebKW, (c) FOX and (d) CNN.

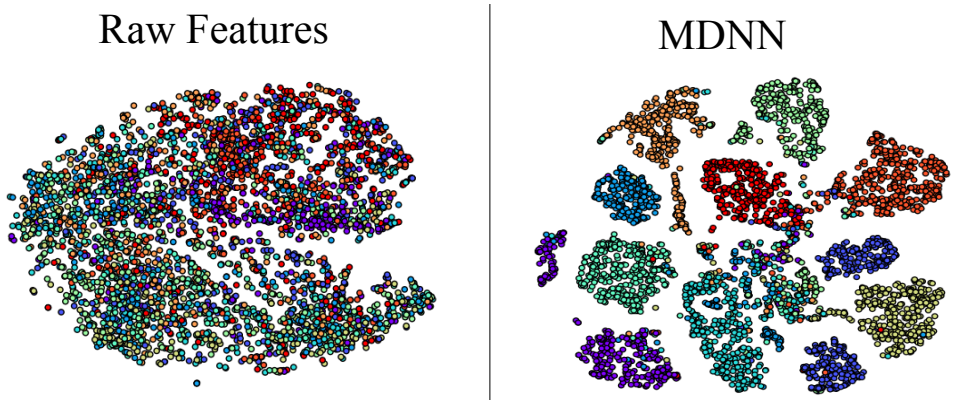


Figure 8: Visualization of randomly selected instances from `Noisy MNIST` dataset in a 2-dimensional space using `t-SNE`. They are mapped to the new space learned by `MDNN`. Color of the instances shows their class.

4.4.6 Subspace Analysis

In this section, the new subspace learned by `MDNN` is investigated and compared with the original feature space. 4000 instances of the training samples with new representation are selected randomly and visualized in 2-dimensional space in Figure 8. They are visualized using a dimensionality reduction algorithm called t-Distributed Stochastic Neighbor Embedding (`t-SNE` algorithm [67]. It is an unsupervised representation learning that is mostly used for visualizing features in a low-dimensional space. It learns mappings from the given feature space to a new space in which similarity of samples is preserved as much as possible. In other words, samples which are close or similar in the source feature space are likely to be close to each other in the new space. It is evident that `MDNN` produces a more discriminative space comparing to original feature space. It learns better representation that is owed to exploiting the label information.

CHAPTER 5

LEVERAGING SEMI-SUPERVISED LEARNING FOR FAIRNESS

5.1 Introduction

The rapid increase in automation of decision-making systems using machine learning approaches has raised significant concerns about the fairness of such models. Different studies have shown that these machine learning models which are designed to help the process of decision-making are not immune to social biases [10, 22]. There is a significant shift towards employing machine learning techniques in many sensitive real-world applications such as credit approval, loan applications, criminal risk assessment, university admissions, and online advertisement. With this new trend, it becomes crucial to consider other aspects and metrics for assessing a model beyond their accuracy. Among those aspects, fairness has gathered close attention in the community as we hope for building a socially responsible and inclusive system. Recently many machine learning algorithms have been proposed to address this problem and make the predictions of the learning algorithms fairer [2, 16, 40, 63, 111].

The naive approach for addressing the fairness problem in machine learning could be to remove or ignore the protected attributes such as sex, gender and age. However, this approach is not practical in many real-world applications [65] mainly because of the two following reasons: 1) there can exist some proxy features or correlation between other features and the sensitive attributes which may reveal them, and 2) there already exists some degree of bias in the labels

of the training data. On the other hand, in many applications, unlabeled data is abundant, and if appropriately leveraged, they also hold less bias compared to labeled data since models are not strongly affected by the labels of the labeled samples. In the same way, other paradigms like unsupervised learning or semi-supervised learning could be to lower degree sensitive to these biases in the data. Additionally, the lack of adequate labeled data poses a major challenge to many machine learning based applications and in some applications, creating a labeled dataset for training such models is expensive and time-consuming. Therefore, leveraging unlabelled data could be a potential solution to the lack of labeled data as well as fairness problems.

Semi-supervised learning approaches have shown promising results in tackling the aforementioned challenges by exploiting the unlabeled data to improve the performance of a classifier in terms of the accuracy [81]. The unlabeled data do not carry label information which can be a significant source of bias in training machine learning systems. The success of semi-supervised approaches in the improvement of model's performance through exploiting the unlabeled data, inspired us to study the effect of unlabeled data on the process of learning a fair classifier. In this paper, we propose a semi-supervised classification algorithm based on neural networks to tackle the fairness in machine learning. To the best of our knowledge, we are the first to propose and study the effect of semi-supervised learning on the fairness of a classifier using neural networks. Our proposed model, called `SSFair`, utilizes Pseudo-Labeling [60] approach to exploit unlabeled data to increase the accuracy and fairness of a classifier. Pseudo-labeling is one of the most common techniques for handling semi-supervised learning.

The proposed model is built with neural networks and can support any fairness measurement which can be defined or approximated as a differentiable function. Different criteria exist to measure fairness in machine learning. We have incorporated three of the most common measurements Demographic Parity, Equalized Opportunity, and Equalized Odds [40] into `SSFair`. We have evaluated `SSFair` on different measurements of fairness in semi-supervised settings and showed the effectiveness of the proposed algorithm to exploit the unlabeled data. We show experimentally that `SSFair` can benefit from unlabeled data to not just improve the accuracy but also improve the fairness of the classifier.

In the next section, an overview of the related works is given. In Section 5.3 definitions of fairness and related measurements are introduced . The proposed approach is presented in detail in Section 5.4. Experimental settings and results are given in Section 5.5.

5.2 Related Works

There are three main approaches proposed to tackle the fairness problem in machine learning, 1) pre-processing, 2) in-processing, and 3) post-processing approach.

In pre-processing approach, the goal is to learn a new representation of the data which is uncorrelated with the protected attributes [1,16,37,63,113]. This new representation can be used for any downstream task such as classification or ranking and any machine learning technique of choice. The main advantage of pre-processing approach is that it eliminates the need for making changes to the machine learning algorithms and therefore is very straightforward to use.

The second approach, in-processing, consists of the techniques that incorporate the fairness constraints into the training process. Most of the works on fairness in machine learning belong to this category [2, 51, 110, 111]. The in-processing algorithms usually address the problem by adding the fairness criterion to the learning algorithm's main objective function as a regularizer. This category is more flexible to optimize different fairness constraints, and the solutions using this approach are considered the most robust ones. Moreover, these category of approaches have shown promising results in terms of both accuracy and fairness.

The third approach is post-processing which aims to make changes on the output of the classifiers in order to satisfy the fairness constraint. One simple form of it is to find a threshold specific for each protected group and use it to control the fairness objective. Although this approach does not need any changes in the classifier, it is not very flexible in optimizing the trade-off between fairness and accuracy.

Our proposed model formulated as a semi-supervised learning based on neural network, falls under the second category, in-processing approaches. It aims at optimizing the fairness constraint during training the classifier. To the best of our knowledge, `SSFair` is the first semi-supervised algorithm based on neural networks introduced for tackling the fairness problem. There are a few works that employ neural networks to optimize the trade-off between fairness and accuracy. Most of these approaches employ adversarial optimization inspired by Generative Adversarial Networks (`GAN`) [35] to train a model for producing a fair representation or an output which is indistinguishable among all of the protected groups [18,64,68,101,114]. However, these methods are not capable of optimizing an arbitrary fairness constraint, at least not explicitly.

Alternatively, in [69] fairness problem is addressed by incorporating the fairness constraints explicitly into the optimization of the neural network during the training. The authors have added several fairness constraints into the loss function of the neural network as a regularization term. This algorithm only handles fully supervised learning setting and thus can not benefit from unlabeled data.

5.3 Fairness Measurements

Defining and measuring the concept of fairness for a machine learning algorithm is not trivial, and a variety of definitions exist to measure and quantify fairness. [40, 52]. These definitions are categorized into two main groups of individual fairness [30, 49] and group fairness [40].

The term of individual fairness is first introduced in [30] to refer to a fairness constraint which is focused on treating similar individuals as similar as possible. The fairness measurement or metrics defined in this category are based on the expectation that similar individuals should get treated similarly and the output of the machine learning algorithm should be close for similar inputs [52,113]. The main drawback of such constraints is the difficulty of defining their similarly metric function. An appropriate similarity function should be capable of ignoring the proxy features which may reveal individual's sensitive information. For this reason, individual fairness cannot be applied widely in real-world problems.

The second group, called group fairness or statistical fairness, is most commonly used in the literature. They divide the individuals or samples into sets of unprotected and protected (or privileged and unprivileged) based on sensitive attributes like race, gender, or age. Then they try to make some statistical measures (e.g. classification error, true positive rate, or false

positive rate) of the performance of the classifier or any other machine learning algorithm equal for both the protected and unprotected groups. The three most common definitions in this category are Demographic Parity, Equalized Opportunity, and Equalized Odds. Our `SSFair` approach can optimize for all of these three fairness objectives. These measurements are defined in Section 5.4 in detail.

There is no consensus on the best definition of fairness, and it is very task-dependent to decide which one to use. In some cases, there exists a trade-off between some of these fairness constraints. It is shown that some of these fairness constraints cannot get satisfied at the same time except in some degenerate or highly constrained special cases [55, 83].

5.4 Proposed Model

In semi-supervised settings, training data consists of a collection of labeled and unlabeled samples. Assume $\mathcal{D} = \left\{(X_i, a_i, y_i)\right\}_{i=1}^{N}$ is the training set consisting of N samples. For each sample i, X_i denotes the feature set, $y_i \in \{0, 1, u\}$ denotes the label, and $a_i \in \{p, n\}$ is the protected attribute which shows whether that sample belongs to the protected set (p) or not (n). Assume the valid values for labels are 0 for non-advantaged outcome, 1 for the advantaged outcome, or u for the unknown labels.

Our goal is to learn a binary classifier function $f(X; \Theta) : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by Θ to optimize two main objectives, the classification accuracy, and fairness. We would model the function $f(.)$ by a neural network. To achieve this goal we define the loss function of the model as:

$$\mathcal{J}(\mathcal{D}; \Theta) = \alpha \mathcal{J}_{\mathcal{C}}(\mathcal{D}; \Theta) + (1 - \alpha)\mathcal{J}_{\mathcal{F}}(\mathcal{D}; \Theta) + \beta \left\|\Theta\right\|_2 \tag{5.1}$$

where $\mathcal{J}_{\mathcal{C}}(\mathcal{D};\Theta)$ indicates the classification loss, and $\mathcal{J}_{\mathcal{F}}(D;\Theta)$ is the fairness loss which imposes fairness on the output of the model. Parameter α controls the trade-off between fairness and accuracy losses. Parameter β controls the regularization term $\|\Theta\|$ which is imposed on all of the networks' weights. Regularization is very important to prevent overfitting specially since limited labeled samples are available.

5.4.1 Classification Loss

The first part of the loss function, the classification accuracy loss $\mathcal{J}_{\mathcal{C}}(\mathcal{D};\Theta)$, is defined over the training samples as:

$$\mathcal{J}_{\mathcal{C}}(\mathcal{D};\Theta) = \sum\nolimits_{1\leq i\leq N} j_c(X_i;\Theta) \tag{5.2}$$

where $j_c(X_i)$ indicates the classification loss for sample X_i and is defined as the cross-entropy between the output of the learned function and the target label:

$$j_c(X_i;\Theta) = \mathbb{1}\{v_i = T\}(q_i \log \hat{y}_i + (q_i - 1)\log(1-\hat{y}_i)) \tag{5.3}$$

where $\hat{y}_i = f(X_i;\Theta), 0 < \hat{y}_i < 1$ indicates the output of the learned function for sample X_i and q_i is X_i's corresponding target label. Target label q_i is defined as the ground truth label y_i if X_i is labeled, while it is defined as $q_i = \mathbb{1}\{\hat{y} \geq 0.5\}$ for unlabeled samples. $v_i \in \{T, F\}$ indicates whether sample X_i should be considered in the learning process or not and will be defined below. $\mathbb{1}$ is an indicator function which zero-outs the samples whose v_i is not T.

We follow the Pseudo-Label approach [60] to handle the unlabeled samples. For all labeled samples, v_i is set to T. For unlabeled samples, only the ones with high confidence output should

get their v_i set to T and remain in the learning process. With a binary classifier, the output value $\hat{y}$ can be utilized to obtain the confidence of the prediction for sample X_i. Therefore v_i is defined as:

$$v_i = \begin{cases} T & \text{if} \quad y_i = 0 \; or \; 1 \\ T & \text{if} \quad y_i = u \; and \; (\hat{y}_i < (1-\lambda) \; or \; \hat{y}_i > \lambda) \\ F & \text{if} \quad otherwise \end{cases} \tag{5.4}$$

where $0 \leq \lambda \leq 1$ defines a threshold which controls the degree of confidence which is needed to consider a predicted label in the learning process.

5.4.2 Fairness Loss

The second term in the loss imposes fairness on the learned function. As discussed in Section 5.3, there is a variety of definitions for fairness and there is no consensus on which one is the best. Our approach is quite flexible in that it can work with any fairness objective as far as it is a differentiable function. This capacity to handle and optimize different definitions is considered a huge advantage for a fairness algorithm since it enables adapting the appropriate fairness definition based on the application. In this paper, the following three most common objectives in group fairness are studied with the proposed model.

5.4.2.1 Demographic Parity

Demographic Parity, also referred to as Statistical Parity, is one of the most common criteria for fairness [10,40]. It measures the difference between the probabilities of predicting advantaged output for the protected and unprotected groups and requires the decision of a classifier to be

independent of the protected attribute a. Its corresponding loss function denoted by $\mathcal{J}_{\mathcal{F}}^{\mathcal{DP}}(\mathcal{D};\Theta)$ is defined as:

$$\mathcal{J}_{\mathcal{F}}^{\mathcal{DP}}(D;\Theta) = |\mathbb{E}[f(X;\Theta|a=p)] - \mathbb{E}[f(X;\Theta|a=n)]| \tag{5.5}$$

with:

$$\mathbb{E}[f(X;\Theta|a=z) = \frac{\sum_{X_i \in D_{a=z}} f(X_i;\Theta)}{|D_{a=z}|} \tag{5.6}$$

where $D_{a=z}$ defines the subset of D where their protected attribute $a = z$.

Demographic parity is backed up by the "four-fifth rule" which recommends that the selection rate for the protected group should not be less than 80% of the unprotected group unless there exists some business necessity [14]. A selection rate of less than 80% can have an adverse impact on the unprotected group. One of the flaws of this measurement is that by selecting qualified samples from the unprotected group while randomly selecting from the protected group, we may achieve high fairness by this measurement [40]. Moreover, it ignores any correlation between the protected attribute a and output prediction y.

5.4.2.2 Equalized Opportunity

This measurement is focused on the fairness for the advantaged outcome. It measures the difference between the probabilities of predicting advantaged output for the protected and unprotected groups while focusing on the individuals with advantaged ground truth [40]. Its corresponding loss function denoted by $\mathcal{J}_{\mathcal{F}}^{\mathcal{ODD}}(\mathcal{D};\Theta,k)$ is defined as the following with $k = 1$:

$$\mathcal{J}_{\mathcal{F}}^{\mathcal{OPP}}(D;\Theta,k) = |\mathbb{E}[f(X;\Theta|a=p,y=k)] - \mathbb{E}[f(X;\Theta|a=n,y=k)]| \tag{5.7}$$

with:

$$\mathbb{E}[f(X;\Theta|a=z,y=k)] = \frac{\sum_{X_i \in D_{a=z} \cap D_{y=k}} f(X_i;\Theta)}{|D_{a=z} \cap D_{y=k}|} \tag{5.8}$$

where $D_{y=k}$ defines the subset of D with label attribute $y = k$.

5.4.2.3 Equalized Odds

This constraint is a more strict version of Equalized Opportunity which focuses on both the groups with advantaged ground truth and non-advantaged ground truth [40]. It can be defined as the following:

$$\mathcal{J}_{\mathcal{F}}^{\mathcal{ODD}}(D;\Theta) = \mathcal{J}_{\mathcal{F}}^{\mathcal{OPP}}(D;\Theta,k=0) + \mathcal{J}_{\mathcal{F}}^{\mathcal{OPP}}(D;\Theta,k=1) \tag{5.9}$$

It is considered a more strict criterion than Equalized Opportunity as it requires for both $y = 1$ and $y = 0$. It enforces the accuracy to be equally high for all of the outcomes while Equalized Opportunity focuses on the advantaged outcome.

5.4.3 Model and Training

The classifier function $f(.)$ is modelled by a multi-layer perception (`MLP`) neural network. The whole model is trained using backpropagation with respect to the loss function in Equation 5.4. Given a set of N samples, we optimize the model using Adam [53] optimization

Procedure 2: Training Procedure of `SSFair`

Input: Training set: N Samples
$\mathcal{D} = \left\{(X_i, a_i, y_i) |\ y^i \in \{0, 1, u\}\ and\ a_i \in \{p, n\}\right\}_{i=1}^{N}$
Number of epochs: E
Batch size: m.
Confidence degree: λ
Output: Model's parameters: Θ
$B = \frac{|\mathcal{D}|}{m}$; // number of batches
for $t = 1, 2, \cdots, T$ **do**
  Shuffle all the samples;
  Partition the training data $\mathcal{D}$ into B batches;
  for $b = 1, 2, \cdots, B$ **do**
    Feedforward propagation of the b^{th} batch through the network;
    Calculate v for all samples in the batch by Equation 5.4 with confidence degree λ;
    Calculate the classification loss ${\mathcal{J}_{\mathcal{C}}}^t$ according to Equation 5.2;
    Calculate the fairness loss ${\mathcal{J}_{\mathcal{F}}}^t$ according to one of the losses in Equation 5.5, Equation 5.7 or Equation 5.9;
    Estimate the total loss $\mathcal{J}^t$ according to Equation 5.1 over the batch b;
    Calculate the gradients by backpropagation;
    Update all the parameters of the neural network (Θ) using Adam;
  end
end
return Θ;

technique over shuffled mini-batches from the data. The overall training procedure of `SSFair` is illustrated in Algorithm 2.

5.5 Experiments

In this section, we evaluate and study our proposed model for the fairness problem. We provide experimental results to support our claim that employing our semi-supervised approach based on neural networks improves the accuracy and fairness for classification task.

5.5.1 Dataset

We evaluate our proposed model on UCI Adult Income Dataset (`ADULT`) [57, 58] and study the task of predicting whether a person makes more than $50K$ or not. This is one of the most commonly used benchmarks for evaluating classification approaches for fairness. The proportion of high income individuals across the two groups of men and women are not equal, and therefore there is no demographic parity in the dataset.

The dataset has 12 features including categorical and continuous features. The detailed list of the features is presented in Table XI. Categorical features are encoded using one-hot encoding. The age feature is bucketized at the boundaries [18, 25, 30, 35, 40, 45, 50, 55, 60, 65]. The "Sex" feature is considered as a protected feature. We have also filtered out the samples with missing values. The post-processed dataset contains 45222 samples with 112 features. We randomly chose 70% of the samples for the train set and left the rest for the test set.

5.5.2 Experimental Setting

The hyperparameters of our proposed algorithm are tuned with validation on a randomly selected 20% of the training data. After setting the hyperparameters, the model is trained on the full training set. Eventually, the results on the test data are reported in the experiments.

In the experiments, for `SSFair` and the baseline (`Manisha et al.` [69]), a Multilayer Perceptron (`MLP`) neural network with 1 hidden layer of size 32 is used to model the function $f(X)$. Rectified Linear Unit (`ReLU`) activation is used for the outputs of the hidden layer. Since the task is binary classification, we use sigmoid function as the activation function on the last layer and get the final output as the result of that. A dropout layer with a dropout

TABLE XI: The list of the features of ADULT dataset with their descriptions.

Attribute Name	**Category**	**Comment**
Annual Income	Categorical	Target variable - whether income> $50K$\$ or not
Sex	Categorical	Protected feature
Age	Continuous	The age of the person
Work Class	Categorical	Type of the person's work
Final Weight	Continuous	Weight of the demographic which the person belongs to
Education	Categorical	Maximum level of the education the person earned
Education Number	Continuous	Number of years the person spent for education
Marital Status	Categorical	Marital situation of the person
Occupation	Categorical	Job of the person
Race	Categorical	Race of the person
Capital Gain	Continuous	Gain though investing
Capital Loss	Continuous	Loss though investing
Hours per Week	Continuous	Number of hours the person works in a week
Native Country	Categorical	Originality of the person

rate of 20% is used after the hidden layer. The regularization parameter β is selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1.0\}$ for each experiment based on the results of the validation process. Finally, the confident degree parameter λ is set to 0.99 for SSFair.

We use the Adam optimizer [53] to train the models. We choose the learning rate of 10^{-3} and use the default values recommended in [53] for the other parameters of the optimizer. Training the neural networks is done by running Adam over 1000 epochs of training data, when using shuffled mini-batches of size 512.

5.5.3 Experimental Results

In this section, we present the results of our experiments on the ADULT dataset to demonstrate the effectiveness of our semi-supervised learning approach for the fairness problem.

5.5.3.1 The Effect of Unlabeled Data on Accuracy and Fairness

We would like to verify that using unlabeled data can help our algorithm to improve on both aspects of accuracy and fairness. We performed experiments by increasing the number of unlabeled samples while keeping the number of labeled samples fixed to 100 to investigate the effect of adding unlabelled data. We have experimented with three different values of $\{0.001, 0.0025, 0.005\}$ for parameter α.

The plot of fairness loss versus the number of unlabeled samples is illustrated in Figure 9. For calculating the fairness loss, the output of the classifier ($\hat{y}$) is binarized with the threshold of 0.5 to provide a binary outcome. Demographic Parity is selected as the fairness loss in this part. As these plots suggest, fairness loss improves as we increase the size of the unlabeled set (note that higher fairness is achieved with fairness loss is lower). This experiment verifies that fairness in our model can benefit from unlabeled data and therefore our approach has been successful in utilizing unlabeled data to improve fairness.

Moreover, we paid special attention to the existing trade-off between accuracy and fairness as well. Particularly, we were interested in understanding whether the improvement in fairness by increasing the size of unlabeled data could be a result of potential losses on the accuracy. To understand this effect, the plot of accuracy versus the number of unlabeled samples is also illustrated in Figure 10. As it is clear from the plots, the accuracy of the classifier increases as we grow the number of unlabeled samples as well. This result validates that our approach provides a solution for using additional unlabeled data to improve both factors of accuracy and fairness.

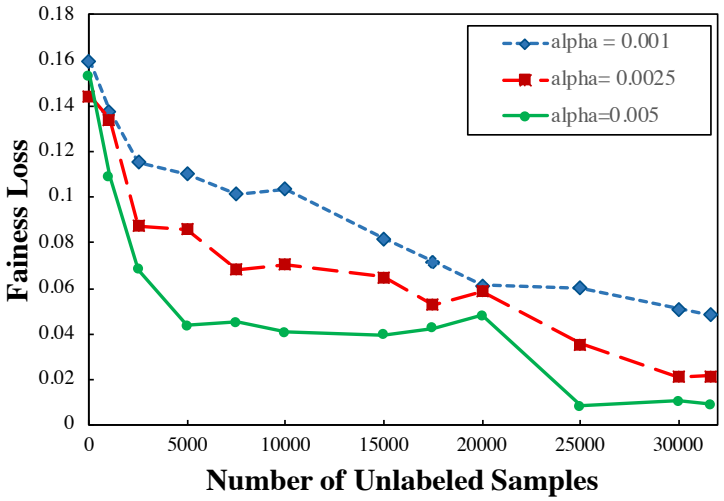


Figure 9: The effect of the number of the unlabeled samples on fairness loss (Demographic Parity).

5.5.3.2 Comparison Against Fully Supervised Approach

In this part, we demonstrate the benefit of our semi-supervised learning approach for the fairness problem versus a fully supervised model. We compare the results of our work with the model proposed by `Manisha et al.` [69] which is a model based on neural networks to address the fairness problem. To the best of our knowledge, it is the only work done on the trade-off between fairness and accuracy using neural networks. This model [69] is fully supervised and is only trained on the labeled samples. The experiments are performed with all of the three fairness objectives introduced in Section 5.4. Moreover, we experimented with a varying number of labeled samples (100, 200, and 300). For experiment with n labeled samples, we randomly chose n samples from the training set and kept their ground truth label while we changed the label of the other samples to u.

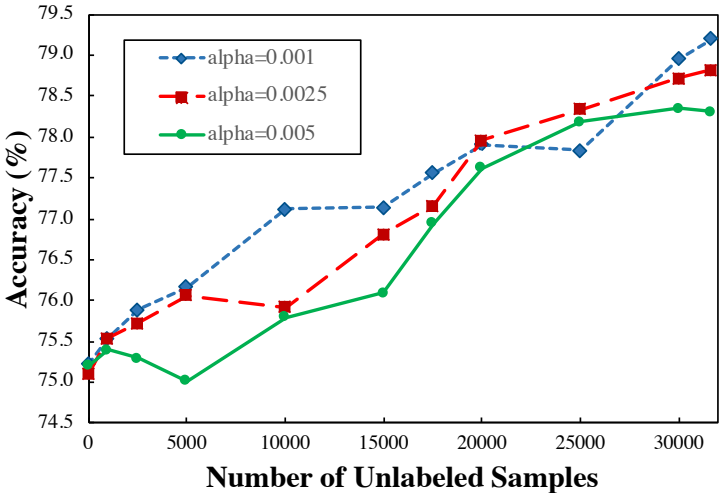
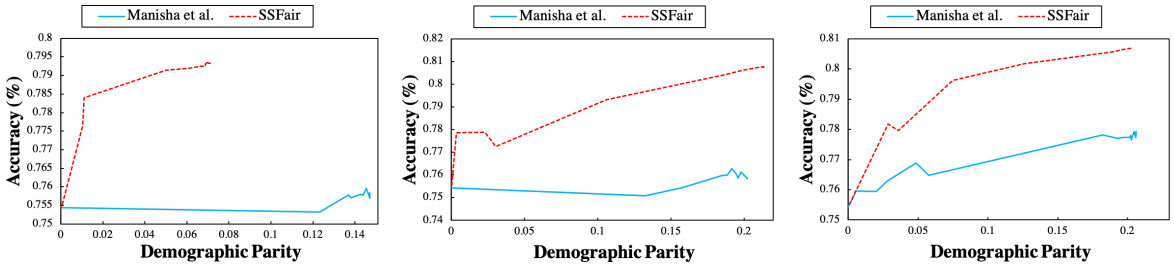


Figure 10: The effect of the number of the unlabeled samples on accuracy.

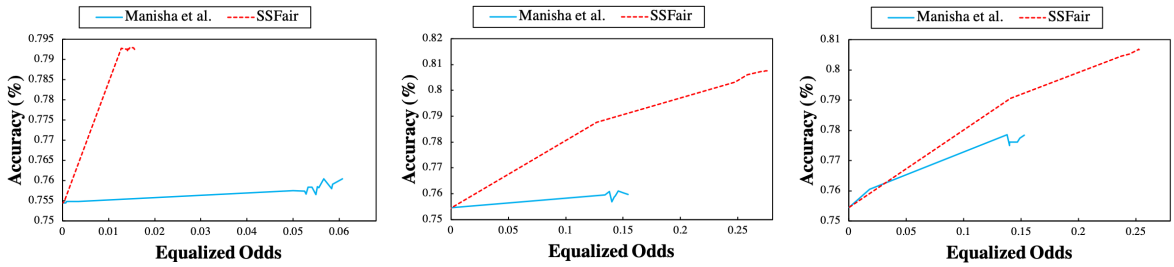
The results for Demographic Parity, Equalized Opportunity, and Equalized Odd losses are illustrated in Figure 11, Figure 12 and Figure 13 respectively. Different points on the curves are obtained by using different values for parameter α which is varied from 10^{-7} to 10^{4} to impose different levels of fairness on the classifier. Generally, there exists a trade-off between accuracy and fairness and parameter α controls this trade-off: increasing α would result in decreasing the accuracy while increasing the fairness. For each value of α, the experiment is repeated five times and the averaged results are reported.

Comparing two algorithms, one which can produce higher accuracy while maintaining the same level of fairness loss is considered the superior one. As it is evident from the results, `SSFair` provides higher accuracy for the same level of fairness loss compared to the approach of `Manisha et al.` This conclusion is consistent for all of the three fairness measurements,



(a) 100 labeled samples (b) 200 labeled samples (c) 300 labeled samples

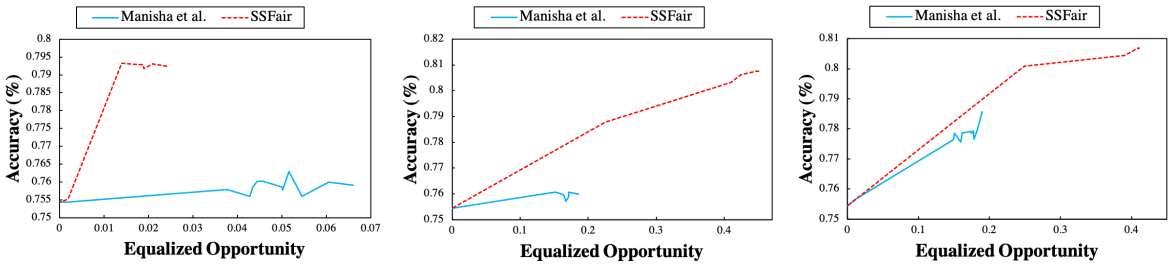
Figure 11: The trade-off between the Demographic Parity loss and the accuracy of `SSFair` compared to `Manisha et al`. The number of labeled samples is 100, 200, and 300 in 11a, 11b, and 11c, respectively.



(a) 100 labeled samples (b) 200 labeled samples (c) 300 labeled samples

Figure 12: The trade-off between the Equalized Opportunity loss and the accuracy of `SSFair` compared to `Manisha et al`. The number of labeled samples is 100, 200, and 300 in 12a, 12b, and 12c, respectively.

suggesting the effectiveness of exploiting unlabeled data by using semi-supervised learning for the fairness problem. It is worth noting that the effect of using unlabeled data is more evident in cases with fewer labeled samples, which indicates that this approach is most helpful in scenarios with scarce labeled data. Our understanding of this behavior is that since unlabeled data does not include any label information, they do not hold biased information for the labels either. Therefore, they can be beneficial not only to the accuracy but also to the fairness of the classifier.



(a) 100 labeled samples (b) 200 labeled samples (c) 300 labeled samples

Figure 13: The trade-off between the Equalized Odds loss and the accuracy of `SSFair` compared to `Manisha et al`. The number of labeled samples is 100, 200, and 300 in 13a, 13b, and 13c, respectively.

Our experiments show that `SSFair` is capable of exploiting the structure and information of unlabeled data to increase the accuracy and fairness compared to a fully supervised model.

CHAPTER 6

CONCLUSION

(Part of the chapter was previously published in [77–79].)

In this thesis, we explored semi-supervised learning using deep neural networks to tackle a variety of problems. We studied and proposed solutions for three research problems: (i) verification problem, (ii) multi-view learning, and (iii) fairness. The effectiveness of the proposed algorithms got evaluated by extensive experiments on various datasets. The main contributions are summarized as follows:

1. Benefiting from the salient structures hidden in the unlabeled data and the ability of deep neural networks in nonlinear function approximation, we proposed semi-supervised deep SEmi-supervised VErification Network (`SEVEN`) for verification tasks. It benefits from two learning components of an autoencoder and a discriminative one in a unified model. These two components are simultaneously trained which lead them to closely interact and influence each other. Experiments demonstrated that `SEVEN` outperforms other state-of-the-art deep semi-supervised techniques in a wide spectrum of verification tasks.

2. We presented a semi-supervised learning model, named `VerVAT`, that learns a distance metric for verification tasks whose training samples consists of negative, positive or unknown pairs. It exploits the unlabeled and labeled data in a joint manner to learn a discriminative feature space. The proposed model is the first verification model for semi-supervised setting which benefits from Virtual Adversarial Training to learn a robust and smooth embedding

space. The experiments demonstrated the effectiveness of the proposed algorithm. It outperformed state-of-the-art deep semi-supervised learning approaches for verification tasks on all the experimented datasets.

3. We introduced a semi-supervised deep neural network model, called `MDNN`, to learn discriminative representations for multi-view problems when labels for some instances are not available. To achieve this, the proposed model maximizes between-class separation and minimizes within-class variation to make the new space discriminative. It benefits the inter-view correlation to exploit the information in unlabeled data. Our model is capable of exploiting the information in both the labeled and unlabeled data in a unified learning process. To the best of our knowledge, the proposed `MDNN` is the first deep network model that learns a common subspace with such properties for semi-supervised multi-view problems. The experimental results demonstrated the effectiveness of `MDNN` in learning discriminative feature spaces and also benefiting from the unlabeled data.

4. We proposed a classifier based on neural networks for semi-supervised learning to tackle the fairness problem. The proposed model, named `SSFair`, benefits Pseudo-labeling approach to exploit the information in the unlabeled data. We evaluated and studied the effect of unlabelled data on learning a fair classifier, and showed experimentally that unlabelled data can be beneficial not just for accuracy but also for fairness. `SSFair` is evaluated on three fairness measurements Demographic Disparity, Equalized Opportunity, and Equalized Odds. In the experiments, it is shown that semi-supervised learning can achieve higher fairness and accuracy compared to the one which uses just the labeled data.

APPENDICES

INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE

TRANSFER OF COPYRIGHT AGREEMENT

Title of Article/Paper: SEVEN: Deep Semi-supervised Verification Networks

Publication in Which Article Is to Appear: 26th International Joint Conference on Artificial Intelligence

Author's Name(s): Vahid Noroozi, Lei Zheng, Sara Bahaadini, Sihong Xie, Philip S. Yu

Please type or print your name as you wish it to appear in print

(Please read and sign Part A only, unless you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign Part A and see item 6 under returned rights.)

PART A–Copyright Transfer Form

The undersigned, desiring to publish the above article/paper in a publication of the International Joint Conferences on Artificial Intelligence, Inc., hereby transfer their copyrights in the above paper to the International Joint Conferences on Artificial Intelligence, Inc. (IJCAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights currently exist or hereafter come into effect, and also the exclusive right to create electronic versions of the paper, to the extent that such right is not subsumed under copyright. The undersigned warrants that he/she is the sole author and owner of the copyright in the above paper, except for those portions shown to be in quotations; that the paper is original throughout; and that the undersigned's right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify IJCAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense IJCAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to IJCAI in the paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorney's fees incurred therein.

Returned Rights

In return for these rights, IJCAI hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. retain all proprietary rights (such as patent rights) other than copyright and the publication rights transferred to IJCAI;
2. personally reuse all or portions of the paper in other works of their own authorship;
3. make oral presentation of the material in any forum;
4. reproduce, or have reproduced, the above paper for the author's personal use, or for company use provided that IJCAI copyright and the source are indicated, and that the copies are not used in a way that implies IJCAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own World Wide Web page or ftp site. Such Web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the IJCAI electronic server (http://www.ijcai.org), and shall not post other IJCAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without IJCAI's written permission;
5. make limited distribution of all or portions of the above paper prior to publication.
6. In the case of work performed under U.S. Government contract, IJCAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes. In the event the above paper is not accepted and published by IJCAI, or is withdrawn by the author(s) before acceptance by IJCAI, this agreement becomes null and void.

VAHID NOROOZI — Author's (or Employer's Representative) Signature

05/23/2017 — Date

Employer for whom work was performed

Title (if not author)

European Association for Signal Processing

EUSIPCO-2019 Conference Copyright Agreement

It is required that Authors publishing at EUSIPCO-2019 in A Coruña, Spain, provide a transfer of Copyright to the European Association for Signal Processing ("EURASIP"), in the following just labeled "EURASIP". This empowers "EURASIP" on behalf of the Author to protect the Work and its image against unauthorized use and to properly authorize dissemination of the Work by means of printed publications, offprint, reprints, electronic files, licensed photocopies, microfilm editions, translations, document delivery, and secondary information sources such as abstracting, reviewing and indexing services, including converting the work into machine readable form and storing in electronic databases. Moreover, in order to be included in the IEEE Conference Publications Program, the written permission from a representative of the copyright owner must grant IEEE the nonexclusive, irrevocable, royalty-free worldwide rights to publish, sell and distribute the copyrighted work for the Conference named above and any content derived from the copyrighted work in any format or media without restriction, as stated below.

EDAS Paper number: # 1570533753

Title of contribution ("Work"): Virtual Adversarial Training for Semi-supervised Verification Tasks

Author(s): Vahid Noroozi, Sara Bahaadini, Lei Zheng, Sihong Xie, Philip S. Yu

Name of Publication: 27th European Signal Processing Conference, 2019 (EUSIPCO-2019)

The Author(s) hereby consents that the publisher appointed by EURASIP will publish the work. (1) The signatory(ies) on this form warrants all other authors/co-authors are properly credited, and generally that the author(s) has the right to make the grants made to "EURASIP" complete and unencumbered. The Author(s) also warrants that the Work is novel and has not been published elsewhere. The Author(s) furthermore warrant that the Work does not libel anyone, infringe anyone's copyright, or otherwise violate anyone's statutory or common law rights. (2) The Author(s) hereby transfers to "EURASIP" the copyright of the Work named above. "EURASIP" shall have the exclusive and unlimited right to publish the said Work and to translate (or authorize others to translate) it wholly or in part throughout the World during the full term of copyright including renewals and extensions and all subsidiary rights. (3) The copyright is not transferred to IEEE. However, the copyright owner grants IEEE the nonexclusive, irrevocable, royalty-free worldwide rights to publish, sell and distribute the copyrighted work for the Conference named above and any content derived from the copyrighted work in any format or media without restriction. (4) The Work may be reproduced by any means for educational and scientific purposes by the author or by others without fee or permission with the exception of reproduction by services that collect fee for delivery of documents. The Author(s) may use part or all of this Work or its image in any future works of his/her (their) own. In any reproduction, the original publication by "EURASIP" must be credited in the following manner: "First published in the Proceedings of the 27th European Signal Processing Conference (EUSIPCO-2019) in 2019, published by EURASIP", and such a credit notice must be placed on all copies. Any publication or other form of reproduction not meeting these requirements will be deemed to be unauthorized. (4) In the event of receiving any request to reprint or translate all or part of the Work, "EURASIP" shall seek to inform the author. This form is to be signed by the Author(s) or in case of a "work-made-for-hire", by the employer. If there is more than one Author, then either all must sign the Copyright Agreement, or one Author signs in consent for all, taking on full responsibility for the content of the publication.

Date: 06/16/2019

Full name(s): Vahid Noroozi

Signature(s): VAHID NOROOZI

Please return this form completed and signed on or before 18th June 2019 by electronic upload of a scanned copy in EDAS in order to ensure publication of your paper in EUSIPCO 2019 proceedings.

Title: Semi-supervised Deep Representation Learning for Multi-View Problems

Conference Proceedings: 2018 IEEE International Conference on Big Data (Big Data)

Author: Vahid Noroozi

Publisher: IEEE

Date: Dec. 2018

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

CITED LITERATURE

1. Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S.: Auditing black-box models for indirect influence. Knowledge and Information Systems, 54(1):95–122, 2018.

2. Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., and Wallach, H.: A reductions approach to fair classification. In International Conference on Machine Learning (ICML), pages 60–69, 2018.

3. Amini, S., Noroozi, V., Bahaadini, S., Philip, S. Y., and Kanich, C.: Deepfp: A deep learning framework for user fingerprinting via mobile motion sensors. In IEEE International Conference on Big Data (Big Data), pages 84–91. IEEE, 2018.

4. Andrew, G., Arora, R., Bilmes, J., and Livescu, K.: Deep canonical correlation analysis. In International Conference on Machine Learning (ICML), pages 1247–1255, 2013.

5. Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al.: End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine, 25(6):954, 2019.

6. Arora, R. and Livescu, K.: Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7135–7139, 2013.

7. Athiwaratkun, B., Finzi, M., Izmailov, P., and Wilson, A. G.: Improving consistency-based semi-supervised learning with weight averaging. arXiv preprint arXiv:1806.05594, 2, 2018.

8. Bahaadini, S., Rohani, N., Coughlin, S., Zevin, M., Kalogera, V., and Katsaggelos, A. K.: Deep multi-view models for glitch classification. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2017.

9. Bahaadini, S., Rohani, N., Katsaggelos, A. K., Noroozi, V., Coughlin, S., and Zevin, M.: Direct: Deep discriminative embedding for clustering of ligo data. In IEEE International Conference on Image Processing (ICIP), pages 748–752. IEEE, 2018.

10. Barocas, S. and Selbst, A. D.: Big data's disparate impact. Calif. L. Rev., 104:671, 2016.

11. Bell, S. and Bala, K.: Learning visual similarity for product design with convolutional neural networks. ACM Transactions on Graphics (TOG), 34(4):98, 2015.

12. Benton, A., Khayrallah, H., Gujral, B., Reisinger, D. A., Zhang, S., and Arora, R.: Deep generalized canonical correlation analysis. arXiv preprint arXiv:1702.02519, 2017.

13. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249, 2019.

14. Bobko, P. and Roth, P. L.: The four-fifths rule for assessing adverse impact: An arithmetic, intuitive, and logical analysis of the rule and implications for future research and practice. In Research in Personnel and Human Resources Management, pages 177–198. Emerald Group Publishing Limited, 2004.

15. Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R.: Signature verification using a siamese time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7(04):669–688, 1993.

16. Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R.: Optimized pre-processing for discrimination prevention. In Advances in Neural Information Processing Systems (NIPS), pages 3992–4001, 2017.

17. Ceci, M., Pio, G., Kuzmanovski, V., and Džeroski, S.: Semi-supervised multi-view learning for gene network reconstruction. PloS one, 10(12):e0144031, 2015.

18. Celis, L. E. and Keswani, V.: Improved adversarial learning for fair classification. arXiv preprint arXiv:1901.10443, 2019.

19. Chapelle, O., Schölkopf, B., and Zien, A.: Semi-Supervised Learning. MIT Press, 2006.

20. Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K.: Multi-view clustering via canonical correlation analysis. In International Conference on Machine Learning (ICML), pages 1–8, 2009.

21. Chopra, S., Hadsell, R., and LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 539–546. IEEE, 2005.

22. Chouldechova, A.: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. In Fairness, Accountability, and Transparency in Machine Learning (FATML), 2016.

23. Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. R.: Good semi-supervised learning that requires a bad gan. In Advances in Neural Information Processing Systems (NIPS), pages 6510–6520, 2017.

24. Dauphin, Y., de Vries, H., and Bengio, Y.: Equilibrated adaptive learning rates for non-convex optimization. In Advances in Neural Information Processing Systems (NIPS), pages 1504–1512, 2015.

25. De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., ODonoghue, B., Visentin, D., et al.: Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature medicine, 24(9):1342, 2018.

26. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6):141–142, 2012.

27. Dhillon, P., Foster, D. P., and Ungar, L. H.: Multi-View Learning of Word Embeddings via CCA. In Advances in Neural Information Processing Systems (NIPS), pages 199–207, 2011.

28. Dorfer, M., Kelz, R., and Widmer, G.: Deep linear discriminant analysis. In International Conference on Learning Representations (ICLR), 2016.

29. Dumoulin, V. and Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285, 2016.

30. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R.: Fairness through awareness. In 3rd Innovations in Theoretical Computer Science Conference, pages 214–226. ACM, 2012.

31. Faruqui, M. and Dyer, C.: Improving vector space word representations using multilingual correlation. In 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 462–471, 2014.

32. Fathony, R., Rezaei, A., Bashiri, M. A., Zhang, X., and Ziebart, B.: Distributionally robust graphical models. In Advances in Neural Information Processing Systems (NIPS), pages 8344–8355, 2018.

33. Foster, D. P., Kakade, S. M., and Zhang, T.: Multi-view dimensionality reduction via canonical correlation analysis. Technical report, Toyota Technological Institute, Chicago, Illinois, Tech. Rep. TTI-TR-2008-4, 2008.

34. Galbally, J., Diaz-Cabrera, M., Ferrer, M. A., Gomez-Barrero, M., Morales, A., and Fierrez, J.: On-line signature recognition through the combination of real dynamic data and synthetically generated static data. Pattern Recognition, 48(9):2921–2934, 2015.

35. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), pages 2672–2680, 2014.

36. Goodfellow, I. J., Shlens, J., and Szegedy, C.: Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR), 2015.

37. Gordaliza, P., Del Barrio, E., Fabrice, G., and Jean-Michel, L.: Obtaining fairness using optimal transport theory. In International Conference on Machine Learning (ICML), pages 2357–2365, 2019.

38. Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., Tatham, R. L., et al.: Multivariate data analysis, volume 6. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

39. Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods. Neural Computation, 16(12):2639–2664, 2004.

40. Hardt, M., Price, E., Srebro, N., et al.: Equality of opportunity in supervised learning. In Advances in neural information processing systems, pages 3315–3323, 2016.

41. Hoffer, E. and Ailon, N.: Semi-supervised deep learning by metric embedding. arXiv preprint arXiv:1611.01449, 2016.

42. Hu, J., Lu, J., and Tan, Y.-P.: Discriminative deep metric learning for face verification in the wild. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1875–1882, 2014.

43. Huang, G., Mattar, M., Lee, H., and Learned-Miller, E. G.: Learning to align from scratch. In Advances in Neural Information Processing Systems (NIPS), pages 764–772, 2012.

44. Hull, J. J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5):550–554, 1994.

45. Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (ICML), pages 448–456, 2015.

46. Iscen, A., Tolias, G., Avrithis, Y., and Chum, O.: Label propagation for deep semi-supervised learning. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5070–5079, 2019.

47. Izenman, A. J.: Linear discriminant analysis. In Modern multivariate statistical techniques, pages 237–280. Springer, 2013.

48. Jain, H., Prabhu, Y., and Varma, M.: Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2016.

49. Jung, C., Kearns, M., Neel, S., Roth, A., Stapleton, L., and Wu, Z. S.: Eliciting and enforcing subjective individual fairness. arXiv preprint arXiv:1905.10660, 2019.

50. Kaiye, W., Qiyue, Y., Wei, W., Shu, W., and Liang, W.: A comprehensive survey on cross-modal retrieval. CoRR, abs/1607.06215, 2016.

51. Kamiran, F., Karim, A., and Zhang, X.: Decision theory for discrimination-aware classification. In International Conference on Data Mining (ICDM), pages 924–929. IEEE, 2012.

52. Kim, M., Reingold, O., and Rothblum, G.: Fairness through computationally-bounded awareness. In Advances in Neural Information Processing Systems (NIPS), pages 4842–4852, 2018.

53. Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

54. Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M.: Semi-supervised learning with deep generative models. In Advances in Neural Information Processing Systems (NIPS), pages 3581–3589, 2014.

55. Kleinberg, J., Mullainathan, S., and Raghavan, M.: Inherent trade-offs in the fair determination of risk scores. In 8th Innovations in Theoretical Computer Science Conference (ITCS 2017), volume 67, page 43, 2017.

56. Koch, G.: Siamese neural networks for one-shot image recognition. Doctoral dissertation, University of Toronto, 2015.

57. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 202–207. AAAI Press, 1996.

58. Kohavi, R. and Becker, B.: Adult data set. UCI machine learning repository.

59. LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning. Nature, 521(7553):436–444, 2015.

60. Lee, D.-H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In ICML Workshop on Challenges in Representation Learning, volume 3, page 2, 2013.

61. Li, Y., Jin, L., Qin, A., Sun, C., Ong, Y. S., and Cui, T.: Semi-supervised auto-encoder based on manifold learning. In International Joint Conference on Neural Networks (IJCNN), pages 4032–4039, 2016.

62. Lopez-Paz, D., Sra, S., Smola, A., Ghahramani, Z., and Schölkopf, B.: Randomized nonlinear component analysis. In International Conference on Machine Learning (ICML), pages 1359–1367, 2014.

63. Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R.: The variational fair autoencoder. arXiv preprint arXiv:1511.00830, 2015.

64. Louppe, G., Kagan, M., and Cranmer, K.: Learning to pivot with adversarial networks. In Advances in Neural Information Processing Systems (NIPS), pages 981–990, 2017.

65. Lum, K. and Johndrow, J.: A statistical framework for fair predictive algorithms. arXiv preprint arXiv:1610.08077, 2016.

66. Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O.: Auxiliary deep generative models. In International Conference on Machine Learning (ICML), pages 1445–1453, 2016.

67. Maaten, L. v. d. and Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov):2579–2605, 2008.

68. Madras, D., Creager, E., Pitassi, T., and Zemel, R.: Learning adversarially fair and transferable representations. In International Conference on Machine Learning (ICML), pages 3381–3390, 2018.

69. Manisha, P. and Gujar, S.: A neural network framework for fair classifier. arXiv preprint arXiv:1811.00247, 2018.

70. Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning (ICANN), pages 52–59, 2011.

71. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K., Hu, Y.-H., Larsen, J., Wilson, E., and Douglas, S.: Fisher discriminant analysis with kernels. In Neural Networks for Signal Processing, pages 41–48, 1999.

72. Miyato, T., Maeda, S.-i., Ishii, S., and Koyama, M.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

73. Mueller, J. and Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

74. Nair, V. and Hinton, G. E.: Rectified linear units improve restricted boltzmann machines. In International Conference on Machine Learning (ICML), pages 807–814, 2010.

75. Neculoiu, P., Versteegh, M., and Rotaru, M.: Learning text similarity with siamese recurrent networks. In 1st Workshop on Representation Learning for NLP, pages 148–157, 2016.

76. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y.: Multimodal deep learning. In International Conference on Machine Learning (ICML), pages 689–696, 2011.

77. Noroozi, V., Bahaadini, S., Zheng, L., Xie, S., Shao, W., and Philip, S. Y.: Semi-supervised deep representation learning for multi-view problems. In IEEE International Conference on Big Data (Big Data), pages 56–64. IEEE, 2018.

78. Noroozi, V., Bahaadini, S., Zheng, L., Xie, S., and Yu, P. S.: Virtual adversarial training for semi-supervised verification tasks. In European Signal Processing Conference (EUSIPCO), pages 972–976. IEEE, 2018.

79. Noroozi, V., Zheng, L., Bahaadini, S., Xie, S., and Yu, P. S.: Seven: deep semi-supervised verification networks. In International Joint Conference on Artificial Intelligence (IJCAI), pages 2571–2577. AAAI Press, 2017.

80. Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S.: Deep metric learning via lifted structured feature embedding. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4004–4012, 2016.

81. Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In Advances in Neural Information Processing Systems (NIPS), pages 3235–3246, 2018.

82. Ororbia II, A., Giles, C. L., and Reitter, D.: Learning a deep hybrid model for semi-supervised text classification. In Conference on Empirical Methods in Natural Language Processing, 2015.

83. Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q.: On fairness and calibration. In Advances in Neural Information Processing Systems (NIPS), pages 5680–5689, 2017.

84. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S.: A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5):92, 2018.

85. Qi, Y., Song, Y.-Z., Zhang, H., and Liu, J.: Sketch-based image retrieval via siamese convolutional neural network. In IEEE International Conference on Image Processing (ICIP), pages 2460–2464. IEEE, 2016.

86. Qian, M. and Zhai, C.: Unsupervised feature selection for multi-view clustering on text-image web news data. In ACM International Conference on Information and Knowledge Management (ICKM), pages 1963–1966. ACM, 2014.

87. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T.: Semi-supervised learning with ladder networks. In Advances in Neural Information Processing Systems (NIPS), pages 3546–3554, 2015.

88. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.

89. Shah, K., Kopru, S., and Ruvini, J. D.: Neural network based extreme classification and similarity models for product matching. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Papers), volume 3, pages 8–15, 2018.

90. Sharma, A., Kumar, A., Daume, H., and Jacobs, D. W.: Generalized multiview analysis: A discriminative latent space. In IEEE Computer Vision and Pattern Recognition (CVPR), pages 2160–2167. IEEE, 2012.

91. Sindhwani, V., Niyogi, P., and Belkin, M.: Beyond the point cloud: from transductive to semi-supervised learning. In International Conference on Machine Learning (ICML), pages 824–831. ACM, 2005.

92. Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958, 2014.

93. Srivastava, N. and Salakhutdinov, R.: Learning representations for multimodal data with deep belief nets. In International Conference on Machine Learning Workshop (ICML), 2012.

94. Stuhlsatz, A., Lippel, J., and Zielke, T.: Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Transactions on Neural Networks and Learning Systems, 23(4):596–608, 2012.

95. Sugiyama, M.: Local fisher discriminant analysis for supervised dimensionality reduction. In International Conference on Machine Learning (ICML), pages 905–912, 2006.

96. Sun, Y., Ren, L., Wei, Z., Liu, B., Zhai, Y., and Liu, S.: A weakly-supervised method for makeup-invariant face verification. Pattern Recognition, 2017.

97. Sun, Y., Chen, Y., Wang, X., and Tang, X.: Deep learning face representation by joint identification-verification. In Advances in Neural Information Processing Systems (NIPS), pages 1988–1996, 2014.

98. Suykens, J. A. and Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters, 9(3):293–300, 1999.

99. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., and Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. arXiv preprint arXiv:1806.05236, 2018.

100. Verma, V., Lamb, A., Kannala, J., Bengio, Y., and Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825, 2019.

101. Wadsworth, C., Vera, F., and Piech, C.: Achieving fairness through adversarial learning: an application to recidivism prediction. arXiv preprint arXiv:1807.00199, 2018.

102. Wang, W., Arora, R., Livescu, K., and Bilmes, J.: On deep multi-view representation learning. In International Conference on Machine Learning (ICML), pages 1083–1092, 2015.

103. Wang, W. and Livescu, K.: Large-scale approximate kernel canonical correlation analysis. CoRR, abs/1511.04773, 2015.

104. Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

105. Wu, L., Shen, C., and van den Hengel, A.: Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. Pattern Recognition, 65:238–250, 2017.

106. Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V.: Unsupervised data augmentation. arXiv preprint arXiv:1904.12848, 2019.

107. Xie, S. and Philip, S. Y.: Active zero-shot learning: a novel approach to extreme multi-labeled classification. International Journal of Data Science and Analytics, pages 1–10, 2017.

108. Xu, C., Tao, D., and Xu, C.: Large-margin multi-viewinformation bottleneck. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

109. Xu, C., Tao, D., and Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634, 2013.

110. Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P.: Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In International Conference on World Wide Web (WWW), pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017.

111. Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P.: Fairness constraints: Mechanisms for fair classification. In Artificial Intelligence and Statistics, pages 962–970, 2017.

112. Zagoruyko, S. and Komodakis, N.: Learning to compare image patches via convolutional neural networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4353–4361, 2015.

113. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C.: Learning fair representations. In International Conference on Machine Learning (ICML), pages 325–333, 2013.

114. Zhang, B. H., Lemoine, B., and Mitchell, M.: Mitigating unwanted biases with adversarial learning. In 2018 AAAI/ACM Conference on AI, Ethics, and Society, pages 335–340. ACM, 2018.

115. Zhang, J., Tian, G., Mu, Y., and Fan, W.: Supervised deep learning with auxiliary networks. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2014.

116. Zheng, L., Noroozi, V., and Yu, P. S.: Joint deep modeling of users and items using reviews for recommendation. In 10th ACM International Conference on Web Search and Data Mining (WSDM), pages 425–434. ACM, 2017.

117. Zhu, J., Ahmed, A., and Xing, E. P.: Medlda: Maximum margin supervised topic models for regression and classification. In International Conference on Machine Learning (ICML), 2009.

VITA

NAME Vahid Noroozi

EDUCATION **Ph.D., Computer Science**, 2019

University of Illinois at Chicago, Chicago, Illinois.

M.Sc., Artificial Intelligence, 2011

Amirkabir University of Technology, Tehran, Iran.

B.Sc., Computer Engineering, 2008

Shiraz University, Shiraz, Iran.