Semi-supervised Learning using Triple-Siamese Network

by

DEBAPRIYA BANERJEE

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2020

ACKNOWLEDGEMENTS

I would like to thank my advising professor Dr. Won Hwa Kim for providing me an opportunity to work under his guidance. I also want to thank him for his kind support and constant encouragement throughout this thesis work. Without his constant support and guidance this research work would not have been successful. I would also like to thank Dr. Deokgun Park and Dr. Elizabeth Diaz for their support and for being part of my thesis committee.

June 3, 2020

ABSTRACT

Semi-supervised Learning using Triple-Siamese Network

DEBAPRIYA BANERJEE, M.S.

The University of Texas at Arlington, 2020

Supervising Professor: Dr. Won Hwa Kim

Missing data problem is inevitable in mostly all research areas including Artificial Intelligence, Machine Learning and Computer Vision where we have modicum knowledge about the complete dataset. One of the key reasons of missing data in AI is insufficiency of accurately labeled data. To solve a classification problem using ML or training a Deep Neural Network model, we need a huge amount of labeled data. It is difficult to get labeled data but unlabeled data is inexpensive and available easily. It is usual that we get no more than a single element per class to train our models due to unavailability of enough labeled training data. Strict privacy control or accidental loss may also cause missing data problem. One of the ways of getting training data labeled is using human-in-the-loop, but budget constraints can prevent that option. The objective of this research is to recover the complete signal or missing labels of the dataset using state-of-the-art Machine Learning and Computer Vision techniques. We propose a novel network trained with a few instances of a class to perform Metric Learning. We then convert our dataset to a graph signal and recover the graph completely using Recovery algorithm in Graph Fourier Transform. Our approach

performs significantly better than Graph Neural Network and other state-of-the-art techniques.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Supervised learning is one of the most popular learning approaches in Machine Learning for classification task where we have plenty of labeled data. But in reality, getting labeled data is difficult. There can be strict privacy control policy which restrict to get labeled data or human error can cause missing label information. Another way of labelling data is using human-in-the-loop, which means getting human support to annotate unstructured and unlabeled data. But it cannot be possible to have enough budget to get all the information labeled by human annotators. These all reasons cause missing label problem. Missing data can lead to misleading results. In this scenario, a very useful technique- Semi-supervised learning can be appropriate to solve missing data problem. Semi-supervised learning approaches need very limited amount of labeled data along with a large number of unlabeled data as input and it learns a better prediction rule than based on labeled data alone. Considering this approach in mind we design a semi-supervised algorithm which is very useful to detect the labels or categories of unlabeled data. The concept behind our approach is - similar data points lie very close to each other in vector space. Therefore, our objective is to learn a good similarity measure to detect the distance of data points in vector space and then detect the labels of unlabeled data through label propagation.

Metric Learning is a popular concept in modern Machine Learning areas to learn a good similarity measure between two entities. Distance metric learning (or simply, metric learning) aims at automatically constructing task-specific distance metrics from supervised data, in a machine learning manner. It is often difficult to design

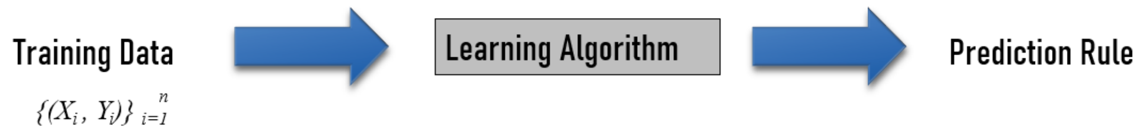

Figure 1.1: Supervised Learning

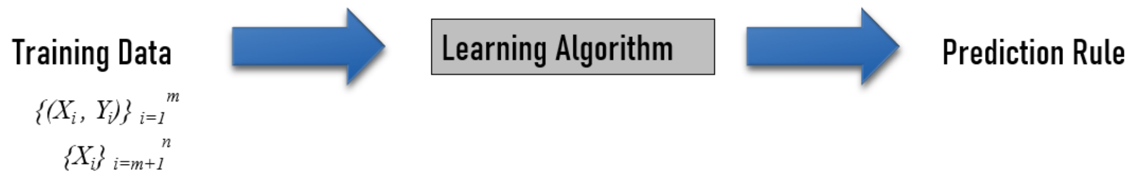

Figure 1.2: Semi-supervised Learning

metrics that are well-suited to the particular data and task of interest. In this project, we design a novel metric learning model to detect the distance between every two images in our dataset and then we propagate the label information from labeled data to unlabeled data. It is assumed that the similar object will have distance very close to zero and they will share the same label information.

Label propagation is an interesting problem in semi-supervised learning where a small subset of data points has label information and the challenge is to propagate labels to the unlabeled points from labeled points through an algorithm. As our data is sparse, we use this sparsity property and apply harmonic analysis of graph and spectral graph theory to predict the labels of unlabeled points.

1.1 Thesis Organization

Rest of the thesis is organized as follows -

- Chapter 2 discusses the work done related to this field of thesis
- Chapter 3 elaborates the preliminary work done in this field
- Chapter 4 describes the detailed design of our approach

- Chapter 5 elaborates the detailed implementation of Triple Siamese network architecture
- Chapter 6 contains all the experiments that are performed as part of this thesis and results obtained for each
- Chapter 7 includes Conclusion and future work for this thesis

CHAPTER 2

RELATED WORK

One-shot learning is one of the way of learning good similarity measure and One-shot learning was first introduced by Fei-Fei et al. (2006) [3]. they assumed that currently learned classes can help to make predictions on new ones when just one or few labels are available. Lake et al. (2015) [4] presented a Hierarchical Bayesian model that reached human level error on few-shot learning alphabet recognition tasks.

Koch et al. (2015) [2] presented a deep learning model based on computing the pair-wise distance between samples using Siamese Networks, Vinyals et al. (2016) [5] presented an end-to-end trainable k-nearest neighbors using the cosine distance.

Another related area of research concerns on graph-structured data. The Graph Neural Network is a deep learning approach to deal with graph-structured data. It was first proposed in Gori et al. (2005) [6]. Kim et al. [7] proposed an SR model to recover multivariate signal in an online manner using Graph Signal Processing. Kipf Welling (2016) [8] proposed deep learning based GNN models on semi-supervised classification problems.

Chang et al. (2016) [9] develop graph interaction networks that learn pairwise particle interactions and apply them to discrete particle physical dynamics. Duvenaud et al. (2015) [10]; Kearnes et al. (2016) [11] study molecular fingerprints using variants of the GNN architecture, and Gilmer et al. (2017) [12] further develop the model by combining it with set representations Vinyals et al. (2015) [5], showing state-of-the-art results on molecular prediction. Appalaraju et al. 2017 [13] proposed Siamese

Network based Deep CNN model using Curriculum learning, Transfer learning for image similarity measure.

Label propagation is another way in semi-supervised learning to detect the labels of unlabeled observation. Puy et al. [14] first proposed Random sampling of bandlimited signals on graphs. Later, Kim et al.2016 [1] proposed Adaptive Signal Recovery on Graphs via Harmonic Analysis for Experimental Design in Neuroimaging. Bronstein et al. (2017) [15] has an exhaustive literature review paper Geometric deep learning: going beyond Euclidean data on this topic.

Another study on nearest neighbor search is very popular in the context of label propagation. Malkov et al. [16] first introduced nearest neighbor search using small world problem. Saito et al. [17] proposed a semi-supervised domain adaption technique via Minimax Entropy. Later, Zhai et al. [18] proposed Self-Supervised Semi-Supervised Learning where they bridged the gap between self-supervised and semi-supervised learning.

CHAPTER 3

PRELIMINARY WORK

We restrict our focus to fashion dataset which has been collected from an online fashion e-commerce store called Myntra. Although we concentrate our focus on fashion dataset, this approach can be useful for any modality. This fashion data consists of a set of images of the fashion products along with their label information. Each of these images has been processed to grayscale and resized to 105×105. It is then processed to extract the features.

One-shot learning approach is a very popular technique to learn domain-specific features from input images whether the number of input images per class is limited. This technique is very useful even for metric learning whether the similarity measure to be calculated between input images. In this project, we use this approach as our baseline model to calculate a good similarity measure between images. Koch et al. [2] focuses on Twin Siamese network to perform metric learning and then reuse that network's features for one-shot learning without any retraining.

Twin Siamese network performs well for one-shot learning task but it takes a large amount of time to converge while training. This model takes two input at a time and calculate their distances to get a similarity score. Therefore, the number of comparison is more ($\binom{N}{k}$ for N number of images) to calculate the distance of all the images in our dataset. This Twin Siamese Network is sensitive with respect to context and this model fails to capture the fine-grain differences between two images.

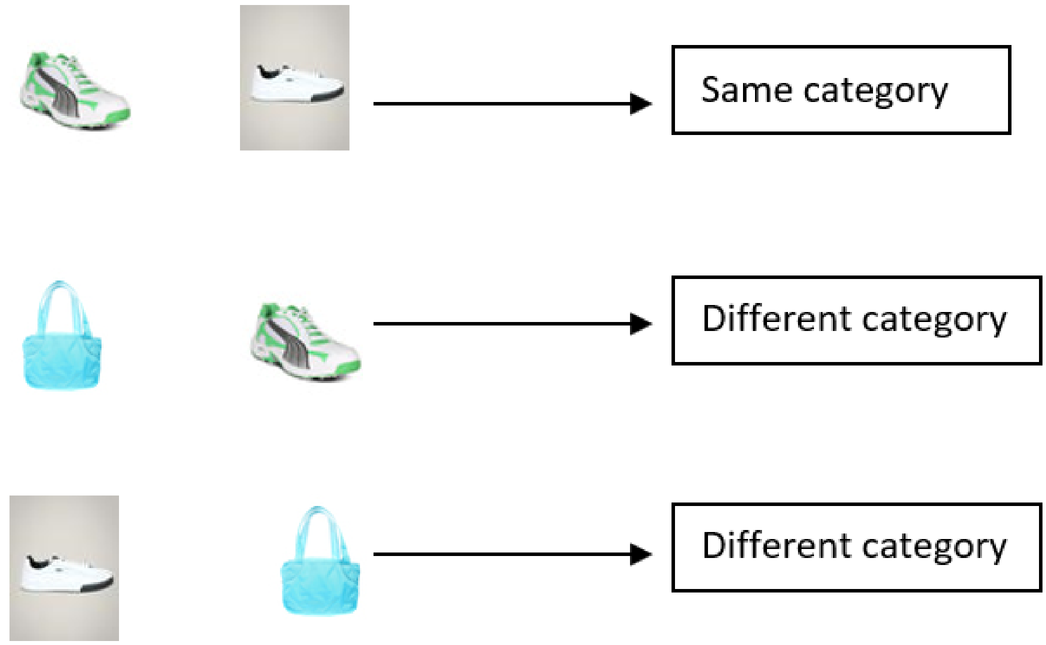

Figure 3.1: One-shot verification task (training)

The above-mentioned problems motivate us to dive deep intro metric learning. We focus on creating a novel network which converges fast and also captures the fine-grain differences of labels.

CHAPTER 4

DESIGN

In this section, we discuss the design approach we propose. Semi-supervised learning is an approach where learning algorithm requires a small number of training data and a large number of unlabeled data. Our goal in this project is to learn a prediction rule to detect the labels of unlabeled data set with the help of small amount of labeled data points. We divide our algorithm in two steps: 1. Metric Learning 2. Label Propagation.

4.1 Metric Learning

Distance metric learning (or simply, metric learning) aims at automatically constructing domain-specific distance metrics from supervised data, in a machine learning manner. In this step, we learn a good similarity measure between two images. In most of the problems, we simply use a standard distance metric e.g. Euclidean distance, cosine similarity etc. where we have prior knowledge of the domain. However, it is difficult problem to solve for a particular data or task. The aim of metric learning is to construct task specific distance metric from a particular data set.

Twin Siamese network is one of the ways of learning distance between two images. The concept of Twin Siamese network is that there will be two parallel networks which share the shame weights. Basically, both of these parallel networks are identical to each other and it learns encoding of input images. We use this network as our baseline model.

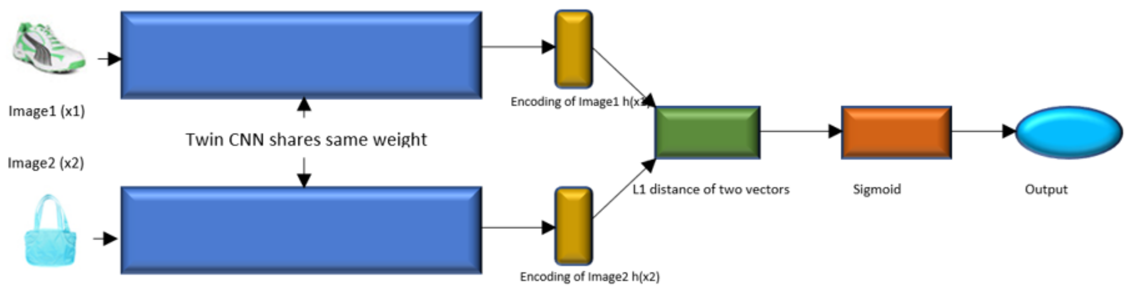

Figure 4.1: Twin Siamese Network

Our baseline model is a CNN version of Twin Siamese network which takes two input images and learns the encoding of two input images through this twin networks. At the end, it learns the L_1 distance of two feature vectors and produce a distance score between 0 and 1 through a sigmoid layer.

The details architecture of this network is: The model consists of a sequence of convolutional layers, each of which uses a single channel with filters of varying size. The network applies a ReLU activation function to the output feature maps, followed by max-pooling with a filter size and stride of 2.

Twin Siamese network perform well compared to all other metric learning mentioned in related work. But this network performs very slow. The convergence time of this network is extremely large (1.6 hrs) and it takes almost 2000 epochs to converge. If we have N number of images in the dataset, the number of comparison happens in this network is $\binom{N}{k}$ which is very large to get the distance of every pair of images in our dataset. In this context, we come up with the concept of Triple Siamese network to optimize this above mentioned points.

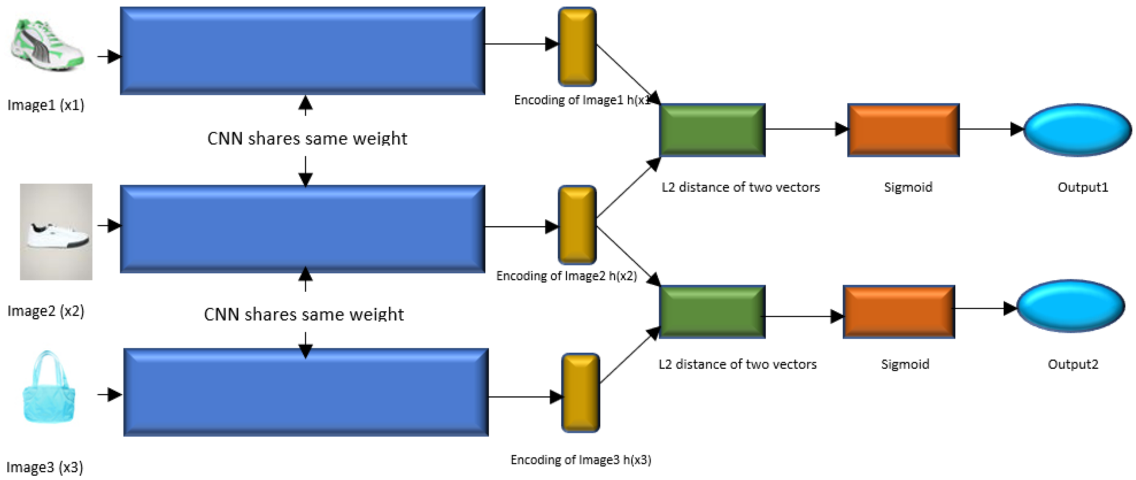

Figure 4.2: Triple Siamese Network

4.1.1 Triple Siamese network

In this network we use three parallel networks which share the same weights among them. The architecture of each network is the same CNN network that we use in Twin Siamese network. Through this network, we aim to learn the similarity function as $f(x_1, x_2), where f(x_1, x_2) = ||h(x_1) - h(x_2)||_2$.

This network takes input as three images(x_1, x_2, x_3) in such a way that x_2 and x_1 taken from same class and x_2 and x_3 taken from different class. The goal is to maximize the distance between x_2 and x_3 and minimize the distance between x_2 and x_1. This network learns two L_2 distances from this two pairs at a time.

4.1.2 Graphical Representation

Once we have information regarding the distance of each pair of images in our dataset, we represent our data in a graphical way. Therefore, we have the Graph $G = \{V, E\}$, where V is a set of images and E is a set of distances of each pair of

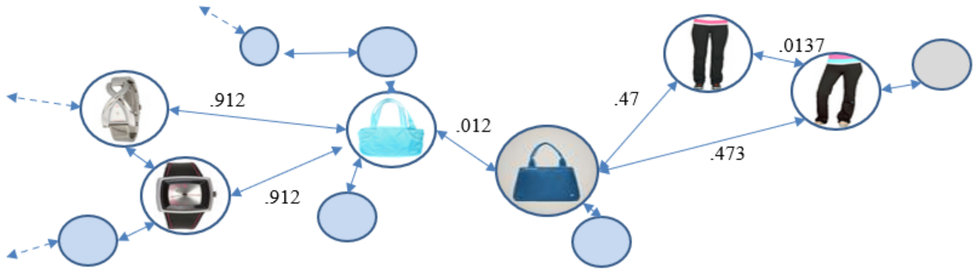

Figure 4.3: Example of Graphical Representation of image data set

images. Hence, $V = \{v_1, v_2 \ldots v_N\}$, $E = \{e_{ij}$ where, $e_{ij} = \text{dist}(v_i, v_j)\}$, and $N =$ total number of images.

4.2 Label Propagation

The concept of Label Propagation is to determine the labels of unlabeled data from a very small amount of labeled data. In this project we consider the graph as graph signal and perform recovery algorithm(Kim et al. [7]) on the graph. We recover the graph signal using harmonic analysis of the graph. We apply two useful concept: 1. Spectral graph theory [19] and 2. Harmonic Analysis of graph [1,7] to perform our recover algorithm.

4.2.1 Spectral Graph Theory

Spectral Graph Theory [19] is the study of the properties of a graph in relationship to the eigenvalues, and eigenvectors of matrices associated with the graph, such as its distance matrix or Laplacian matrix. We have the graph $G = \{V, E\}$ is represented by a set of vertices V of size N and a set of edges E that connects the vertices. Another matrix, Graph Laplacian $L = D – W$ where a degree matrix $D_{N \times N}$, is a diagonal matrix with the i^{th} diagonal element is the sum of edge weights connected to the i^{th} vertex and $W_{N \times N}$ is a distance matrix where it is the most common way to represent a graph G where each element a_{ij} denotes the distance between the i^{th} and the j^{th} vertices.

4.2.2 Harmonic Analysis of Graph

Harmonic Analysis of Graph [1, 7] refers to utilize Fourier transform/Wavelet transform of the original signal and perform filtering in frequency domain. The reason behind using harmonic analysis is we want to make use of sparsity in terms of representations obtained in the Fourier/wavelet space of the graphs. By constructing different shapes of band-pass filters in the frequency space and transforming them back to the original space, we can construct a mother wavelet ψ on the nodes of a graph comes from its representation in the frequency space. For this implementation, using spectral graph theory we use orthonormal bases to design a kernel function $g()$ in the space spanned by the bases. Graph Laplacian L provides eigenvalues and corresponding eigenvector for N number of vertices. This eigenvectors is used to define graph Fourier transform of a function $f(n)$ defined on a vertices n. If the signal $f(n)$ lives in the spectrum of the first k eigenvectors, we say that $f(n)$ is k-bandlimited. Graph Fourier Transform of a function $f(n)$ defined on the vertices n as

$$\hat{f}(l) = \sum_{n=1}^{N} f(n)\chi_l^*(n) \tag{4.1}$$

and

$$f(n) = \sum_{l=1}^{N} \hat{f}(l)\chi_l(n) \tag{4.2}$$

where, χ is a set of eigenvectors which provide orthonormal bases and χ^* is conjugate of χ. the graph Fourier coefficient $\hat{f}(l)$ is obtained by the forward transform and the original function $f(n)$ can be reconstructed by the inverse transform.

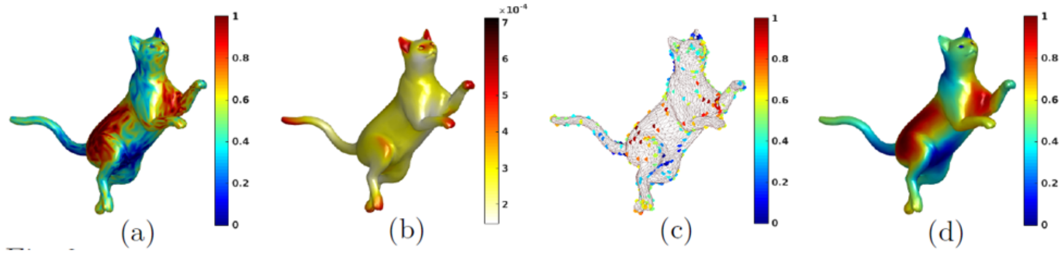

Figure 4.4: A toy example of our framework on a cat mesh (N = 3400). a) Band-limited random signal in [0, 1] with noise, b) Sampling probability p1, c) Sampled signal at m = 340 locations out of 3400, d) Recovered signal using our method with only k = 50. [1]

4.2.3 Problem Set-up

Now we have the problem set-up as Graph signal $f \in \mathbb{R}^{N \times p}$ where, measurement p at each node is one-hot encoding of node labels and N is number of images/number of nodes. We have available partial observation at m vertices, where $m << N$. Therefore, partial signal $y \in \mathbb{R}^{m \times p}$ and the goal is to complete the full graph and recovered the signal f from partially observed y.

4.2.4 Recover Algorithm

As part of Recovery algorithm [1, 7, 14], we solve the optimization problem:

$$g^* = \arg\min_{g \in \mathbb{R}^n} ||\mathcal{P}_\Omega^{-1/2}(Mg - y)||_2^2 + \gamma g^T h(\mathcal{L}) g \tag{4.3}$$

where $\mathcal{P}_\Omega = diag(p(\Omega)), h(\mathcal{L}) = \sum h(\lambda_l)\chi_l\chi_l^T$, $Mf = y$, $p(\Omega)$ is random uniform distribution probability. Sample m nodes from N (random uniform sampling strategy) and the locations where we observe the signal be denoted as $\omega = \{\omega_1, \omega_2 \ldots \omega_m\}$. Based on the m observations, we can build a projection matrix [7] $M_{m \times N}$ as

$$M_{ij} = \begin{cases} 1, & \text{if } j = \omega_i \\ 0, & \text{o.w.} \end{cases}$$

This framework solves for a solution to Eq. 4.3 entirely in a dual space by projecting the problem to a low dimensional space where we search for a solution of size $k << N$ [1]. We take graph Fourier transform of the function as g and $\hat{g}_k$ are the first k coefficients. Reformulating Eq. 4.3 using $g = V_k \hat{g}_k$ [1] where, $V_k = [\chi_1, \chi_2, ..., \chi_k]$ as

$$\hat{g}_k^* = \arg \min_{\hat{g}_k \in \mathbb{R}^k} ||\mathcal{P}_\Omega^{-1/2}(MV_k\hat{g}_k - y)||_2^2 + \gamma(V_k\hat{g}_k)^T h(\mathcal{L})(V_k\hat{g}_k) \tag{4.4}$$

From Eq. 4.3 we recover a low-rank estimation $g^* = V_k\hat{g}_k^*$ that reconstruct our original signal f. Fig 4.4 shows a toy example of this recovery algorithm. Here, given a cat mesh with N = 3400 vertices, we define a random signal $f \in [0, 1]$ that is band-limited in the spectrum of $\mathcal{L}$ with Gaussian noise of $N(0, 0.1)$, p_1 is the sampling distribution and sample size $m = 340$ vertices which is 10% of the total vertices without replacement. Our estimation g using $k = 50$ bases is shown in Fig. 4.4 d). The error between the true f and g is extremely small despite using such little data to begin with [1].

CHAPTER 5

IMPLEMENTATION

CNN based Siamese networks were first introduced by Koch et al. [2] for one-shot image recognition. Our Triple Siamese network architecture is inspired by Twin Siamese network but instead of two we have three parallel networks. This three parallel networks individually are CNN networks. Each CNN network consists of sequence of convolutional layers each of which uses a single channel with filters of varying size and a fixed stride of 1. The network applies a ReLU activation function to the output feature maps, by max-pooling with a filter size and stride of 2. Each layer takes the following form:

$$h_{1,n} = max(0, W_{l-1,l}^{T} * h_{1,l-1} + b_l) \tag{5.1}$$

$$h_{2,n} = max(0, W_{l-1,l}^{T} * h_{2,l-1} + b_l) \tag{5.2}$$

$$h_{3,n} = max(0, W_{l-1,l}^{T} * h_{3,l-1} + b_l) \tag{5.3}$$

Where W is the shared weight matrix and b_l is shared bias vector and $*$ is the valid convolutional operation corresponding to each convolutional filter and the input feature maps.

This layer followed by a fully-connected layer and at the top we have another layer to compute the L_2 distance between the feature vectors generated by three networks and each L_2 distance is passed through a softmax output to get a distance between two pairs of inputs.

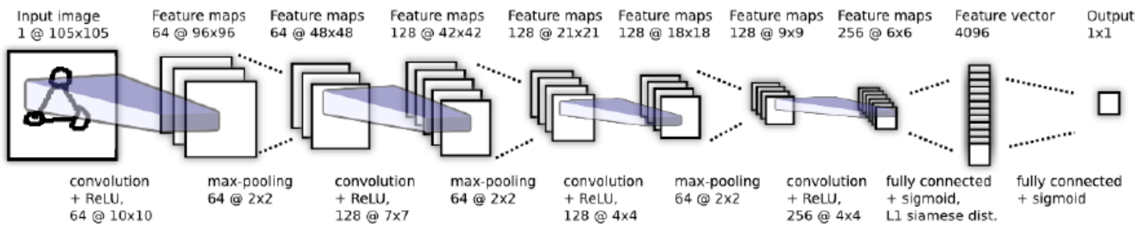

Figure 5.1: Detail architecture of a single convolutional Neural Network used in Siamese Network [2]

Figure 5.2: Average training accuracy calculated by performing 2-way one-shot learning

Fig 5.1 shows an example of a CNN architecture used in Triple Siamese network. In Triple Siamese network the 4095 unit fully-connected layer is followed by another layer where we calculate the L_2 distance of each pairs.

5.1 Learning

We initialize all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 10^{-2}. Biases were initialized from a normal distribution with mean 0.5 and standard deviation 10^{-2}. In the fully-connected layers, the biases were initialized in the same way as the convolutional layers, but the weights were drawn from a much wider normal distribution with zero-mean and standard deviation 2×10^{-2} according to Koch et al. [2]. In the network, training is preformed by feeding the network with three samples (x_1, x_2, x_3) at a time where, x_2 and x_1 are of the same class, and x_3 is of different class. The goal is to maximize the distance between x_2 and x_3 and minimize the distance between x_2 and x_1. At the top, there is a comparison layer to compare both the similarities, followed by sigmoid layer to normalize the output in the range of 0 and 1. We use MSE as the loss function here. The back-propagation algorithms runs to update the weights for all three network simultaneouly. Triple Siamese networks converges after 150 epochs while training with fashion dataset.

5.2 Validation

For every pair of input images, this model generates a similarity score between 0 and 1. N-way one-shot methods applies to measure the performance of this network. We consider N as 2 to perform one-shot learning and repeat this k times to calculate percentage of prediction p as:

$$p = \frac{(100 \times n)}{k} \tag{5.4}$$

For validation, we use 50 trials. Therefore, k is 50.

5.3 Twin Siamese vs Triple Siamese network

Triple Siamese network converges very fast compared to Twin Siamese network. To detect the distance between each pairs of images in our dataset, the number of comparisons required less in Triple Siamese network, and we get two distance scores at a time. Triple Siamese Network has two parallel layers to calculate distance metrics, which gives two distance measures at a time. This approach helps to reduce the training time in Triple Siamese network. The objective of using three layers of CNN network is to feed the network with three input images in such a way that two images are from same class and the third one is from different class and then we make two pairs(two from same class and other two from different class) out of these three input images to calculate two pairs of distance. This model helps to minimize the distance between the two images in the first pair and maximize the distance between the images in the second pair. Through this model we achieve to capture fine-grained differences between classes which reside very closely in vector space.

CHAPTER 6

EXPERIMENTS AND RESULTS

In this section, we demonstrate the result of our algorithm using Fashion dataset. We have collected the image dataset from Myntra Fashion website which has 4000 images with 31 different labels or category e.g. handbags, shoes, socks, watches etc. For our experiment, we first create a graph using our Triple Siamese network with the images, where images are defined as vertices/nodes and edges denote the similarity score between two images. Measurements at each vertex were defined by one-hot encoding of label vector representing object labels where non-zero elements indicate whether the corresponding objects exist in the image. As our dataset consists of 31 labels, each of the measurement will be a 1×31 vectors. Therefore, we get a $f_{4000X31}$ matrix for our entire image dataset which served as the ground truth of our algorithm. We assumed our sample m to be chosen using uniform random sampling and $m << N$. Therefore, m is our partialy observed dataset. The k value of our experiment used 500 which gives optimal accuracy.

We perform our semi-supervised learning algorithm on this dataset first to create a graph using our proposed Triple Siamese Network, then we applied our label propagation algorithm to recover the signal at each vertex. We compare our algorithm with the baseline model Twin Siamese network and k-NN along with state-of-the-art Graph Convolutional Network. Fig 6.1, 6.2 and 6.3 demonstrate the step-by-step operations of our algorithm. Fig 6.1 indicates the initial state of our dataset with graphical representation. Fig 6.2 indicates the state after metric learning and fig 6.3 indicates the final state after label propagation.

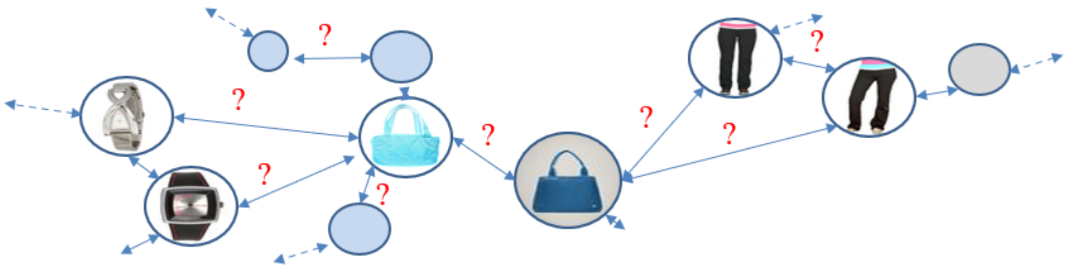

Figure 6.1: Initial state

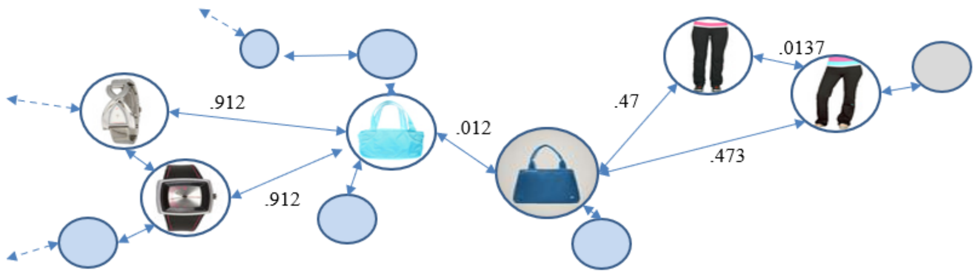

Figure 6.2: Metric Learning

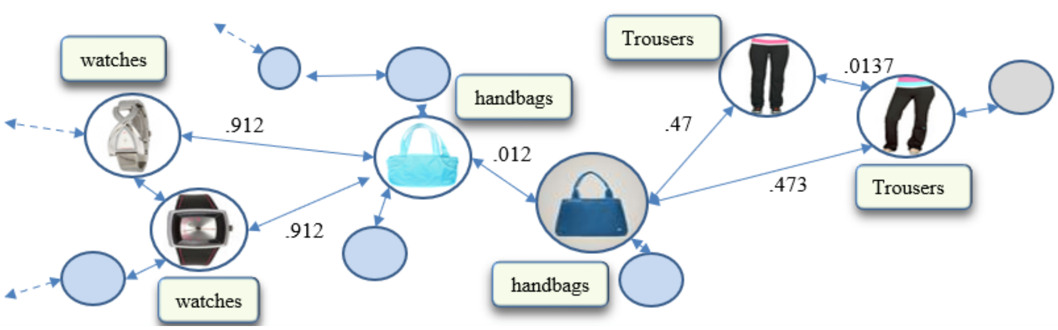

Figure 6.3: Label Propagation

6.0.1 **Results**

After recovering the labels of all the images in our dataset, we channelize our output to a sigmoid layer to convert the recovered labels as binary. Since we have chosen our sample at random uniform sampling, we performed each experiment 50 times to get average accuracy and average precision. Our algorithm outperform all

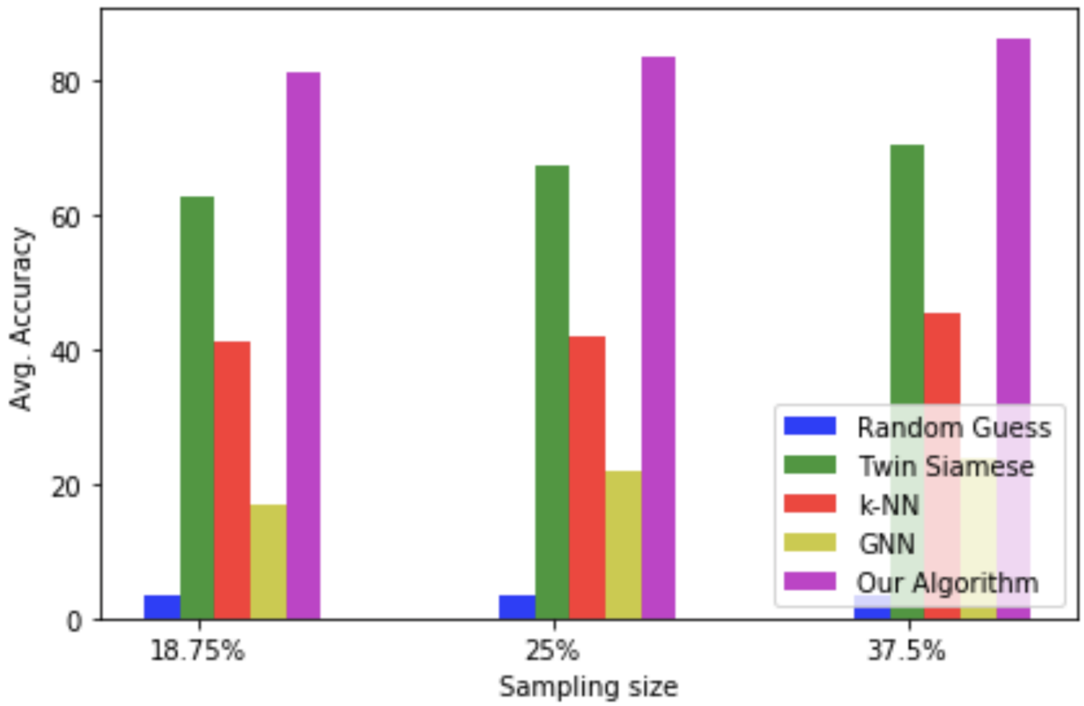

Figure 6.4: Avg. Accuracy comparison

other baselines and state-of-the-art algorithms. We have shown the performance of our Triple Siamese Network with Recovery algorithm in Fig 6.4 and Fig 6.5.

We observe the performance of our algorithm with different sample size. We sample 18.7%, 25% and 37.5% data points as our partial observation and using this we recover the signal. We observe that the performance of our algorithm increases with increasing the sample size.

Models	**P.O. 750**	**P.O. 1000**	**P.O. 1500**
Triple Siamese network and Recovery Algorithm	81.1%	83.7%	86.4%
Twin Siamese network and Recovery Algorithm	62.8%	70.2%	79.3%
Triple Siamese network and k-NN	41.2%	50%	45.5%
Graph Neural Network	17%	21.8%	23.7%
Random Guess	3.2%	3.2%	3.2%

Table 6.1: **Accuracy Comparison of our model with other models for different Partial Observation size**

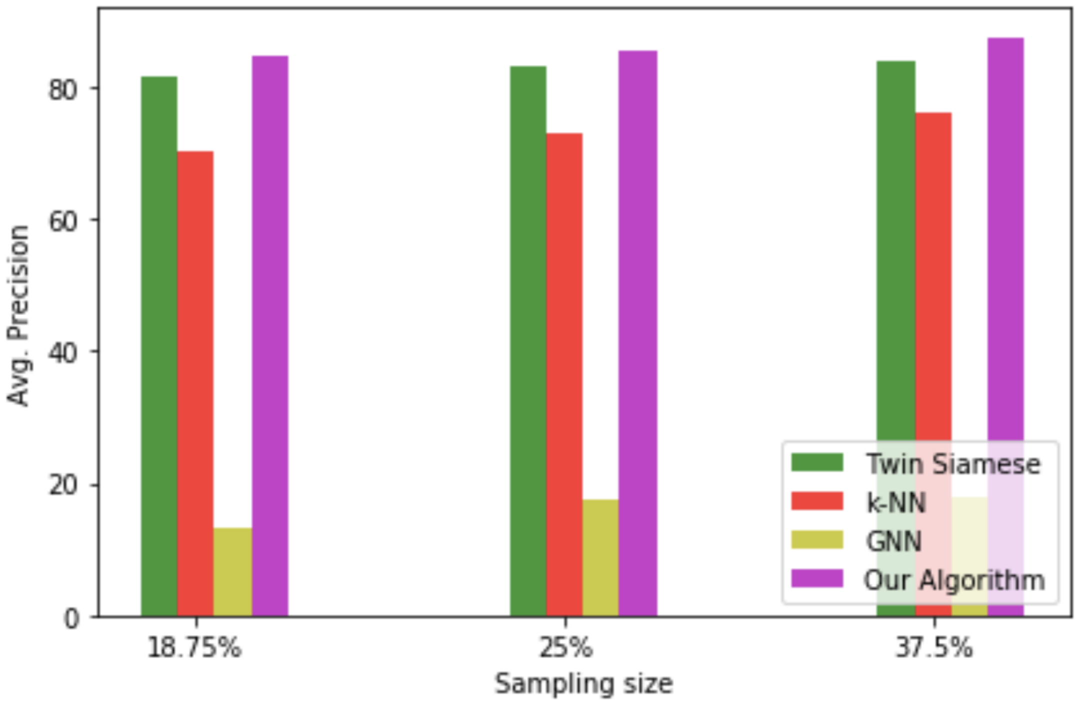

Figure 6.5: Avg. Precision comparison

We compare the performance of our model with different partial observation size and we observe that even if with 750 partial observation size out of 4000 data points, our model performs significantly better compared to other models.

Models	**P.O. 750**	**P.O. 1000**	**P.O. 1500**
Triple Siamese network and Recovery Algorithm	84.6%	85.5%	87.6%
Twin Siamese network and Recovery Algorithm	71.4%	82.7%	82.8%
Triple Siamese network and k-NN	60.4%	65.8%	67.3%
Graph Neural Network	13.3%	17.4%	18%

Table 6.2: **Precision Comparison of our model with other models for different Partial Observation size**

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis we introduced a novel approach for semi-supervised learning where we have limited number of labeled data to train our model. As the backbone of our algorithm we assume that similar data points reside close to each other in vector space and different data points resides far away from each other in vector space.We then discussed different metric learning approaches to retrieve similarity measure from image feature vectors. We also discussed about the novel algorithm for metric learning which we designed in this project. Our metric learning model converges fast and able to capture the fine-grain differences between class labels. We then discussed about the concept of Spectral Graph Theory and the application of harmonic analysis on the graph signal to recover the signal from partially observed data. Our algorithm performs significantly better(20% better compared to baseline model) than others even for very small number of labeled data.

This work can be extended towards different modality, where we can merge Natural Language Processing and Image Processing to develop more powerful metric learning model. We can also extend our work towards designing a Deep Learning based Label Propagation algorithm.

REFERENCES

[1] W. H. Kim, S. J. Hwang, N. Adluru, S. C. Johnson, and V. Singh, "Adaptive signal recovery on graphs via harmonic analysis for experimental design in neuroimaging," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9910. Springer, 2016, pp. 188–205. [Online]. Available: https://doi.org/10.1007/978-3-319-46466-4_12

[2] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," 2015.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[4] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015. [Online]. Available: https://science.sciencemag.org/content/350/6266/1332

[5] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3630–3638. [Online]. Available: http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf

[6] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 729–734 vol. 2, 2005.

[7] W. Hwa Kim, M. Jalal, S. Hwang, S. C. Johnson, and V. Singh, "Online graph completion: Multivariate signal recovery in computer vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[9] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," *CoRR*, vol. abs/1612.00341, 2016. [Online]. Available: http://arxiv.org/abs/1612.00341

[10] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *CoRR*, vol. abs/1509.09292, 2015. [Online]. Available: http://arxiv.org/abs/1509.09292

[11] S. M. Kearnes, K. McCloskey, M. Berndl, V. S. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *Journal of Computer-Aided Molecular Design*, vol. 30, pp. 595–608, 2016.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.

[13] S. Appalaraju and V. Chaoji, "Image similarity using deep CNN and curriculum learning," *CoRR*, vol. abs/1709.08761, 2017. [Online]. Available: http://arxiv.org/abs/1709.08761

[14] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *CoRR*, vol. abs/1511.05118, 2015. [Online]. Available: http://arxiv.org/abs/1511.05118

[15] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, pp. 18–42, 2017.

[16] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *CoRR*, vol. abs/1603.09320, 2016. [Online]. Available: http://arxiv.org/abs/1603.09320

[17] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," *CoRR*, vol. abs/1904.06487, 2019. [Online]. Available: http://arxiv.org/abs/1904.06487

[18] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," 2019. [Online]. Available: https://arxiv.org/pdf/1905.03670.pdf

[19] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *ArXiv*, vol. abs/0912.3848, 2009.

BIOGRAPHICAL STATEMENT

Debapriya Banerjee was born in Kolkata, India. She received her Bachelor of Technology degree in Computer Science and Engineering from West Bengal University of Technology, Kolkata, India in 2012. After that she worked as a NLP Researcher at Accenture Research Labs, Bangalore, India, and Data Engineer at Walmart Labs, Bentonville, AR, USA. In the Fall of 2018, she started her graduate studies in Computer Science at The University of Texas, Arlington. She worked as an Data Science intern at Target.com, Sunnyvale, CA in summer 2019. She received her Master of Science in Computer Science from The University of Texas at Arlington, in June 2020. Her research interests include Computer Vision, Natural Language Processing, Multimodal Analysis, Machine Learning.