Emperical Evaluation of Graph Based Semi Supervised Learning Techniques

Deepak Rishi

April 18, 2016

1 Introduction

1.1 What is the problem?

Supervised Learning is the most widely used concept for training machine learning models. However, learning a reliable model usually requires plenty of labeled data.

This problem is particularly prevalent when doing text classification. annotating text data requires more time than labelling images. While labeled data is difficult to obtain, unlabeled data is available in large quantity and easy to collect.

1.2 Why is it an important problem?

Procuring labelled data can be sometimes very expensive to obtain. It is therefore very important to infer some relevant information from unlabelled data.

Semi-supervised learning is attractive because it can potentially utilize both labeled and unlabeled data to achieve better performance than supervised learning. This reduces the annotation effort, which leads to reduced cost.

This project aims to explore the area of Graph based Semi-Supervised Learning, which is based on the assumption that the data set has some underlying structure which can be exploited for classifying instances. Figure 1 shows an example of such a case.

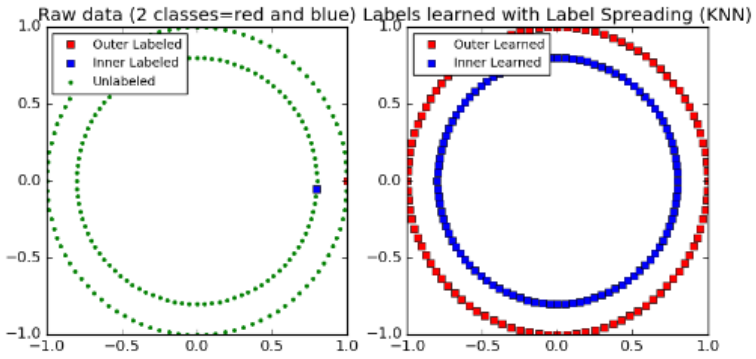


Figure 1: Example of Semi Supervised Learning [Scikit-Learn]

2 Techniques to tackle the problem

Section 1 introduced the concept of semi supervised learning. In this section I explore Graph based Semi-Supervised Learning (SSL).

The reasons for choosing Graph Based SSL are (Talukdar and Subramanya, 2012).

• Many datasets are naturally represented by a graph. Examples include but not limited to web, citation network, social network..

• Uniform representation for heterogeneous data.

• Easily parallelizable, scalable to large data.

Graph-based SSL methods operate on a graph where a node represents a data instance and a pair of nodes are connected by a weighted edge. Some nodes in the graph are labeled, and some not.

Graph construction methods can be classified into the following two categories.

• Task-independent Graph Construction : These methods are unsupervised in nature and do not take into account the label of the data while constructing the graph. Examples of this technique include $k - Nearest\ Neighbours$ and $\epsilon - Neighbourhood$.

• Task-dependent Graph Construction : These methods take into account the labels of the data and also make use of unlabeled data to construct the graph. Examples include Inference-driven Metric Learning (IDML) (Dhillon et al., 2010) which is a semi-supervised metric learning algorithm which exploits labeled as well as widely available unlabeled data to learn a metric. The learned metric can in turn be used to construct a task-specific graph.

In this Project I focus on Task-independent Graph Construction using kernel and *Distance Metric* learning approaches.

2.1 Metric Learning

Many algorithms such as $k - Nearest\ Neighbours$ rely on the distance metric for the input data patterns to compute the degree of similarity between different objects. Metric Learning is the task of learning a distance function over a space of objects.

A distance metric $D(x, y)$ should satisfy the following 4 properties

• Nonnegativity $D(x, y) \geq 0$

• Identity of indiscernibles, $D(x, y) = 0$, iff x=y, else it is called a pseudo distance metric

• Symmetry , $D(x, y) = D(y, x)$

• Subadditivity $D(x, y) + D(y, z) \geq D(x, z)$

2.1.1 Linear Distance Metric

Linear Distance Metric (also known as Generalized Mahalanobis distance) first project's the data into some space and then evaluates the pairwise data distance as their Euclidean distance in the projected space. This is explained as follows

Generalized Mahalanobis distance : is defined as

$$D(x, y) = \sqrt{(x - y)^T M (x - y)} \tag{1}$$

For D to be a distance metric M has to be *symmetric* and *positive definite*. Thus, M can be decomposed by cholesky decomposition into a upper or lower triangular matrix (L) and it transpose.

$$D(x, y) = \sqrt{(x - y)^T L^T L (x - y)} \tag{2}$$

$$D(x, y) = \sqrt{(Lx - Ly)^T (Lx - Ly)} \tag{3}$$

When M is the covariance matrix then D is called the Mahalanobis distance.

2.1.2 Large Margin Nearest Neighbour (LMNN)

$LMNN$ (Weinberger and Saul, 2009) is a supervised distance metric learning approach which learns a pseudometric using semidefinite programming .

$LMNN$ learns a pseudometric which transforms the data such that all points in the training dataset are surrounded by their k nearest neighbours.

The optimization problem can be formulated as

$$\begin{aligned} min_M \sum_{i,j \in N_{i,j}} & D(x_i, x_j) + \sum_{i,j,l} \epsilon_{i,j,l} \\ & \forall i, j \in N_{i,j}, y_l \neq y_j \end{aligned} \tag{4}$$

subject to the constraints

$$\begin{aligned} M &\geq 0 \\ \epsilon_{i,j,l} &\geq 0 \end{aligned} \tag{5}$$

In equation 4 i, j are the target neighbours belonging to the same class/neighbourhood N. y_l is the imposter class example.

2.1.3 Information-Theoretic Metric Learning (ITML)

Information-Theoretic Metric Learning (ITML) (Davis et al., 2007) is a supervised Metric Learning approach which minimizes the differential relative entropy between two multivariate Gaussians under constraints on the distance function, which can be formulated into a Bregman optimization problem by minimizing the LogDet divergence subject to linear constraints.

The LogDet divergence is a loss function that describes the distance between Positive Definite Matrices. It is defined as

$$D_{ld}(X, Y) = trace(XY)^{-1} - logdetXY^{-1} - d \tag{6}$$

where d is the dimension (number of rows or columns, since its symmetric).

The goal of ITML is to learn a similarity metric d_A which is close to some starting metric d_{A_0}. Instead of comparing the distance metrics it compares the relative entropy of the gaussians formulated by them and minimizes the KL *divergence* between them.

$$\begin{aligned} d_A(x,y) &\rightarrow N(x|\mu, A) \\ d_{A_0}(x,y) &\rightarrow N(x|\mu, A_0) \end{aligned} \tag{7}$$

The optimization problem can be formulated as (Davis et al., 2007)

$$\begin{aligned} \min_{A} \quad & \int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A_0) \log\left(\frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A_0)}{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A)}\right) d\mathbf{x} \\ \text{subject to} \quad & d_A(\mathbf{x}_i, \mathbf{x}_j), \leq u, \ (i,j) \in S \\ & d_A(\mathbf{x}_i, \mathbf{x}_j) \geq \ell, \ (i,j) \in D \\ & A \succeq 0 \end{aligned}$$

where S is the set of similar examples, D is the set of dissimilar examples, l and u are predetermined thresholds (determined by 5th and 95th percentile of distribution) for similarity and dissimilarity measures.

The optimization problem formulated above is equal to the half of the Logdet divergence in equation 6.

2.1.4 Relation to Kernel Learning

The distance metric learning problem can also be formulated as a kernel learning problem subject to the pairwise constraints between different points. Davis et al. (2007) show that to derive the optimum kernel from the learnt distance metric the following theorem can be used.

$$K_{optimum} = X^T A_{optimum} X \tag{8}$$

2.2 Label propagation

After the graph is constructed, I used the iterative label propogation algorithm (Zhu and Ghahramani, 2002) to propogate the labels from the labelled nodes to the unlabbled ones.

Estimated labels on both labeled and unlabeled data are denoted by $\hat{Y} = (Y^l, Y^u)$, where Y^l is not allowed to differ from the given labels. The algorithm can be summerized as follows, where W is the similarity matrix.

Compute the diagonal degree matrix $\mathbf{D}$ by $\mathbf{D}_{ii} \leftarrow \sum_j W_{ij}$
Initialize $\hat{Y}^{(0)} \leftarrow (y_1, \ldots, y_l, 0, 0, \ldots, 0)$
Iterate
1. $\hat{Y}^{(t+1)} \leftarrow \mathbf{D}^{-1}\mathbf{W}\hat{Y}^{(t)}$
2. $\hat{Y}_l^{(t+1)} \leftarrow Y_l$

until convergence to $\hat{Y}^{(\infty)}$
Label point x_i by the sign of $\hat{y}_i^{(\infty)}$

3 Empirical evaluation

The dataset used for empirical evaluation is the 20 Newsgroup dataset. Two classes (atheism and science) were used. Total dataset size was 1065.

• The dataset required some preprocessing which included Stemming, Lemmatization, Stop Word Removal, Conversion to lowercase of all the sentences in a document.

• After that term frequency-inverse document frequency (tf-idf) features were extracted.

• Principle Component Analysis (PCA) was applied to reduce the dimension of the feature vector from 6000 $\rightarrow$ 100. This was done to improve computational efficiency.

• Apart from $ITML$ and $LMNN$ the following distance metrics were used to construct the graph -: *Cosine distance*, *Covariance Matrix of the training set*, *Cosine distance on original data and Gaussian Kernel*.

• The graph for the label propagation was constructed by $k - Nearest\ Neighbour$ approach. The best k was selected by 4 fold cross validation.

• Each algorithm was tested on 783 test examples.

• The parameter C for SVM was chosen by doing a grid search on a log scale.

• The 100 components retained after PCA made no difference to the accuracy of the Linear SVM. This meant that the data is still separable to some degree.

• For all the algorithms (except for SVM) an asymmetric graph (with fixed k) was constructed.

• In order to incorporate the balance in classes of the labelled examples, Stratified K Fold cross validation was used.

• $LMNN$ and $ITML$ are the best known techniques for learning a distance metric matrix, hence they were used for this project.

• All the plots for the original data and the data transformed by the learnt metric are projected into 2D by using $t - SNE$ (Hinton and Bengio) (for visualization purposes only)

• To transform the data into the new space of the learnt metric function, Cholesky Decomposition of the learnt Mahalanobis matrix is taken.

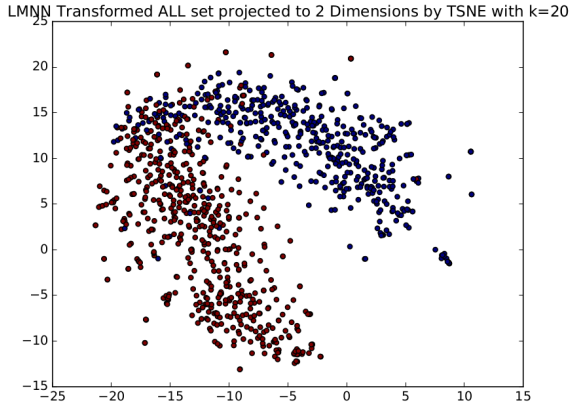
The following table 1 shows the results of the Semi-Supervised Learning Algorithms and SVM.

Labelled Examples	ITML	LMNN	Mahalanobis	Cosine	Cosine no PCA data	Gaussian Kernel	Linear SVM
262	87.6 %	91.1 %	46.9 %	90.8 %	90.6 %	55.2 %	90.4 %

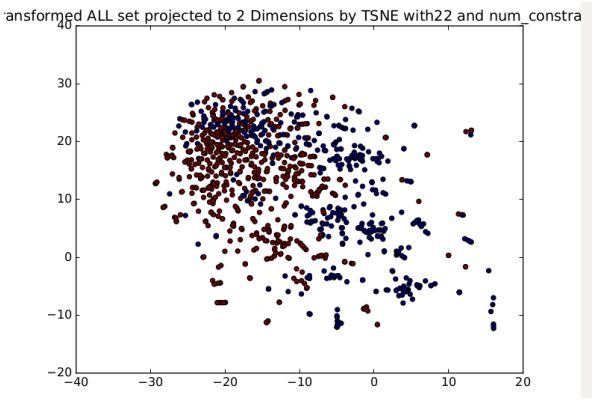
Table 1: Comparison between SVM and Different Similarity Matrices for Label Propagation

The various plots concerning the algorithms mentioned above are shown in the next page.

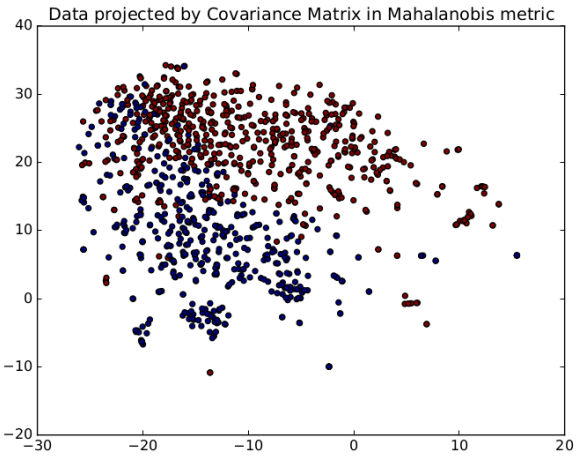
3.1 Plots



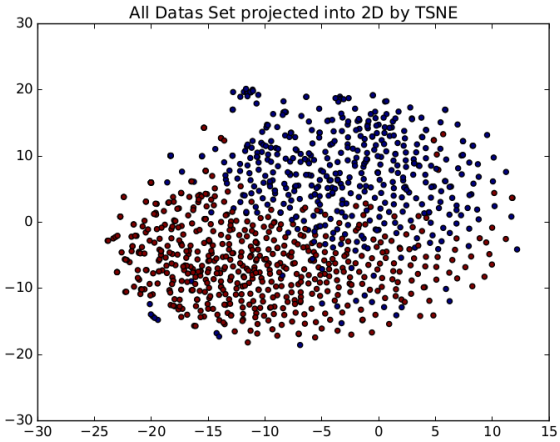
(a) Data Transformed by LMNN, with k=20



(b) Data Transformed by ITML, with k=22

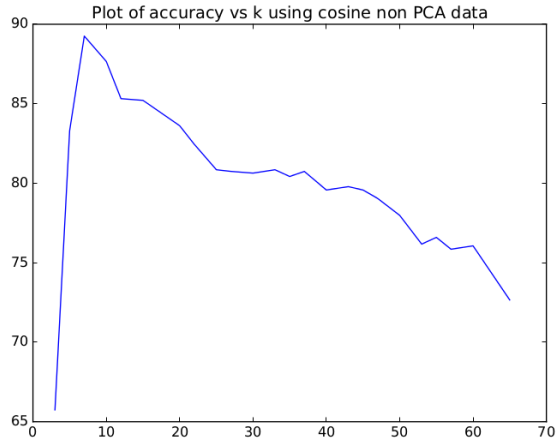


(c) Data Transformed by Cholesky Decomposition of Covariance Matrix of the training data, with k=60

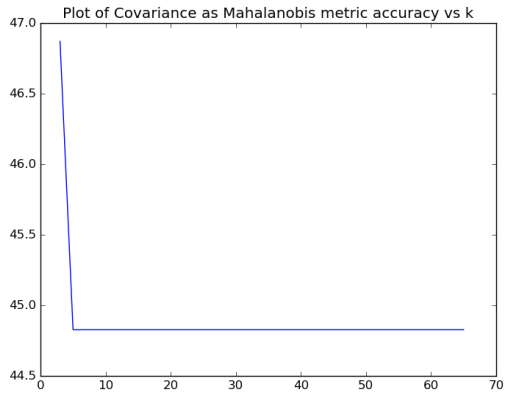


(d) Original Dataset

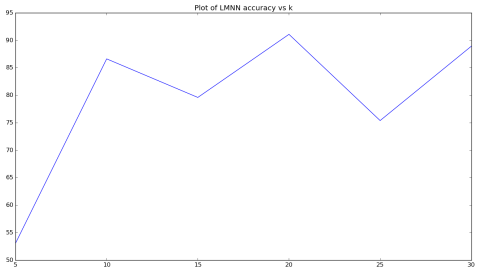
Figure 2: Original and Transformed Datasets in 2D by TSNE



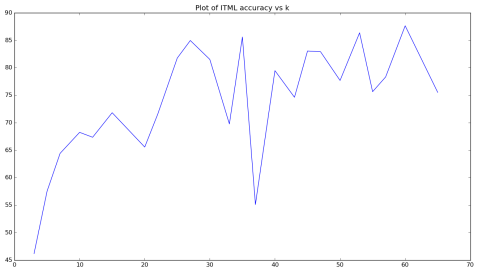
(a) Accuracy curve for Cosine Similarity on original data



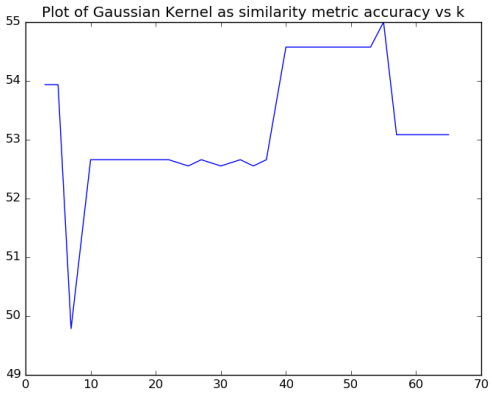
(b) Accuracy curve for Covariance Matrix as Mahalanobis metric on PCA transformed data



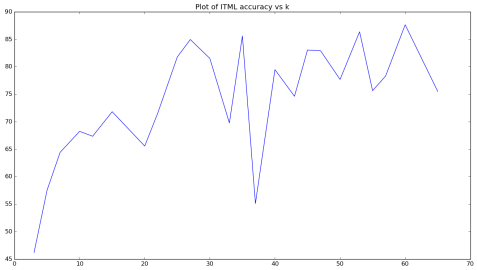
(a) Accuracy curve for LMNN Similarity on original data



(b) Accuracy curve for ITML Similarity on PCA transformed data



(c) Accuracy curve for Gaussian Kernel Similarity on original data



(d) Accuracy curve for ITML Similarity on PCA transformed data

Figure 4: Accuracy curves for various algorithms

4 Conclusion

From Table 1 it is clear that $LMNN$ beats the rest of the algorithms. Cosine based similarity matrix comes close and then SVM. $ITML$ gives a slightly lower accuracy than the three algorithms mentioned above, however it the the fastest in terms of convergence and computation.

As can be seen in 2 , the learned metric transformed datasets have much similar points closer to each other than in the original dataset.

This project explored the area of graph based semi supervised learning. Various different distance metrics were compared to one another.

LMNN though gave the best result, is very expensive to compute. It took 12 hours on my PC to run 4 folds of cross validation and a grid search on the parameters. Therefore, it is not scalable to large datasets. LMNN worked as expected. Below is visual representation of how it would work for a 2 dimensional case.

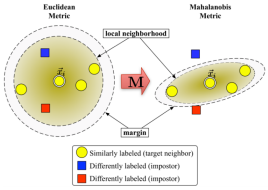


Figure 5: LMNN illustration (Weinberger and Saul, 2009)

ITML, on the other hand converged relatively very quickly (in 10-15 iterations). Though it is scalable to large datasets and also works well in online learning, it was not able to get an accuracy higher than LMNN or the Cosine Based Similarity.

I believe due to the nature of the dataset, Cosine Similarity worked well, though in general it will not be able to perform well. Basically the idea of Graph based Semi Supervised Learning is to label the unlabeled data with high accuracy and then use an inductive algorithm to generalize the test data.

4.1 Future Directions

Expectation Maximization (EM) algorithm can also be used in semi supervised setting. Nigam et al. (2000) showed the use of EM to improve upon the accuracy for Naive Bayes. Nigam et al. (2000) used counts and MAP estimates as features. In this project I used $tf - idf$ as feature vectors. I would want to experiment with Gaussian Naive Bayes and EM on a text dataset. Also, there a a few more inference algorithms for graph based settings which I would want to implement. Some examples include Measure Propagation (Subramanya and Bilmes, 2011) and Modified Adsorption (Talukdar and Crammer, 2009). I will be using these concepts for my thesis which will be on Natural Language Processing and Social Network Analysis.

References

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273523. URL `http://doi.acm.org/10.1145/1273496.1273523`.

Paramveer S. Dhillon, Partha Pratim Talukdar, and Koby Crammer. Learning better data representation using inference-driven metric learning. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 377–381, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858842.1858911`.

Geoffrey Hinton and Yoshua Bengio. Visualizing data using t-sne. In *Cost-sensitive Machine Learning for Information Retrieval 33*.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May 2000. ISSN 0885-6125. doi: 10.1023/A:1007692713085. URL `http://dx.doi.org/10.1023/A:1007692713085`.

Scikit-Learn. Semi-Supervised Learning Scikit Learn. `http://scikit-learn.org/stable/modules/label_propagation.html`. Accessed: 2016-04-08.

Amarnag Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. *J. Mach. Learn. Res.*, 12:3311–3370, November 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2078212`.

Talukdar and Subramanya. Graph Based Semi Supervised Learning Tutorial. `http://graph-ssl.wdfiles.com/local--files/blog%3A_start/graph_ssl_acl12_tutorial_slides_final.pdf`, 2012. Accessed: 2012-04-08.

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 442–457, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04173-0. doi: 10.1007/978-3-642-04174-7_29. URL `http://dx.doi.org/10.1007/978-3-642-04174-7_29`.

Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1577069.1577078`.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.