# Performance Comparisons of Semi-Supervised Learning Algorithms

**Te Ming Huang** HUANGJH@WIN.CO.NZ
School of Engineering, The University of Auckland, 20 Symonds Street, Auckland, New Zealand

**Vojislav Kecman** V.KECMAN@AUCKLAND.AC.NZ
School of Engineering, The University of Auckland, 20 Symonds Street, Auckland, New Zealand

## Abstract

We experimentally compare performances of five different methods for solving semi-supervised learning tasks proposed recently. In particular, we compare the Low Density Separation (LDS) algorithm with the original Consistency Method (CM) and Gaussian Random Fields Model (GRFM) as well as with the applications of the latter two to the new input's representation obtained by the graph-distance derived kernel. The experiments show that the efficiency of the method depends primarily upon whether one solves two- or multi-class recognition problem. For two-class problems, as long as the cluster assumption of data holds, LDS algorithm provides slightly smaller error, while for the multi-class semi-supervised tasks both CM and GRFM show superior performances. For the two-class problems without cluster structure of the data LDS algorithm is superior to the other ones used here.

## 1. Introduction

The two most popular machine learning groups are the so-called *supervised* and *unsupervised* learning methods. In the former a learning machine attempts to learn the input-output relationship (dependency or function) $f(\mathbf{x})$ by using a training data set $X = \{[\mathbf{x}(i), y(i)] \in \Re^m \times \Re, i = 1,...,n\}$ consisting of $n$ pairs $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, …, $(\mathbf{x}_n, y_n)$, where the inputs $\mathbf{x}$ are $m$-dimensional vectors $\mathbf{x} \in \Re^m$ and the labels (or system responses) $y \in \Re$ are continuous values for regression tasks and discrete (e.g., Boolean) for classification problems. In the *unsupervised learning* there are only raw data $\mathbf{x}_i \in \Re^m$ without the corresponding labels $y_i$ (i.e., there is a 'no-teacher' in a shape of labels). The most popular algorithms belonging to this group are various clustering and (principal or independent) component analysis routines. However, today we are frequently facing instances in which the learning is characterized by the presence of (usually) a small percentage of labeled data only. In this novel setting, the learning problem is to predict the label (or the belonging to some class) of the unlabeled data points. This task belongs to the so-called *semi-supervised* or *transductive inference* problems. The main reason for an appearance of the unlabeled data points is usually expensive, difficult and slow process of obtaining labeled data. Thus, labeling brings the costs and often it is not feasible.

Recently several approaches to the semi-supervised learning were proposed. Here, we compare the LDS algorithm as given in Chapelle and Zien (2005), with the CM as presented in Zhou et al. (2004), and with the GRFM as introduced in Zhu et al. (2003). Benchmarking LDS is challenging and it follows from the fact that Chapelle and Zien have shown its superiority in respect to five other semi-supervised methods, namely to the SVM, manifold, Transductive SVM (TSVM), graph and Gradient Transductive SVM (∇TSVM). (See the details in their paper). The LDS algorithm is an efficient combination of the last two mentioned methods namely, of the graph approach and ∇TSVM algorithm. In such a combination, one first calculates the graph-based distances that emphasize low density regions between clusters, and then a novel Chapelle-Zien's ∇TSVM algorithm which places the decision boundary in the low density regions is applied. More about the algorithms that will be compared is given in section 2 below. Because both CM and GRFM are the algorithms of the same type as the manifold method in Chapelle and Zien (2005), in the rest of the paper they will be referred to as the manifold approaches when discussed together.

Paper is organized as follows: in section 2 we introduce the methods to be compared. Section 3 describes the data sets used. Section 4 shows the experimental results obtained by the free-downloadable software for large scale semi-supervised learning SemiL (Huang and Kecman, 2004). The concluding section ends the presentations here and proposes possible avenues for the further research in this novel area of semi-supervised learning.

Appearing in *Proc. of the 22st ICML Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, August 2005, 

## 2. Algorithms Implemented

The LDS algorithm is developed on the strong belief that the cluster assumption for the data is necessary for a development of the successful semi-supervised learning algorithm. The cluster assumption is also present in the foundation of both the CM and GRFM algorithms, and in this respect all three methods are similar. The LDS algorithm is a two steps procedure – in the first step one calculates the graph-based distances that emphasize low density regions between clusters and, in the second part, by using the gradient descent, one optimizes the transductive SVM which places the decision boundary in low density regions. The latter is the property of the algorithm that gives its name. The authors claim that the combination of the two methods exploits the cluster assumption in the best possible way. By comparing LDS with five other state-of-the-art semi-supervised learning algorithms they actually show its superiority. The LDS algorithm involves tuning several parameters and for more details one should refer to Chapelle-Zien's paper.

The two competing algorithms here are the Consistency Method (Zhou et al., 2004) and Gaussian Random Fields Model (Zhu et al., 2003). Both approaches are also based on the belief that 'adjacent' points and/or the points in the same structure (group, cluster) should have similar labels. This can be seen as a form of regularization (Smola and Kondor, 2003) pushing the class boundaries toward regions of low data density similarly to LDS. This regularization is often implemented by associating the vertices of a graph to all the (labeled and unlabeled) samples, and then formulating the problem on the vertices of the graph (Krishnapuram, 2004). Both algorithms have similar property of searching the class boundary in the low density region and in this respect they have similarity with the ∇TSVM method too. Thus, it is somehow natural to compare the different algorithms developed around the same principles. This led us to using CM and GRFM to the same data sets as in Chapelle-Zien's paper. Similarly, it was a natural idea to replace the second part of the LDS (namely the ∇TSVM part) by both the CM and GRFM algorithms. Thus, the last two algorithms compared in this paper (and dubbed here with a prefix graph &) are the combinations of the graph-based distances with the CM and GRFM. (Recall that the LDS algorithm is the combination of the graph-based distances with the ∇TSVM method). More precisely, both CM and GRFM are applied to a new representation of $\mathbf{x}_i$ which is computed by performing multidimensional scaling to the matrix of squared $\rho$-path distances, i.e., steps 1 to 6 of the LDS algorithm in Chapelle-Zien's paper are used and then followed by CM or GRFM.

As pointed out by Huang and Kecman (2004), the performance of CM and GRFM can be affected by the balance of the labeled data. Data sets are considered as being balanced if each class has the same number of labeled data. It has also been shown that a misbalance in the labeled data can deteriorate the performance of both CM and GRFM significantly. This phenomenon can be understood by interpreting CM algorithm in terms of random walks on graph as shown in Zhou and Schölkopf (2004). In this setting, one can find the expected number of steps for a random walk starting at some initial position or vertex $x_i$ to reach $x_j$ and then to return. This expectation is often referred to as the commute time between the two positions and the CM algorithm uses a normalized commute time $\bar{\mathbf{C}}$ as a measure of closeness for classification (more details can be found in Zhou and Schölkopf (2004)). A lazy random walking is determined by the transition probability matrix $\mathbf{P} = (1-\alpha)\mathbf{I} + \alpha\mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{W}$ is the affinity matrix, $\mathbf{D}$ is a diagonal matrix with its $(i, i)^{\text{th}}$ element equal to the sum of the $i$-th row of $\mathbf{W}$ and $\alpha$ is a constant in (0,1). It has been shown that the normalized commune time satisfies

$$\bar{\mathbf{C}}_{ij} \propto \bar{\mathbf{G}}_{ii} + \bar{\mathbf{G}}_{jj} - \bar{\mathbf{G}}_{ij} - \bar{\mathbf{G}}_{ji} \text{ if } x_i \neq x_j ,$$

where $\bar{\mathbf{G}}$ is the inverse of the normalized Laplacian matrix $(\mathbf{I} - \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})$. If we now consider a binary classification that is given by $\mathbf{f} = (\mathbf{I} - \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})\,\mathbf{y}$, then the classification is based on the comparison between

$$p_+(x_i) = \sum_{\{j|y_j=+1\}} \bar{\mathbf{G}}_{ij} \quad \text{and} \quad p_-(x_i) = \sum_{\{j|y_j=-1\}} \bar{\mathbf{G}}_{ij}$$

as shown in Zhou and Schölkopf (2004). This means, we are labeling an unlabeled point by summing up and comparing the normalized commute times of the point to all the positive labeled points and to all the negative labeled points. With more positive labeled points than the negative ones, the mean of $\mathbf{f}$ will more likely to be greater than zero and vice verse. As a results, unlabeled data will be more likely to be classified into the class with more labeled data by the CM algorithm. GRFM is also based on the similar principle in classifying unlabeled data, but instead of using the normalized commune time, GRFM uses the hitting time in a random walk as a measure of closeness. As a result, its performance will be affected in the same way when the labeled data is unbalanced. To correct this problem, Huang and Kecman (2004) proposed a novel decision strategy dubbed as the normalization which improves the performances of both CM and GRFM substantially, in the cases when the labeled data is unbalanced. The normalization step normalizes the output $\mathbf{f}$ from each binary classification to have the mean of zero and standard deviation of one. Thus, each class is treated equally. In this work, the normalized versions of CM and GRFM will also be included and in Table 2, results obtained by them are denoted by an exponent $N$.

## 3. Test Data Sets

In this work, the same five data sets used in Chapelle and Zien (2005) are used for comparing the performances of various semi-supervised learning algorithms. Data are available at http://www.kyb.tuebingen.mpg.de/bs/people/chapelle/lds/). An overview of data sets can be found in Table 1. First, it should be said that the cluster assumption holds for all data sets except for the g10N data set. This

fact will show in the results of Table 2 where, for all four manifold algorithms, there will be the biggest error for g10N data set. Although the same data sets are used, the test setting used in this work is slightly different than the one implemented in Chapelle and Zien (2005). In order to make the results statistically more significant, the mean error rates (for the manifold approaches here) were calculated using 50 random splits of labeled and unlabeled data. The only exception is for the Coil20 data. In this data set, the four manifold algorithms under investigation are applied to the same 10 random splits used in Chapelle and Zien (2005). The reason for such a test setting is because Chapelle and Zien (2005) selected 2 labeled data from each class, i.e., the labeled data is balanced in each class and this may also alter the outcome of the simulations.

*Table 1*. Summary of test data sets

| DATA SET | CLASSES | DIMENSION | POINTS | LABELED |
|---|---|---|---|---|
| COIL20 | 20 | 1024 | 1440 | 40 |
| G50C | 2 | 50 | 550 | 50 |
| G10N | 2 | 10 | 550 | 50 |
| TEXT | 2 | 7511 | 1946 | 50 |
| USPST | 10 | 256 | 2007 | 50 |

In terms of model's parameters selections, and in order to reduce the computational time, we fixed some of the parameters in the algorithms and only considered combinations of values on a finite grid for the rest of the parameters. For the original CM methods, we fixed the $\alpha$ parameter to 0.99 and we only varied the $\sigma$ parameter which determines the width of the Gaussian functions used in calculation of the affinity matrix **W** as follows; $W_{ij} = \exp(-\| x_i - x_j \|^2 / 2\sigma^2)$ if $i \neq j$ and $W_{ii} = 0$. Because the $2\sigma^2$ value plays a major role to the performance of the manifold algorithms, we tried to find the optimal value of $2\sigma^2$ between 0.005 and 200,000. However, it is important to point out that in some problems, it is not possible to solve the problem for $2\sigma^2$ being small value such as 0.005, because the conditional number of the Laplacian and normalized Laplacian matrices used in the GRFM and CM algorithm respectively, will be very high and, consequently there will be problems with their inversions. In terms of the graph-based distance approach used in the LDS methods, we only tested the approach with values of $\rho$ equal to 1, 4 and 16. Also, the Chapelle-Zien's parameter $\delta$ is fixed at 0.1 and a full graph was used to construct matrix **D** of squared $\rho$-path distances. (Note that **D** mentioned for GRFM method is not the same matrix as the one used here). For all data sets, except for the Coil20 one, 10 and 100 nearest neighbors are used for a construction of the affinity matrix **W** which is needed for CM and GRFM. In the Coil20 data, using the 10 nearest neighbors would not produce a fully connected affinity matrix and only until the number of nearest neighbors exceeds 80, a fully connected affinity matrix could have been generated. The normalized versions of the CM and GRFM as proposed in Huang and Kecman (2004) have also been used in the simulations. All the simulations in this work (except for a Coil20 data experiments when a Matlab based code was used) are generated using the software package SemiL which is a software package designed to solve large scale semi-supervised learning problems using CM and GRFM. For the simulations that require calculation of graph-based distances, Matlab based code from Chapelle and Zien was used to generate a new representation of $\mathbf{x}_i$. The new representation of $\mathbf{x}_i$ is then the input for a SemiL routine.

## 4. Results

### 4.1 Performance Comparison Between LDS and Manifold Approaches

Table 2 shows the lowest error rates achieved by CM and GRFM based approaches for all the five data sets included in this study. The results for the LDS methods have been taken directly from Chapelle and Zien (2005) with 10 random splits and they are used as references. Basic observations are as follows. First, both CM and GRFM preceded by the calculation of the graph-based distances are better for the multi-class problems than LDS, while the latter one is (slightly) better for the two-class ones. Second, for the two-class problems, performance of GRFM is close to the results of LDS, and taking the stricter testing criterion used in our experiments (50 random runs compared to 10 ones in Chapelle-Zien's paper) they may be even or, there might be some advantages for GRFM method as long as the cluster assumption for the data is fulfilled. For the g10n data set, without the cluster structure, LDS perform much better than manifold methods as expected.

*Table 2*. Comparisons of the mean test error rates of five semi-supervised algorithms

| DATA SET | LDS | CM | GCM | GRFM | GGRFM |
|---|---|---|---|---|---|
| COIL20 | 4.86 | 8.9 | **1.5** | $9.83^N$ | 2.9 |
| G50C | **5.62** | $7.25^N$ | $7.38^N$ | $6.56^N$ | $6.84^N$ |
| G10N | **9.72** | $22.29^N$ | $23.66^N$ | $17.93^N$ | $20.8^N$ |
| TEXT | **5.13** | $13.6^N$ | $13.09^N$ | $7.27^N$ | $7.33^N$ |
| USPST | 15.8 | $9.74^N$ | **8.75**$^N$ | $10.69^N$ | $9.3^N$ |

LDS = Low Density Separation, CM = Consistency Method, GCM = Graph + Consistency Method, GRFM = Gaussian Random Fields Model, GGRFM = Graph + Gaussian Random Fields Model

For the Coil20 data set, the lowest error rate of only 1.5% is achieved by combining the graph-based distances and the CM. The improvement in performance as a result of using the graph-based distances for CM and GRFM is quite significant in this case from 8.9% to 1.5% (6 times better) and 9.83% to 2.9% (3.3 times better) respectively.

In this data set, both manifold based algorithms outperform the LDS approach and this coincides with the fact that manifold method used in Chapelle and Zien (2005) performs better than ∇TSVM which is the base classifier for the LDS method. The normalized model did not perform as well as the non-normalized model. This can be attributed to the fact that the $2\sigma^2$ value used here is very small ($2\sigma^2 = 0.005$) as well as to the use of balanced labeled data.

For a g50c data set, the LDS method performs the best. However, the difference in performance between the manifold methods and the LDS method is much closer (6.56% vs 5.62%) than the difference (17.3% vs 5.62%) shown in Chapelle and Zien (2005). Similarly, in the text data set, the performance difference is also shrunken from 11.71% vs 5.13% to 7.33% vs 5.13%. These changes are attributed mostly to the normalization step that lowered the error rate by reducing significantly the effect of the unbalanced data. Also, the error of 7.33% was obtained by the GRFM method that implements Laplacian matrix. The use of the graph-based distance does not significantly alter the performance in all data sets except for the Coil20 data and partly for the USPST data. The same phenomenon is presented in Chapelle-Zien's paper too. The answer to the question of why the algorithms behave in this manner requires more considerations in the future.

For g10n data set, the performance of LDS is better than the results of the manifold methods. This particular data set is generated in such a way that the cluster assumption does not hold. Therefore, it is not surprising that the manifold methods, relying strictly on the cluster assumption, have higher error rate. In contrast, LDS which is based on ∇TSVM performs much better than the manifold approaches. This may be due to the fact that ∇TSVM is based on the idea of margin maximization as SVMs which does not rely on the cluster assumption. Also, the incorporation of the graph-based distances does not help for the non-clustered data very much.

In the USPST data set, the normalized version of CM with graph-based distances achieved the lowest error rate of 8.75%. Also, the performances of all the manifold methods (with or without using the graph-based distances) are significantly better than the performance of the LDS method. This is again attributed to two causes; first manifold algorithms perform better for multi-class problems and second the normalization step helps in the case of unbalanced data. In this multi-class problem the use of graph-based distances also improves the performance of both CM and GRFM methods.

The simulation results suggest that incorporating graph-based distances to semi-supervised learning methods can bring more or less substantial performance improvement in multi-class problems only. These improvements can be found not just for the ∇TSVM as shown in Chapelle and Zien (2005), but also for the manifold approaches used here.

Another interesting trend is that using the graph-based distance with the manifold methods works the best when the value of $\rho$ is in the lower region meaning either 1 or 4 for all data sets. More investigations are needed to explain this phenomenon.

The reason why the manifold approach is better in the two multi-class problems may be due to the fact that manifold approaches perform global optimization over all $n$ classifiers, while the ∇TSVM designs separately $n$ classifiers by maximizing the margin of each classifier. The cost function of ∇TSVM is non-convex (Joachims, 1999), and it always finds some suboptimal solutions for each particular classifier. In addition to that, it is well known that the sum of suboptimal solutions can not and does not produce an overall optimum. Hence, the performance of ∇TSVM will not be as optimal as the manifold approach in multi-class problems.

### 4.2 Normalization Steps and the Size of $\sigma$ Parameter

From Table 2 it is clear that the normalized models dominated in the most of the data sets. Thus, it is important to know when and how the normalization step should be applied to the manifold algorithms.

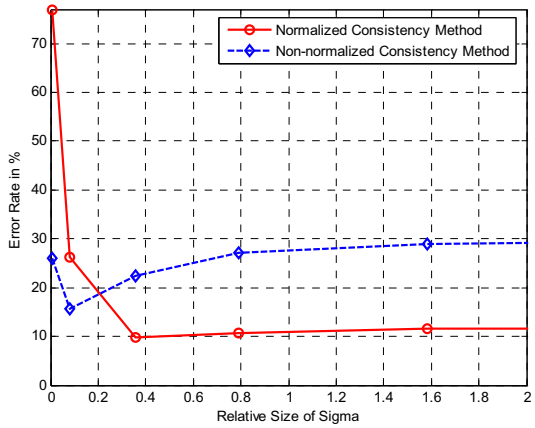


***Figure 1.*** The effect of normalization step and size of σ parameter on the USPST data set. The relative size of σ is calculated by finding out the ratio between 2σ2 and the mean value of all the non-zero elements in the affinity matrix W.

During the extensive simulations on these data sets, a very clear relationship between the size of $\sigma$ parameters and the performance of the normalized model is observed across all the data sets. Figure 1 shows the performance of the normalized CM and non-normalized CM with various $\sigma$ parameters on a 10 nearest neighbor graph for the USPST data set. The performance of both models stays relatively constants, as size of $\sigma$ gets larger than certain value.

However, with larger $\sigma$, the performance of the normalized model (error rate is 12%) is far more superior to the one of the non-normalized model (error rate is 30%). In contrast, the performance of the non-normalized

model is better than the one of the normalized model when $\sigma$ is relatively small. This means that the effect of unbalanced data discussed in Huang and Kecman (2005) is more noticeable as the size of $\sigma$ gets larger. This phenomenon can be explained as follows: when the $\sigma$ value is large, the influence of the distance between the data points becomes less important, because in such a setting even a distant pairs of point will have relatively large similarity in the affinity matrix **W**. As a result, this will make the classification of the unlabeled point dominated by the number of labeled points in each class. The normalized procedure tries to remove this effect by standardizing the output of **F**. This also explains why the non-normalized models perform better than the normalized ones in the coil20 data sets. This is because the size of $2\sigma^2$ used is only 1.7% of the mean value of the non-zero element in the **W** matrix.

The result shown in this section provides some guidelines as to when the normalization step can be used in relation with the $\sigma$ parameter in order to obtain better performance. It also shows two possible zones where the $\sigma$ parameter is optimal. For a given problem, one needs to compare the performance of the normalized model with relatively large $\sigma$, to the non-normalized model with relatively small $\sigma$ and a better model should be found.

## 5. Conclusions

In this work, four different manifold algorithms (basic CM and GRFM and their derivatives) are applied to five different test data sets and compared to the LDS method. We have found that the manifold algorithms have much better performance in both multi-class data sets, whereas the LDS performs slightly better for the two-class data holding cluster assumption. This may be due to the fact that the cost function of the manifold approach is convex, whereas the one for ∇TSVM is non-convex. Thus, the solution of ∇TSVM is not as optimal as the ones from manifold approaches. For the two-class data set without cluster structure (g10n) the LDS method performs much better than the manifold algorithms. This preliminary result suggests that the manifold algorithms may be more suitable for handling multi-class problems than the LDS and ∇TSVM methods. However, more investigations need to be done in the future in order to confirm the findings here and to explore the possibilities of the algorithms discussed.

By combing the graph-based distances and the manifold methods, the performance of the algorithms is greatly improved in multi-class data sets only. It seems that the use of the graph-based distances does not help the manifold approaches for the two-class problems.

This work also demonstrates when the normalization step can benefit the performance of the manifold method in relation to the choice of the shape (width) parameter $\sigma$. The results suggest that with relative large value of $\sigma$ parameter, the normalization can improve the performance of the algorithm substantially. On the other hand, when a relative small value of $\sigma$ is more appropriate for a given data set, the normalization procedure does not seem to provide significant improvements. This gives some guidance when performing the model parameters selection for the manifold type of algorithms discussed in this work.

## References


Chapelle O., Zien A., (2005). *Semi-Supervised Classification by Low Density Separatio*n, 10th International Workshop on Artificial Intelligence and Statistics, AI STATS 2005, Barbados

Huang, T.-M., Kecman, V., (2004). *Semi-supervised Learning from Unbalanced Labeled Data – An Improvement*, in 'Knowledge Based and Emergent Technologies Relied Intelligent Information and Engineering Systems', Eds. Negoita, M. Gh., et al., Lecture Notes on Computer Science 3215, pp. 765-771, Springer Verlag, Heidelberg

Huang, T.-M., Kecman, V., (2005). *Semi-supervised Learning from Unbalanced Labeled Data- An improvement*, Special Issue of KES International Journal, ISO press, Netherlands

Huang, T.-M., Kecman, V., (2004). *SemiL, Software for solving semi-supervised learning problems*, [Available from: http://www.support-vector.ws/html/semil.html ]

Joachims, T., (1999). *Transductive inference for text classification using support vector machines*, Proceedings of the Sixteenth International Conference of Machine Learning, pp.200-209.

Krishnapuram, B., (2004). *Adaptive classifier design using labeled and unlabeled data*, PhD Thesis, Dept of ECE, Duke University

Smola, A. Kondor, R., (2003). *Kernels and regularization on graphs*, COLT/Kernel Workshop

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., Schölkopf, B., (2004). *Learning with Local and Global Consistency*, Advances in Neural Information Processing Systems 16, (Eds.) Thrun, S., L. Saul and B. Schölkopf, MIT Press, Cambridge, Mass., pp. 321-328

Zhou, D., Schölkopf, B., (2004), *A Regularization Framework for Learning from Graph Data*, Workshop on Statistical Relational Learning at Twenty-first International Conference on Machine Learning.

Zhu, X.-J., Ghahramani, Z., Lafferty, J., (2003), *Semi-supervised learning using Gaussian fields and harmonic functions*, Proceedings of the Twentieth International Conference on Machine Learning, Washington DC