

# Semi-supervised Learning of a Markovian Metric

Avleen S. Bijral <sup>\*‡</sup>

Manuel E. Lladser <sup>‡§</sup>

Gregory Grudic <sup>\*¶</sup>

## Abstract

The role of a distance metric in many supervised and semi-supervised learning applications is central in the success of clustering algorithms. Since existing metrics like Euclidean do not necessarily reflect the true structure (clusters or manifolds) in the data, it becomes imperative that an appropriate metric be somehow learned from training or labeled data. Metric learning has been a relatively new topic in data mining and machine learning, though most work that deals with this topic learns a suitable linear transformation of the original data. This transformation is usually learned using training data and has been shown to improve test data classification accuracy. In this paper we present a Markov random walk based semi-supervised method for metric learning. Our method differs from the aforementioned techniques in that we use minimal labeled data and we do not assume any Mahalanobis type metric structure on the data. We create a computationally efficient nearest neighbor graph representation of the data and pose a semidefinite program that learns the random walk on the associated graph. This is used to generate a distance measure between all unlabeled points and the performance is compared against other important metrics using the k-NN classification rule.

**Keywords:** Metric learning, semi-supervised, discrete Markov chains, Markov random walks on graphs

## 1 Introduction

The problem of choosing an appropriate metric in data mining tasks is a central one. Classifier systems (semi-supervised or supervised) usually rely on a notion of similarity between data points in a collection, the most common being the Mahalanobis type metric of the form  $(x - y)^T M(x - y)$ , where  $x, y \in \mathbb{R}^n$  and  $M$  is positive definite. In the recent years, some approaches like [10] learn a linear transformation  $A$ , such that  $M = A^T A$ , using training data

and then evaluating it on test data. While such approaches have shown reasonable improvements in classification accuracies, it seems to us that the linearity assumptions forced by the choice of such metrics is limiting for many applications which have intrinsic nonlinearities.

Intuitively, a more realistic, albeit difficult approach would require minimal training data and little or no linearity assumption to learn an appropriate metric. In this paper we describe an approach along these lines.

Consider a set of data points  $X \subset \mathbb{R}^n$  where each data point  $x_i$  belongs to one of the known number  $C \geq 2$  of classes and at least one labeled point per class. The problem we pose is, for a connected graph with adjacency matrix  $H$  that imposes a certain structure on the given data, to learn suitable Markovian transition probabilities on the edges based on the minimum absorption time criteria to be described later. Our experiments show that using edge weights derived from this scheme instead of just locally Euclidean distances results in superior classification accuracy.

The problem is inspired from the paradigm of semi-supervised learning [4, 5, 6], wherein the data points are assumed to lie on some underlying manifold or cluster. The key assumption that these algorithms make is that nearby points and also points in the same manifold are likely to have the same label. The process of classifying the unlabeled points involves taking the few labeled points and then spreading the labels on some approximation of the manifold, usually a nearest neighbor graph.

In this paper we use the concept of absorbing Markov random walks on a graph to firstly derive a simple semi-supervised algorithm. We show how to learn an optimal transition probability matrix  $P$  by solving an optimization problem based on the ideas from the algorithm. These probabilities are used to construct a distance measure between all points. To the best of our knowledge the approach of learning a metric with no generic form (e.g. Mahalanobis) using semi-supervised learning has not been previously attempted in the literature.

We feel that the ideal testing algorithm for a distance measure is the k-NN rule [18], since it often gives competitive results and its performance is a direct indicator of the quality of the distance measure used. In the experimental section we compare the performance of the  $k$ -NN classifier on our distance matrix against some other significant techniques and show superior results.

<sup>1</sup>Department of Computer Science, University of Colorado at Boulder, 430 UCB, Boulder, CO 80309-0430.

<sup>2</sup>Department of Applied Mathematics, University of Colorado at Boulder, PO Box 526 UCB Boulder, CO 80309-0526.

<sup>3</sup>avleen.bijral@colorado.edu

<sup>4</sup>manuel.lladser@colorado.edu

<sup>5</sup>gregory.grudic@colorado.edu

In the next section we present the basic intuition of the proposed technique by first describing a simple semi-supervised binary classifier and constructing the necessary machinery for the optimization problem. Later we describe a method to pose a tractable solution to the problem and test the solution obtained on different types of datasets using comparisons with benchmark metrics.

## 2 Absorbing Markov Chains and Semi-supervised learning

Consider a set of  $n$  unlabeled points  $x_1, \dots, x_n$  and  $m$  labeled points  $x_{n+1}, \dots, x_{n+m}$ . For simplicity and clarity, we assume two classes of points, the classes correspond to labels  $+1$  and  $-1$ . The  $m$  labels contain labels from both classes.

We now construct a  $K$  nearest neighbor graph for our  $n + m$  points, where each point is connected to only its  $K$  nearest neighbors. Let the adjacency matrix for the graph be  $W$ . The weight of each edge is the Euclidean distance between the corresponding nodes or points. To convert this into a probability transition matrix, first we compute the affinity matrix  $A$ , where the affinity between points  $x_i$  and  $x_j$  is inversely related to the distance between these points, similar to the approach in [6], such that

$$(2.1) \quad A_{ij} = \begin{cases} e^{-W_{ij}} & , \text{ if } i \neq j; \\ 0 & , \text{ otherwise.} \end{cases}$$

For a Markov random walk defined on this graph, where the nodes are the states, there is a corresponding transition probability matrix  $P = D^{-1}A$ . The diagonal matrix  $D$  corresponds to the normalizing factor so that  $P$  is a stochastic matrix i.e.  $D_{ii} = \sum_j A_{ij}$ , for all  $i$ .

A Markov process defined on our graph would be absorbing [7] if one or more nodes were absorbing. This implies that the probability of exiting absorbing nodes will be 0 and the probability of a self-loop would be 1. All the other nodes correspond to transient states. Mathematically, node  $i$  is absorbing if  $P_{ii} = 1$  and  $P_{ij} = 0$ , for all  $j \neq i$ .

To transform this idea to a semi-supervised setting, we mark the labeled points as absorbing states. Now we have two sets of absorbing states, each corresponding to one class. We partition the matrix  $P$  with respect to these two sets of absorbing states as follows

$$(2.2) \quad P_+ = \begin{bmatrix} Q_+ & R_+ \\ 0 & I \end{bmatrix},$$

and

$$(2.3) \quad P_- = \begin{bmatrix} Q_- & R_- \\ 0 & I \end{bmatrix}.$$

Here  $R_+$  and  $R_-$  correspond to each of the blocks of the two sets of labeled points in the full transition matrix

respectively. Similarly  $Q_+$  and  $Q_-$  correspond to the blocks of transient states. For simplicity of notation we assume equal number of labeled points for both classes.

It is known [7] that the vector of expected number of steps before absorption of the  $n$  unlabeled points by the two different sets of absorbing states is given as

$$(2.4) \quad T_+ = (I - Q_+)^{-1}e,$$

$$(2.5) \quad T_- = (I - Q_-)^{-1}e,$$

where  $e$  is the vector of all 1's. The expected number of steps provides us with an intuitive measure of discriminating between the two classes. For any given point, the lower the average number of steps before absorption, the higher should be the probability of it belonging to the respective class. It is also evident that if the number of labeled points increase the likelihood of absorption will also increase. Thus, we can get our simple classifier with hard assignments for an unlabeled point  $x_i$  as

$$(2.6) \quad c_i = \text{sgn}(T_+^i - T_-^i).$$

## 3 Learning Transition Probabilities

It follows from the discussion in the previous section that the expected time of absorption may be used as an indicator of the classification label for an unlabeled point. Since intuitively, the time of absorption depends on the geometry of the underlying data graph, the random walk on the graph should be such that the absorption time is minimized for each unlabeled point. In a general multiclass scenario, we would like to minimize the absorption time across all classes. In particular, if  $P$  is the transition matrix for all data points (labeled and unlabeled) and  $Q$  is the corresponding sub-stochastic sub-matrix of all the unlabeled or transient states, we have

$$(3.7) \quad P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}.$$

The absorption time vector for all the unlabeled points is then  $T = (I - Q)^{-1}e$  and based on our heuristic the optimal choice for  $Q$  should correspond to the solution of the problem

$$(3.8) \quad \begin{aligned} & \min \|T\|_2 \\ & \text{s.t. } Qe \leq e; \\ & \quad Qe \neq e; \\ & \quad Q \geq 0; \\ & \quad Q_{ij} = 0 \text{ if } H_{ij} = 0. \end{aligned}$$

Above we minimize the total absorption time subject to the constraints that  $Q$  is strictly sub-stochastic with non-negative entries and that it respects the given adjacency matrix  $H$  of the original graph. For computational efficiency we only

consider nearest neighbor graphs and we do not require  $P$  and hence  $Q$  to be symmetric. It was noted in [15] that forcing symmetricity can obscure cluster structure.

From the optimization criterion, it is evident that the total absorption time depends on the entries of  $(I - Q)^{-1}$ . Since on the other hand

$$(3.9) \quad \|(I - Q)^{-1}e\|_2 \leq n \|(I - Q)^{-1}\|_2,$$

from standard matrix norm inequalities [8, 9], the above problem may be relaxed to minimizing  $\|(I - Q)^{-1}\|_2$ . Now, it can be shown that for a connected graph and an associated and strictly sub-stochastic matrix  $Q$ ,

$$\begin{aligned} \|(I - Q)^{-1}\|_2 &= \left\| \sum_{k=0}^{\infty} Q^k \right\|_2 \leq \sum_{k=0}^{\infty} \|Q^k\|_2, \\ &\leq \sum_{k=0}^{\infty} \|Q\|_2^k. \end{aligned}$$

Thus a related optimization problem but perhaps more tractable than (3.8) is

$$(3.10) \quad \begin{aligned} \min \quad & \|Q\|_2 \\ \text{s.t.} \quad & Qe \leq e; \\ & Qe \neq e; \\ & Q \geq 0; \\ & Q_{ij} = 0 \text{ if } H_{ij} = 0. \end{aligned}$$

But notice that only those rows of  $Q$  corresponding to points that have a labeled point as a nearest neighbor will be strictly sub-stochastic, all others will sum to one. Since we are assuming only very few or at least one labeled point per class (as compared to a large number of unlabeled points), those sub-stochastic rows will sum close to 1. We could therefore relax the above problem even further by forcing  $Q$  to be stochastic i.e.  $Qe = e$ . This leads us to the optimization problem

$$(3.11) \quad \begin{aligned} \min \quad & \|Q\|_2 \\ \text{s.t.} \quad & Qe = e; \\ & Q \geq 0; \\ & Q_{ij} = 0 \text{ if } H_{ij} = 0. \end{aligned}$$

The above problem can be written as an instance of a semidefinite program (SDP) by introducing a scalar  $t$  to bound the norm of the objective function as follows

$$(3.12) \quad \begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \begin{bmatrix} tI & Q \\ Q^T & tI \end{bmatrix} \succeq 0; \\ & Qe = e; \\ & Q \geq 0; \\ & Q_{ij} = 0 \text{ if } H_{ij} = 0. \end{aligned}$$

The semidefinite constraint stems from a fact about Schur complements [12, 13], that

$$t \geq \|Q\|_2 \text{ if and only if } \begin{bmatrix} tI & Q \\ Q^T & tI \end{bmatrix} \succeq 0.$$

In the standard SDP format we can write the inequalities in the above problem using a linear matrix inequality (LMI) as

$$(3.13) \quad \begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \text{diag}\left(\begin{bmatrix} tI & Q \\ Q^T & tI \end{bmatrix}, Q(\cdot)\right) \succeq 0; \\ & Qe = e; \\ & Q_{ij} = 0 \text{ if } H_{ij} = 0. \end{aligned}$$

Since the above is a semidefinite program, it is also convex and therefore a global minimum can be computed efficiently.

It is worth to mention here that the optimal solutions to problems (3.10) and (3.11) may be different because in the former we require  $Q$  to be strictly sub-stochastic. This simplification is justified due to the small number of labeled points as compared with the unlabeled ones. Indeed, back to the original optimization problem (3.8), when  $Q$  is stochastic the inverse of the matrix  $(I - Q)$  does not exist and the upper bound that lead to (3.10) is infinite. A mathematical justification to go from problem (3.10) to (3.11) is that the expected time starting from state  $i$  that the random walk stays in the transient class after some finite but large number of steps  $M$  is given by the  $i$ -th entry of  $\sum_{k=1}^M Q^k e$ . This allows us to not deal with the inverse matrix of  $(I - Q)$  directly i.e.  $M = +\infty$  yet retaining the intuition about the optimal choice for  $Q$ .

Now the edge weights that denote the distance between neighboring nodes should be function of the probabilities of optimal transitions. The higher the probability of transition between nodes, the smaller should be the distance between them. In most spectral clustering type and semi-supervised algorithms the affinity matrix is usually obtained by taking the exponential of the normalized edge weights. So to determine the weights between neighboring nodes we compute

$$(3.14) \quad W_{ij} = \begin{cases} e^{-Q_{ij}^*} & , \text{ if } i \neq j; \\ 0 & , \text{ otherwise.} \end{cases}$$

To obtain the distance between all pairs of points we use the Dijkstra's shortest path algorithm [22], which can be efficiently computed in  $O(|E| \log |V|)$ , where  $|E|$  and  $|V|$  are the number of edges and vertices respectively. Note that Dijkstra's algorithm assumes a connected graph. This can be guaranteed by choosing a  $K$  (for Nearest Neighbor graph) based on the cross validation procedure. (For a  $K$  resulting in a disconnected graph, some distances would be infinite.)

## 4 Experiments

**4.1 Synthetic Data** As a sanity check we test the distance matrix obtained on two generated examples, the two moon dataset (figure 1) and two circles as shown in figure 2.

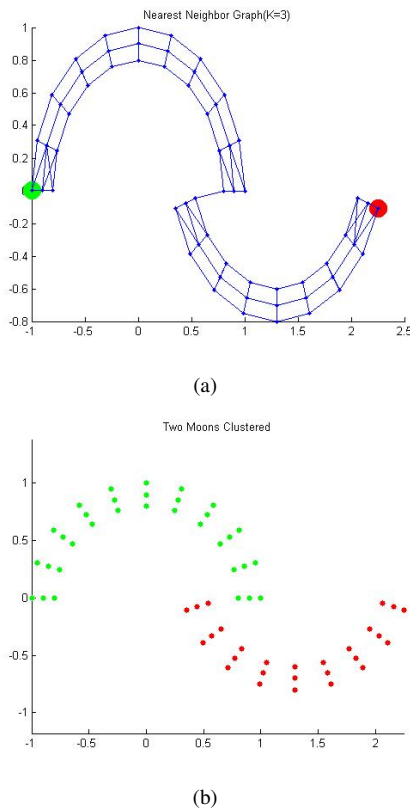


Figure 1: c) Two Moons Nearest Neighbor Graph d) The actual clustering

The graph was obtained with  $K = 3$  for two moons and  $K = 4$  for the circles and the clustering was produced with the k-nn rule using the shortest path distance as described above. This example shows that the optimization problem is able to learn the intrinsic structure of the synthetic sets with two labeled points. In the next section we conduct tests on real world datasets.

**4.2 Real Data** For all experiments a nearest neighbor graph was used for efficient computation. The value of  $K$  was manually tuned and the adjacency matrix  $H$  was obtained using standard Euclidean distances. We used the SDPT solver [21] for solving the semidefinite program. It is based on an interior point method and can handle around a few 1000 or so edges. This limits the size of the datasets we can test our approach on, providing an avenue for future where we intend to study scalable solutions for large datasets. Note however that the optimization problem

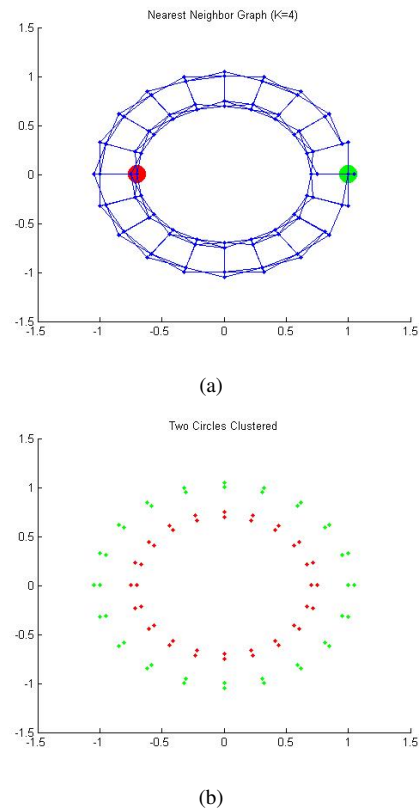


Figure 2: c) Two Circles Nearest Neighbor Graph d) The actual clustering

doesn't depend directly on the dimensionality of the data and that the metric is not limited by its form, Mahalanobis etc.

One ideal way of evaluating a metric is the classification accuracy using the k-NN classification algorithm as discussed before. We use this approach across all experiments. The value of  $k$  for classification was chosen using cross-validation and a different value is selected for different datasets for each comparison.

**4.2.1 USPS Digits** We conduct tests on various sub samples of the 16x16 USPS digits dataset. We divide the training set into subsets of group of 4 digits according to the relativity difficulty of classification. Digits (1, 2, 3, 4) are easier to tell apart than (5, 6, 7, 8). Thus naturally we use these two sets of size 104, with 26 digits in each class. The subsets used are USPS-1 (1,2,3,4), USPS-2 (5,6,7,8), USPS-3 (3,5,8,9) and USPS-4 (1,3,7,8). For our experiments this number basically set the limit for the SDP solver, since the number of constraints generated tend to be very high and these solvers do not scale well with the number of constraints. One digit each is randomly labeled to get the graph structure as described in the second section. For the remaining 100 digits, a 100x100 distance matrix is learned which is evaluated us-

k-NN error(%)	Euclidean	Tangent	Absorption
USPS-1	12.72	11.34	8.09
USPS-2	12.64	11.85	10.32
USPS-3	14.03	11.59	10.56
USPS-4	9.69	8.17	5.88
AT&T	5.95	5.80	4.55
k-NN error(%)	Euclidean	Mahdist	Absorption
Wine	27.78	18.56	22.96
Iris	5.77	9.30	3.33
Balance	21.81	21.33	21.44

Table 1: Sample test error rates for datasets using k-NN classifier.

ing the k-NN rule.

For the test accuracy, the results are averaged over 100 random 80/20 splits of the remaining 100 digits. The results are compared against standard Euclidean and the tangent distance [17], which is an ideal comparison especially for the USPS data. A C implementation is available [23]. These results are then averaged over 10 randomly selected different subsets of the above mentioned digits. Table 1 shows the test classification accuracies. Our measure(absorption metric) clearly results in substantial improvement over Euclidean and the tangent distance. We don't show comparisons with the standard Mahalanobis distance [20] on the high dimensional datasets as they apparently don't do well if the number of points are less than the dimensionality of the data.

**4.2.2 AT&T Faces** The AT&T face recognition dataset [24] contains 400 grayscale images of 40 persons in 10 different poses. The size of each image is 92x112. We used each image without down sampling. We selected a sub sample of 10 persons. Training and test images for each person were created by randomly choosing 7 images for training and 3 images for testing. The value for  $k$  was set to 3 for the test classification. The results were averaged over 100 iterations. This was then averaged various subsets of 10 people. Comparisons were made against standard Euclidean and Tangent distance. The results are shown in table 1. The results seem to indicate that the performance improvement is significant on data which can have some intrinsic manifold structure like digits and faces etc.

**4.2.3 UCI Data** We also tested our metric on measure on the wine,iris and balance datasets from the UCI machine learning repository [25]. The iris(4 attributes) and wine (4 attributes) datasets were divided into 80 and 20 training and

test points respectively belonging to 3 classes. A subset of the balance dataset with 4 attributes was split into 60 and 20 points belonging to 3 classes. The results were as usual averaged over 100, 70/30 splits of the subsets of the data, the error on each subset in turn being averaged 100 times. The value of  $k$  for the nearest neighbor classifier was obtained using cross validation. In these experiments we also compared against the standard Mahalanobis distance (Mahadist) using the inverse of the covariance matrix as a weighting [20].

## 5 Analysis

Our experiments reveal that except for the balance and wine dataset, we consistently outperform the other important metrics. As discussed before we essentially minimize a measure on the geometry of the underlying graph. This could indicate that we tend to do better on relatively higher dimensional datasets where the metric probably learns some intrinsic manifold structure wherein most points lie. In lower dimensions like for wine and balance datasets, the structure is possibly close to being linear and little or no improvement can be expected using our technique.

## 6 Related Work

There is existing work on learning a metric that mostly deals with a Mahalanobis type metric learned using substantial training data. Various formulations have been proposed, some from a probabilistic point of view [19], others more attuned towards minimizing the error margin on test data e.g. [10, 11]. From a k-NN rule perspective and hence with similar goals as our technique, the recent large margin nearest neighbor technique [10], trains the output Mahalanobis metric with the goal that the  $k$  nearest neighbors always belong to the same class and at the same time points from other classes remain separated by a large margin. From our preliminary tests it wasn't fair to compare this with our measure, since it seems to rely on larger training sets for significant improvement as is also evident from the experiments given in the paper.

As discussed before we are not aware of any technique that learns a generic metric from very few labeled points (based on some structural assumptions) and our research could lead to further improvements in this direction.

From a random walk perspective, related Markov methods have been employed for the purposes of classification. More specifically, [1] assumes a metric to arbitrarily convey a probability transition metric (proportional to weights) over the data to cluster it; [2] also assumes a metric from the beginning to arbitrarily claim a probability transition matrix based on the original metric which is then used to propagate labels using accumulated correlations; [3] also assumes a metric over the data to induce a probability transition metric over an  $M$ -NN graph (where labeled points are absorbing)

which is used to produce a probability distribution over the labels of each unlabeled point. While our setting is similar to these three references, we only assume a metric to embed our data in a manifold (chosen but not necessarily restricted to be a k-NN graph) but then use a probabilistic argument and optimization procedure to learn a more reasonable metric over that manifold.

## 7 Conclusion and Future Work

In this paper we presented a novel semi-supervised framework for learning a metric on a graph described by a given adjacency matrix. The technique employs the methodology of absorbing Markov chains wherein the few labeled points are described as absorbing and the unlabeled points as transient states. A semidefinite optimization problem is posed that minimizes the total absorption time for the unlabeled points. This problem is posed with structural constraints corresponding to the given data graph and other probabilistic ones. In the true form it is deemed intractable and a relaxation is proposed that preserves the original intuition. The optimization returns a random walk described by a stochastic matrix  $Q^*$  which is optimal in sense of the underlying geometry of the data. The suitability of our approach is then demonstrated by obtaining shortest path distances on real and synthetic datasets.

For future work we intend to fore mostly scale the optimization to large datasets. Some of our preliminary studies have indicated that sub-gradient methods [14] hold promise towards that direction. We also intend to clear some theoretical issues, like the effect of the number of labeled points on the learned random walk. There is also the need to study alternative relaxations of the problem that might lead to more efficient solutions.

## References

- [1] D. Zhou and B. Schlkopf. *Learning from Labeled and Unlabeled Data Using Random Walks*. Pattern Recognition, Proceedings of the 26th DAGM Symposium, 237-244. (Eds.) C.E. Rasmussen, H.H. Blthoff, M.A. Giese and B. Schlkopf, Springer, Berlin, Germany, 2004.
- [2] Xueyuan Zhou and Chunping Li *Text Classification by Markov Random Walks with Reward*. , DMIN05, Las Vegas, USA, 2005.
- [3] A. Azran. *The Rendezvous Algorithm: Multiclass Semi-Supervised Learning with Markov Random Walks*. Proc. International Conference on Machine Learning 2007 (ICML07).
- [4] M. Belkin and P. Niyogi, *Semi-supervised learning on manifolds*, Machine Learning, 56, 209-239, 2004.
- [5] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schlkopf, *Learning with local and global consistency*, In Advances in Neural Information Processing Systems (NIPS), volume 16, 2004.
- [6] M. Szummer and T. Jaakkola, *Partially labeled classification with markov random walks.*, In Advances in Neural Information Processing Systems (NIPS), volume 14, 2002
- [7] P. Bremaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer Verlag, May 1999.
- [8] R. A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.
- [9] Y. Saad, *Matrix Computations*, Johns Hopkins University Press 3rd edition, 1996.
- [10] Kilian Q. Weinberger, J.Blitzer and L.K.Saul, *Distance Metric Learning for Large Margin Nearest Neighbor Classification*, In Advances in Neural Information Processing Systems (NIPS), volume 17, 2006.
- [11] C.Domeniconi, D.Gunopulos and J.Peng, *Large Margin Nearest Neighbor Classifiers*, IEEE Transactions on Neural Networks, 16(4):899-909, 2005.
- [12] M.J. Todd, *Semidefinite Optimization*, Cambridge University Press, Acta Numerica 10 (2001) 515-560.
- [13] L. Vandenberghe and S. Boyd, *Semidefinite Programming*, SIAM Review, 38(1):49-95 March 1996.
- [14] S.Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [15] M. Meila and W. Pentney, *Clustering by Weighted Cuts in Directed Graphs*, In proceedings SIAM Conference on Data Mining, 2007.
- [16] P. Simard, Y. LeCun, J.S. Denker and B. Victorri, *Efficient Pattern Recognition Using a New Transformation Distance*, Neural Networks: Tricks of the Trade, 1996 NIPS Workshop.
- [17] P. Simard, Y. LeCun, J.S. Denker, *Transformation Invariance in Pattern Recognition-Tangent Distance and Tangent Propagation*.In Advances in Neural Information Processing Systems (NIPS), volume 6, 1993.
- [18] T. Cover and P.Hart, *Nearest Neighbor Pattern Classification*, In IEEE Transactions in Information theory, IT-13, pages 21-27, 1967.
- [19] J. Goldberger, S.Roweis, G.Hinton and R.Salakhutdinov, *Neighborhood Components Analysis*, In Advances in Neural Information Processing Systems (NIPS), volume 17, 2006.
- [20] P.C. Mahalanobis, *On the Generalized Distance in Statistics*, Proceedings of the National Institute of Science, India 12(1936) 49-55.
- [21] K.C. Toh, M.J. Todd and R.H. Tutuncu, *SDPT3 version 4.0 (beta) – a MATLAB software for semidefinite-quadratic-linear programming*, <http://www.math.nus.edu.sg/~mat-tohkc/sdpt3.html>.
- [22] E. W. Dijkstra, *A note on two problems in connexion with graphs*. In: Numerische Mathematik. 1 (1959), S. 269-271
- [23] D. Keyser. *A Tangent Distance Implementation*. [www.iupr.org/~keyser/files/i6/td/](http://www.iupr.org/~keyser/files/i6/td/)
- [24] AT&T. *Databases of Faces*. [www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html](http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html)
- [25] UCI. *Machine Learning Repository*. [mllearn.ics.uci.edu/MLRepository.html](http://mllearn.ics.uci.edu/MLRepository.html)