Regularized Minimum Volume Ellipsoid Metric for Query–based Learning

Karim Abou-Moustafa and Frank Ferrie
The Artificial Perception Laboratory
Centre for Intelligent Machines, McGill University
3480 University street, Montreal, QC, Canada H3A 2A7
{karimt,ferrie}@cim.mcgill.ca

Abstract

We are interested in learning an adaptive local metric on a lower dimensional manifold for query–based operations. We combine the concept underlying manifold learning algorithms and the minimum volume ellipsoid metric to find the nearest neighbouring points to a query point on the manifold on which the query point is lying. Extensive experiments on various standard benchmark data sets in the context of classification showed very promising results when compared to state of the art metric learning algorithms.

1. Introduction

Query–based operations are used in a plethora of algorithms in the literature on machine learning, pattern recognition, computer vision, image retrieval and data mining. A typical scenario is to have a set of high dimensional vectors (images, image patches, feature vectors, etc) where it is required to find a set of nearest neighbors or matching points to a query point. The Euclidean distance is usually the measure of choice for assessing the similarity between points, but despite its success in many applications, there are a few reasons to doubt the full validity of this metric when dealing with high dimensional real life data.

One reason is the curse of dimensionality; that is, in high dimensional spaces, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions [5]. Hence the notion of similarity becomes less accurate in such high dimensional spaces. The second reason stems from the typical characteristics of real life data: 1) High dimensional, highly structured and nonlinear such as images, text documents, proteins, etc. 2) Measured from various sources at different scales and with various degrees of variability and correlation, and 3) Prone to various sources of noise that may largely deviate measurements and raise outliers in the data. These combines characteristics will be referred to as *"data complexity issues"*. The third reason which stems from the definition of the Euclidean distance, combines and builds over the two aforementioned ones. That is, by expanding the squared Euclidean norm $\|\mathbf{x}-\mathbf{y}\|_2^2$ to $(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T\mathbf{I}(\mathbf{x}-\mathbf{y})$, where $\mathbf{I}$ is the identity matrix, one directly obtains an instance of the general family of Mahalanobis distances between points $\mathbf{x}$ and $\mathbf{y}$: $D_S(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})$, where $\mathbf{S}$ is a symmetric and positive definite matrix. Replacing $\mathbf{S}$ by $\mathbf{I}$ implies that the Euclidean distance takes it for granted that all variables are independent, the variance across all dimensions is one and that covariances among all variables are zero–a situation that is hardly attained in real life data. Therefore, the Euclidean distance, by definition, ignores the structure, scale, variance and correlations in the data and consequently it is wise to say that *"in the absence of clear evidence of Euclidean geometry, the metric structure should be inferred from the data"* [13].

Contribution : We are interested in learning a metric for query–based operations. We combine the concepts underlying manifold learning algorithms and the minimum volume ellipsoid metric (MVEM) [1] in a unified algorithm that tries to overcome the aforementioned problems. That is, given a data set $\mathcal{X}$ of some high dimensional points and a query point $\mathbf{x}_q$ of similar dimensionality, we are interested in learning a similarity measure based on the information in $\mathbf{x}_q$ and the data set $\mathcal{X}$; i.e. that is adaptive for each new query point, such that each $\mathbf{x}_q$ can better define its nearest neighbors or matching points from $\mathcal{X}$. This is different from previous metric learning algorithms that focused on learning a metric specifically for k–NN (nearest neighbors) classification, exemplified by [19, 12, 22], in that the proposed algorithm is unsupervised, self–adaptive for each new query point, and defines the metric on the lower dimensional manifold on which the query point is lying.

Our paper will proceed as follows: Section 2 will briefly review some related work in the literature. Section 3 will analyze the minimum volume ellipsoid metric, discuss its drawbacks and proceeds with our new proposed algorithm;

finally, experimental results are presented in Section 4 and concluding remarks are drawn in Section 5.

2 Related Work

The earliest work on metric learning was done by to Short and Fukunaga [19] where they define an optimal distance measure to minimize the difference between the empirical k-NN error (on a finite sample) and the asymptotic k–NN error (the twice optimal Baye's error bound). More recently, three main streams dominate the literature on metric learning; global metric learning using labels or similarity constraints (side–information), locally supervised metric learning similar to [19], and unsupervised metric learning exemplified by [20, 17, 14]. The first stream of metric learning will be briefly reviewed and the interested reader can see [24] for a comprehensive review on the subject.

Most of the algorithms in the first stream learn a global metric through the general family of Mahalanobis distances $\mathrm{D}_A(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})$ and the differences between these algorithms are due to the constraints defining each metric. In metric learning using labels, [10] defines a differentiable probability function (softmax) using $\mathrm{D}_A(\mathbf{x},\mathbf{y})$ with $\mathbf{A}$ as its parameter. This function is optimized to maximize the probability of correct classification using the labels in the training set. In an extended work, [9] uses the same objective function to map all points that belong to the same class into a single point. Alternatively, [22] searches for a matrix $\mathbf{A}$ that defines a linear transformation such that k–NN of the same class are always kept together while samples from other classes are separated by a margin.

In learning with similarity constraints, [18] uses similarity constraints in the form of triplet relative comparisons; i.e. for samples $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ the relative comparison information is in the form of $\mathrm{D}_A(\mathbf{x},\mathbf{y}) > \mathrm{D}_A(\mathbf{x},\mathbf{z})$. Hence, the objective is to find a matrix $\mathbf{A}$ with minimum rank that respects such constraints. Using a different form of constraints, positive and negative constraints (or similarity and dissimilarity information respectively), [23] tries to find a matrix $\mathbf{A}$ that will keep similar points close to each other while keeping dissimilar points far from each other, i.e. using positive and negative similarity constraints. RCA [2], in a simpler setting, uses only positive constraints and tries to find a matrix $\mathbf{A}$ that keeps similar points close to each other.

3 Modifying the MVE metric

In previous work [1], we have introduced the Minimum Volume Ellipsoid Metric (MVEM) as well as the Minimum Volume Ellipsoid of Nearest Neighbors (MiniVenn) algorithm to learn the MVEM. The MVEM is a metric measure (or similarity measure as illustrated later) that is defined independently for each point (referred to as a query point) in a data set based on the information in small neighborhood around it. Hence, the MVEM is locally defined for each query point.

First, we briefly review the MiniVenn algorithm. Consider a data set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$ that is drawn from a probability distribution $p(\mathbf{x})$, and a query point $\mathbf{x}_q \in \mathbb{R}^d$ that is also assumed to be drawn from $p(.)$, where d is the dimensionality of the input space, and n is the number of samples. The first step in MiniVenn is to find the m nearest neighbors to $\mathbf{x}_q$ from the set $\mathcal{X}$ using the Euclidean distance. Under the concept of locality [3], it is assumed that in a small neighborhood $\mathcal{N}_{\mathbf{x}_q}$ around $\mathbf{x}_q$ (ϵ–ball or m nearest neighbors), points will tend to be similar and share some common properties. Such similarities can be characterized by means of the neighborhood's covariance matrix (or local covariance) $\mathbf{S}_q$, and the induced Mahalanobis distance can measure the similarity between $\mathbf{x}_q$ and its neighbors while taking correlations and variances into consideration. In other words, the distance between $\mathbf{x}_q$ and any other point $\mathbf{x}_i \in \mathcal{X}$ will be based on the information in $\mathbf{x}_q$ and $\mathcal{N}_{\mathbf{x}_q}$ that is encoded in $\mathbf{S}_q$. However, due to the curse of dimensionality effect, the high nonlinearity of real life data and noise, estimating such a local covariance matrix using a Maximum Likelihood estimator (MLE) is hard and not reliable [7]. Therefore, instead of using a MLE, MiniVenn uses a robust estimator, the Minimum Volume Covering Ellipsoid (MVCE) estimator [16], to compute an accurate and a robust estimate for $\mathbf{S}_q$. Finally, the estimated $\mathbf{S}_q$ is used to define a more accurate Mahalanobis distance to measure similarity between $\mathbf{x}_q$ and any other point in $\mathcal{X}$.

3.1 Limitation of the MiniVenn algorithm

MiniVenn, and consequently the MVEM, suffered from three drawbacks. The first drawback arises from using the Euclidean metric in a high dimensional space to find the nearest neighbors of $\mathbf{x}_q$. As mentioned in the introduction, due to the curse of dimensionality the notion of similarity in a small neighborhood around $\mathbf{x}_q$ will be inaccurate. Second, in order to compute the robust estimate using the MVCE estimator, MiniVenn uses a slow convex optimization package [11] that hinders the usage of the MVEM in practical situations that require fast query–based operations. Deploying the MVEM in a query–based learning context requires a fast and efficient algorithm to compute the MVCE estimator. Finally, the literature on manifold learning algorithms assumes that the data actually lies on or near a lower dimensional nonlinear manifold that captures most of the data variability and is embedded in the high dimensional input space. MiniVenn in its current design state does not take this assumption into consideration; our objective is that MiniVenn should define the metric on the lower dimensional manifold on which $\mathbf{x}_q$ is lying.

Algorithm 1 Regularized Minimum Volume Ellipsoid of Nearest Neighbors : *Learns a local metric for query point $\mathbf{x}_q$ on the manifold on which $\mathbf{x}_q$ is lying.*

Require: $\mathcal{X}_{n\times d}$, $\mathbf{x}_q$, m, τ and ρ where $\mathcal{X}_{n\times d}$ is the training set with n d-dimensional samples, $\mathbf{x}_q$ is the query point, $m \geq d+1$ is a user input that controls the size of the neighborhood, $\tau > 0$ is the threshold to select the leading (tangent) directions with large eigenvalues along the manifold and $\rho \in [0,1]$ is the MVEM regularization parameter.

1: Find the set $\mathcal{N}_{\mathbf{x}_q}$ that has the m similar points to $\mathbf{x}_q$ using the similarity measure in Section 3.2.
2: Compute the robust estimate of $\mathbf{S}_q$ defined by the MVCE estimator for the set $\mathcal{N}_{\mathbf{x}_q}$ and centre $\mathbf{x}_q$ using Titterington algorithm [21].
3: Compute the eigen decomposition of $\mathbf{S}_q = \mathbf{V}\mathbf{L}\mathbf{V}^T$ where $\mathbf{V} = [V_1 \ \ldots \ V_d]$, $\mathbf{L} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ are the matrices of eigenvectors and eigenvalues respectively and $\lambda_1 > \lambda_2 > \cdots > \lambda_d$.
4: Select the d_0 leading eigenvalues such that $\lambda_{[1\,:\,d_0]} > \tau$ and form the matrix $\tilde{\mathbf{L}} = \mathrm{diag}(\rho, \ldots, \rho, \frac{1}{\lambda_{d_0+1}}, \ldots, \frac{1}{\lambda_d})$
5: **return** $\tilde{\mathbf{S}}_q^{-1} = \mathbf{V}\tilde{\mathbf{L}}\mathbf{V}^T$

3.2 The modified MiniVenn algorithm

The modified MiniVenn algorithm, shown in Algorithm (1), tries to overcome the above mentioned limitations by modifying the definition of local neighborhoods, by modifying the computation of the MVCE estimator, and finally by adding two extra steps for manifold detection. The algorithm proceeds as follows. In step 1, the algorithm defines a local neighborhood $\mathcal{N}_{\mathbf{x}_q}$ for $\mathbf{x}_q$ based on a similarity measure different than the Euclidean distance. In step 2, similar to the original MiniVenn, the algorithm computes the robust estimate of the covariance matrix $\mathbf{S}_q$ using the MVCE of the set $\mathcal{N}_{\mathbf{x}_q}$. The difference here is in the algorithm used to compute the MVCE of $\mathcal{N}_{\mathbf{x}_q}$. Steps 3 an 4 are the new steps in the algorithm and they are concerned with manifold detection, estimation of the local intrinsic dimensionality at $\mathbf{x}_q$, and MVEM regularization. In the following, each modification and addition will be explained in more detail.

Redefining local neighbourhoods : The definition of $\mathcal{N}_{\mathbf{x}_q}$ for the query point $\mathbf{x}_q$ in the original MiniVenn used the Euclidean distance as a similarity measure to find the m nearest neighbors to $\mathbf{x}_q$. Since our major objective is to find the most similar points to $\mathbf{x}_q$, one can define a different similarity measure between points and use it to define local neighborhoods. Indeed, selecting appropriate neighbourhoods is a key factor to the success of local learning algorithms [14, 20]. A flexible and easy to compute similarity measure is the dot product between two vectors. That is, let $s_i = \mathrm{Sim}(\mathbf{x}_q, \mathbf{x}_i) = \langle \mathbf{x}_q^{'}, \mathbf{x}_i^{'} \rangle = \mathbf{x}_q^{'T}\mathbf{x}_i^{'}$, be the similarity measure between $\mathbf{x}_q$ and $\mathbf{x}_i$, where $\mathbf{x}_q^{'} = \mathbf{x}_q/\|\mathbf{x}_q\|_2$, $\mathbf{x}_i^{'} = \mathbf{x}_i/\|\mathbf{x}_i\|_2$ and $\mathbf{x}_i \in \mathcal{X}$. This is equivalent to finding the m nearest neighbors for $\mathbf{x}_q^{'}$ after normalizing all the points to lie on the unit sphere in $\mathbb{R}^d$. This similarity measure can mitigate the curse of dimensionality effect and due to its flexibility, it can be easily extended to accommodate different similarity measures using the kernel trick.

Fast computation of the MVCE : Let $\mathcal{N}_{\mathbf{x}_q} = \{\mathbf{x}_j \mid 1 \leq j \leq m,\ \mathbf{x}_j \in \mathcal{X}\}$ be the set of similar neighbors to the point $\mathbf{x}_q$ using the above defined similarity measure. The MVCE of $\mathcal{N}_{\mathbf{x}_q}$ with centre $\mathbf{x}_q$ is denoted by $\mathcal{E}$ and is parameterized by a symmetric and positive definite matrix $\mathbf{S}_q \in \mathbb{R}^{d\times d}$ as follows [4]:

$$\mathcal{E} \quad = \quad \{\mathbf{x}_j \mid \|\mathbf{S}_q^{-\frac{1}{2}}\mathbf{x}_j - \mathbf{b}\|_2^2 \leq 1,\ \forall j\} \tag{1}$$

where $\mathbf{b} = \mathbf{S}_q^{-\frac{1}{2}}\mathbf{x}_q$. Since $V(\mathcal{E}) \propto \det(\mathbf{S}_q^{-1})$, where $V(\mathcal{E})$ is the ellipsoid's volume, minimizing this volume can be formulated as follows:

$$\min_{\mathbf{S}_q} \ \log\det \mathbf{S}_q, \ \text{ s.t. } \ \|\mathbf{S}_q^{-\frac{1}{2}}\mathbf{x}_j - \mathbf{b}\|_2^2 \leq 1,\ \forall j \tag{2}$$

The objective and the constraints in (2) are convex in $\mathbf{S}_q$, therefore this optimization problem has a unique global optimal solution. However, as in [1], directly solving this optimization problem using standard convex optimization libraries such as CVX [11] showed to be computationally expensive and not efficient for practical situations. Alternatively, the dual of this optimization problem, thanks to Titterington [21], is easier to optimize and has a very fast and efficient algorithm for its computation (see [21] for algorithm details) :

$$\begin{aligned} \max_{\mathbf{S}_q, \Phi} \quad & \log\det(\mathbf{S}_q) \\ \text{s.t.} \quad & \mathbf{S}_q = \sum_{j=1}^{m} \phi_j (\mathbf{x}_j - \mathbf{x}_q)(\mathbf{x}_j - \mathbf{x}_q)^T + \gamma\mathbf{I} \\ & \mathbf{\Phi} \in \mathbb{R}^m, \ \ \mathbf{\Phi} \geq 0, \ \ \mathbf{\Phi}^T\mathbf{e} = 1 \end{aligned} \tag{3}$$

where Φ is the vector of dual variables ϕ_j, $\gamma \geq 0$ and $\gamma\mathbf{I}$ is an extra constraint that guarantees a minimal diameter of the ellipsoid in all directions to prevent the ellipsoid from collapsing to zero volume especially in large dimensional spaces [6].

Manifold detection : The literature on manifold learning algorithms assumes that despite of the high dimensionality of the data in the input space, most of the data variability can be captured by far fewer dimensions known as the intrinsic dimensionality of the data. Accordingly, it is assumed that the data lies on or near (due to noise) a lower dimensional nonlinear manifold that is embedded in the high dimensional input space. However, due to *data complexity issues* the data might not actually lie on a single nonlinear manifold, but rather on or near several disconnected nonlinear manifolds [13]. It is this last observation that motivates

our objective to let MiniVenn define the metric on the lower dimensional manifold on which $\mathbf{x}_q$ is lying. To detect this manifold, its intrinsic dimensionality and the distance measure in that neighborhood, MiniVenn performs an eigen decomposition and a regularization step for the robust estimate $\mathbf{S}_q$. The benefit of the eigen decomposition is twofold: 1) It can estimate the intrinsic dimensionality of the data using Fukunaga's algorithm [8] by means of the number of dominating eigenvalues of $\mathbf{S}_q$ (which is the role of parameter τ), and 2) The orthogonal eigenvectors of $\mathbf{S}_q$ decide which vectors are tangent or normal to the underlying manifold. That is, the eigenvector associated with the smallest eigenvalue (or the component with lowest variance in $\mathcal{N}_{\mathbf{x}_q}$) is normal to the manifold, while the eigenvector associated with the largest eigenvalue (or the component with highest variance in $\mathcal{N}_{\mathbf{x}_q}$) is tangent to the manifold and the latter is the main direction of interest since it is the direction that goes along the manifold and contributes the most to the dissimilarity measure in the neighborhood of $\mathbf{x}_q$. Note that in a d–dimensional space and for a d_0–dimensional manifold with $d_0 \ll d$, there will be approximately d_0 tangent vectors associated with the largest eigenvalues.

The Mahalanobis distance, however, measures the similarity using $\mathbf{S}_q^{-1}$, i.e. by taking the inverse of the eigenvalues, thus assigning small weights to high variance components (tangent eigenvectors) and large weights to low variance components (normal eigenvectors). It is at this point that the regularization parameter ρ is needed to emphasize the contribution of the main tangent vectors over the contribution of normal and less significant tangent vectors. More specifically, ρ influences the notion of similarity of the MVEM, however this is task dependent since it can tune the MVEM according to the objective of the task under consideration.

3.3 Generalization of the MVEM

Generalization of the MVEM is controlled by the MiniVenn's four parameters: m, τ, ρ and implicitly γ to compute the optimization in (3). While m and τ reflect the topological properties of the data, ρ influences the notion of similarity of the obtained metric. Using [8], τ can be fixed for a data set since it is a threshold on the normalized eigenvalues. Similarly, γ can be fixed for each data set separately although it was fixed to either 0 or 0.1 in all our experiments. More attention however, is required to select m and ρ. A large value of m will over smooth the main tangent directions of the patch on which $\mathbf{x}_q$ is lying, while a very small value will lead to crude and rather fragile estimates of these directions. An intuitive approach is to select m and ρ via an optimization procedure. This can be achieved by linking the two parameters to an objective function that can to be optimized. The optimal objective function in this case would be the objective function of the task under consideration, and implicitly, results in the metric (or the MVEM) being tuned to maximize or minimize this objective function. For instance, in the case of our experiments on query-based learning, m and ρ were optimized by a grid search to minimize the expected zero–one loss $E[L(Y, f(X))] = E[1-\delta(Y, f(X))]$ (or miss-classification rate) on the available training set, where Y is the true label of the input X, $f(X)$ is the decision obtained from the classifier, and the $\delta(.,.)$ is the Kronecker delta function. Accordingly, since there is a training phase to optimize m and ρ directly on the task's objective function, the MVEM is expected to generalize well on unseen data sets.

It is also worth noting that when MiniVenn forms a local neighborhood for the query point, it does not depend on labels or side-information [23] from the data, but rather on the similarity measure and the parameter m. This is unlike other metric learning algorithms that rely on the availability of *a priori* information in the form of fully/partially labeled data or side–information. The importance and contribution of any *a priori* knowledge only appears when optimizing m and ρ as mentioned earlier. Therefore, MiniVenn can be considered an unsupervised metric learning algorithm in that regard.

Table 1. The fifteen UCI [15] data sets used in our experiments with the number of classes, size and dimensionality.

ID	Dataset	Classes	Size	Dim.
bal	Balance	3	625	4
bup	Bupa	2	345	6
gla	Glass	7	214	9
hou	HouseVotes	2	341	16
ion	Ionosphere	2	350	33
iri	Iris	3	150	4
lym	Lymphography	4	148	18
pag	Pageblocks	5	5473	10
new	NewThyroid	3	215	5
pim	Pima	2	768	8
seg	Segment	7	2086	18
tic	TicTacToe	2	958	9
wdb	WDBC	2	569	30
win	Wine	3	168	13
yea	Yeast	10	1484	6

4 Experimental results

In order to assess the validity and generalization of the modified MiniVenn algorithm, we have conducted extensive experiments in the context of classification on a large variety of data from standard benchmark data sets. In our experiments, fifteen data sets were used from the UCI Machine Learning Repository [15], shown in Table 1, with various sizes, number of classes and dimensionality. Since there

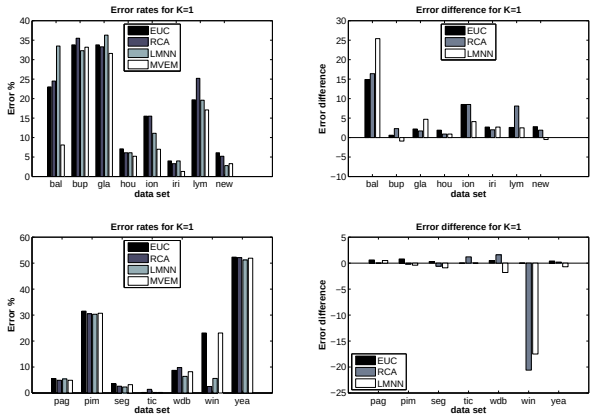

Figure 1. *Left.* Comparing error rates for k–NN classification ($k = 1$) using Four different metrics: EUC, RCA [2], LMNN [22] and MVEM on the 15 UCI data sets. *Right.* Difference in error for EUC, RCA, LMNN against the MVEM. Positive value implies the MVEM is better and a negative value implies the other metric is better. Ex: Error(EUC) − Error(MVEM).

is no explicit training and test sets for these data sets, 10 Folds Double Cross Validation (FDCV) were used to report the errors on all the data sets. For query–based learning, a k–Nearest Neighbour (k–NN) classifier was used with three different values for $k = (1, 3, 5)$. Only raw data was used in the experiments without any kind of preprocessing. For comparisons, the k–NN classifier using the MVEM was compared with k–NN classifiers using three different metrics[1]: Euclidean (EUC), large margin nearest neighbours (LMNN) [22] and relevant component analysis (RCA) [2]. Since RCA depends on the availability and the amount of side–information (true labels in the context of classification), all the true labels for a given training set were provided to RCA in order to peel off any doubts about its performance.

Results analysis: Figures 1, 2 and 3 show the error rate and the difference in error for the three classifiers, $k = 1$, $k = 3$ and $k = 5$ respectively, using the four metrics on all data sets. Standard errors are not shown due to space limitations. Each figure is formed of four plots. The first row of each figure shows the first eight data sets, while the second row shows the remaining 7 data sets. The second column in each figure shows the difference in error between any of the three metrics (EUC, LMNN, RCA) and the MVEM. This should give a tangible feeling on the performance of the MVEM. It can be seen clearly that in most cases, the MVEM is placed first with a large margin in error difference, or placed second with a very small margin in error difference against the competing metric. The MVEM consistently scores better than the Euclidean metric and very competitive with a more dedicated algorithm like LMNN which was specifically designed to learn a metric that minimizes the error of k–NN classification (i.e. discriminative training). Similar behaviour is observed when comparing between MVEM and RCA (with 100% available true positive constraints). Although, as shown in Figure 4, the MVEM is slightly better in its overall performance, statistical significance tests showed that on average, the MVEM is not significantly different than the more dedicated algorithms LMNN and RCA. This is a very interesting result since MiniVenn had less *a priori* information during training and yet it showed similar performance. These results motivate us to extend the proposed algorithm and metric to the domain of clustering and unsupervised learning with complete absence of side–information and labels.

[1]The source code for LMNN and RCA was downloaded from the author's website.

5 Conclusion

We have introduced an algorithm for learning an adaptive metric for query–based operations. The algorithm combines ideas from the minimum volume ellipsoid metric and of manifold learning algorithms to define a metric on the lower dimensional manifold of the query point. In the context of classification and using a k–NN classifier, the metric showed very promising results in that regard and is competitive with other metric learning algorithms in the literature.

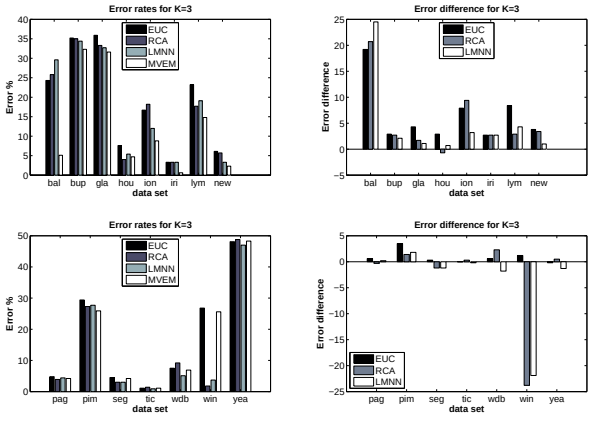

Figure 2. Error rates and error differences for k–NN classification using $k = 3$. Please see caption of Figure 1 for explanation.

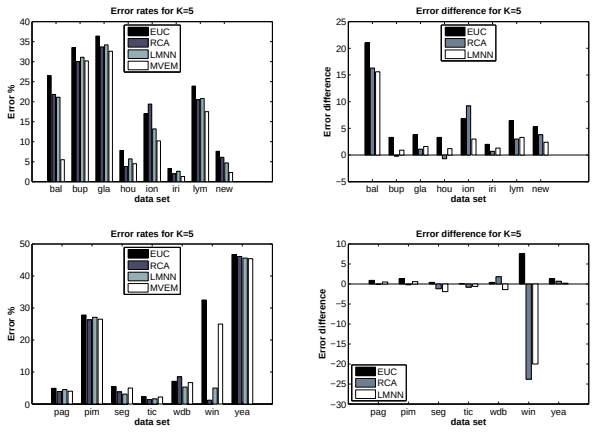

Figure 3. Error rates and error differences for k–NN classification using $k = 5$. Please see caption of Figure 1 for explanation.

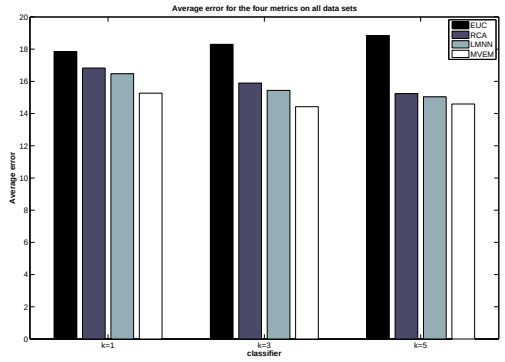

Figure 4. Average error for the three classifiers using the four metrics on all data sets.

References

[1] K. Abou-Moustafa and F. Ferrie. The minimum volume ellipsoid metric. In *LNCS 4713, 29th Symposium of the German Association of Pattern Recognition, Heidelberg*, pages 335–344. Springer, 2007.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[3] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

[4] S. Boyd and L. Vandenberghe, editors. *Convex Optimization*. Cambridge Univ. Press, 2004.

[5] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k–means clustering. In *Proceedings of the 24th ICML, Corvallis, OR*. 2007.

[6] A. Dolia, T. De Bie, C. Harris, J. Shawe-Taylor, and D. Titterington. The minimum volume covering ellipsoid estimation in kernel-defined feature spaces. In *Proceedings of the 17th ECML, Berlin, September*. Springer, 2006.

[7] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[8] K. Fukunaga and R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. on Computers*, 20(2):176–183, 1971.

[9] A. Globerson, S. Roweis, G. Hinton, and R. Salakhutdinov. Metric learning by collapsing classes. In NIPS *18*, pages 451–458. MIT Press, 2006.

[10] J. Goldberg and S. Roweis. Neighbourhood component analysis. In NIPS *17*, pages 513–520. MIT Press, 2005.

[11] M. Grant, S. Boyd, and Y. Yinyu. Matlab software for disciplined convex programming, 2005. `http://www.stanford.edu/~boyd/cvx`.

[12] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbour classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(6):607–615, 1996.

[13] G. Lebanon. Metric learning for text documents. *IEEE Trans. PAMI*, 28(4):497–508, 2006.

[14] T. Lin and H. Zha. Riemannian manifold learning. *IEEE. Trans. PAMI*, 30(5):796–809, 2008.

[15] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[16] P. Rousseeuw. Multivariate estimation with high breakdown point. In *Proc. of the Fourth Pannonian Symposium on Mathematical Statistics*, volume 3, pages 283–297, 1983.

[17] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding (lle). *Science*, 290(5500):2323–2326, 2000.

[18] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In NIPS *16*. MIT Press, 2004.

[19] R. Short and K. Fukunaga. The optimal distance measure for nearest neighbour classification. *IEEE Trans. on Information Theory*, 27(5):622–627, 1981.

[20] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, November 2000.

[21] D. Titterington. Estimation of correleation coefficients by ellipsoidal trimming. *Journal of Royal Statistical Society*, 27(3):227–234, 1978.

[22] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In NIPS *18*, pages 1473–1480. MIT Press, 2006.

[23] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In NIPS *15*, pages 505–512. MIT Press, 2003.

[24] L. Yang. Distance metric learning: A comprehensive review. Technical report, Dept. of Computer Science and Engineering, Michigan State University, 2006.