Adv Data Anal Classif
DOI 10.1007/s11634-017-0286-x

REGULAR ARTICLE

# Local generalized quadratic distance metrics: application to the *k*-nearest neighbors classifier

**Karim Abou-Moustafa**[1] · **Frank P. Ferrie**[2]

Received: 21 October 2016 / Revised: 10 April 2017 / Accepted: 17 April 2017


**Abstract** Finding the set of nearest neighbors for a query point of interest appears in a variety of algorithms for machine learning and pattern recognition. Examples include *k* nearest neighbor classification, information retrieval, case-based reasoning, manifold learning, and nonlinear dimensionality reduction. In this work, we propose a new approach for determining a distance metric from the data for finding such neighboring points. For a query point of interest, our approach learns a generalized quadratic distance (GQD) metric based on the statistical properties in a "*small*" neighborhood for the point of interest. The locally learned GQD metric captures information such as the density, curvature, and the intrinsic dimensionality for the points falling in this particular neighborhood. Unfortunately, learning the GQD parameters under such a local learning mechanism is a challenging problem with a high computational overhead. To address these challenges, we estimate the GQD parameters using the minimum volume covering ellipsoid (MVCE) for a set of points. The advantage of the MVCE is two-fold. First, the MVCE together with the local learning approach approximate the functionality of a well known robust estimator for covariance matrices. Second, computing the MVCE is a convex optimization problem which, in addition to having a unique global solution, can be efficiently solved using a first order optimization algorithm. We validate our metric learning approach on a large variety of datasets

✉ Karim Abou-Moustafa
aboumous@ualberta.ca

Frank P. Ferrie
ferrie@cim.mcgill.ca

[1] Department of Computing Science, ATH 3-55, University of Alberta, Edmonton, AB T6G 2E8, Canada

[2] Department of Electrical and Computer Engineering, McGill University, McConnell Engineering Building, Room 441, 3480 University Street, Montreal, QC H3A 2E9, Canada

Published online: 25 April 2017

and show that the proposed metric has promising results when compared with five algorithms from the literature for supervised metric learning.

**Keywords** Query-based operations · $k$ Nearest neighbors · Distance metric learning · Minimum volume covering ellipsoid · Minimum volume ellipsoid estimator

**Mathematics Subject Classification** 62H30 · 68T10

## 1 Introduction

There are various algorithms in machine learning, data mining, and signal processing, that implicitly or explicitly require finding a small set of nearest neighbors (or similar points) for a query point of interest $\mathbf{x}_q \in \mathbb{R}^d$ from a finite set of points with high-dimensionality $d$. For instance, finding such a set of nearest neighbors (NNs) appears in $k$-NN classifiers, information and multimedia retrieval, case-based reasoning, manifold learning, spectral clustering, and nonlinear dimensionality reduction algorithms (Cover and Hart 1967; Fukunaga 1972; Short and Fukunaga 1981; Macleod et al. 1987; Hu et al. 2011; Belkin and Niyogi 2003; Coifman and Lafon 2006; Dornaika and El Traboulsi 2015). The input data points for these algorithm can be high-dimensional feature vectors describing text documents, video clips (or frames), speech frames, image patches, proteins or peptides, etc. Such algorithms rely on a notion of distance to measure the similarity between the high-dimensional points. These distances or similarity measures are usually imposed on the data under the assumption that the data is embedded in a space that is already endowed with a distance metric. For instance, the $k$-NN classifier and neural networks consider the embedding space to be $\mathbb{R}^d$, while support vector machines (SVMs) and kernel methods consider the embedding space to be a reproducing kernel Hilbert space (RKHS) that is obtained through some kernel operations.

In this paper, we depart from this assumption and propose to learn a data dependent distance metric for finding the NNs of a query point $\mathbf{x}_q$, without being overly dependent on class labels or side information in the dataset. While the Euclidean distance $\|\mathbf{x}-\mathbf{y}\|_2$, Minkowski type distance, and other off-the-shelf distance (and similarity) measures, are the conventional measures used for this type of distance operation, there are a few reasons to doubt the appropriateness of these distances when dealing with high-dimensional real-world data. One reason is the curse of dimensionality; that is, the structure of high-dimensional spaces defies our visualization for three dimensional geometry since such spaces are extremely sparse and the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions (Ding and Li 2007). A second reason stems from the complex nature of real-world data: (*i*) highly structured and nonlinear (e.g. images, text documents, proteins, etc.); (*ii*) measured from various sources at different scales and with various degrees of variability and correlation; and (*iii*) prone to various sources of noise that may largely deviate measurements and raise outliers in the data. The third reason is that most off-the-shelf distance and similarity measures are unaware of the data's underlying structure and distribution, and assume implicitly a constant density over the entire

input space—a situation that is hardly attained in real-world data. In real-world finite datasets, regions with low probability distribution are poorly sampled and hence poorly represented in the data; a phenomenon known as *the uneven sample distribution* in the input space.

Distance metric learning has emerged as an important research topic to address some of the aforementioned limitations (Yang 2006; Kulis 2013). Although the idea is not new for supervised learning—in particular for the $k$-NN classifier (Short and Fukunaga 1981)—recent work on metric learning tries to find suitable embeddings for the data that can reveal more about its structure. Examples include manifold learning algorithms, spectral embedding methods, algorithms for direct metric learning in the input space, and their various extensions (Shepard 1962; Kruskal 1964; Tenenbaum et al. 2000; Roweis and Saul 2000; Belkin and Niyogi 2003; Xing et al. 2003; Schultz and Joachims 2004). Unfortunately for our context, the existing metric learning approaches are still inadequate for two reasons. First, metric learning algorithms are either supervised or semi-supervised and hence they do not meet our objective that tries to be less dependent on class labels and side information in the data. Second, metric learning algorithms do not take into consideration the varying sample density in the input space. To see this, note that these algorithms usually learn a generalized quadratic distance (GQD) metric: $\|\mathbf{x}-\mathbf{y}\|_{\mathbf{A}} = \sqrt{(\mathbf{x}-\mathbf{y})^\top \mathbf{A}(\mathbf{x}-\mathbf{y})}$, where $\mathbf{A}$ is a symmetric positive definite (PD) matrix. We observe here that $\mathbf{A}$ is defined over the entire input space and does not vary according to the data distribution.

In this paper we propose an unsupervised approach that defines adaptive distance metric functions for finding accurate nearest neighbors of a query point of interest $\mathbf{x}_q$. Our approach assumes that the data lie on a low dimensional manifold that varies smoothly around $\mathbf{x}_q$, and hence we expect that the points falling in a *small neighborhood* of $\mathbf{x}_q$ will share similar properties. This suggests that in order to find better NNs for $\mathbf{x}_q$, one can rely on the local properties for the neighborhood of $\mathbf{x}_q$ rather than the global properties of the entire dataset. To this end, we propose to define a GQD metric for each query point of interest $\mathbf{x}_q$ based only on the structure information in the neighborhood of $\mathbf{x}_q$, such as density, correlations, curvature, and intrinsic dimensionality. Note that this is different from learning a global GQD metric parameterized by $\mathbf{A}$ under some constraints from class labels and/or side–information.

In our approach, each point $\mathbf{x}_i$, for $i = 1, \ldots, n$, defines its own symmetric PD matrix $\mathbf{A}_i$ based on the $m$ nearest neighbors falling in the neighborhood of $\mathbf{x}_i$. This neighborhood is denoted by $\mathcal{N}(\mathbf{x}_i)$. Since in most applications $m \ll d$, computing an estimate for the covariance matrix $\mathbf{A}_i$ entails two difficulties: (*i*) an appropriate estimator for $\mathbf{A}_i$, and (*ii*) an efficient algorithm for computing this estimate. For estimation, it is desirable to obtain a reliable estimate for $\mathbf{A}_i$ given the few $m$ samples in $\mathcal{N}(\mathbf{x}_i)$. For computation, a fast and efficient algorithm is required to compute this estimate since it will be used in a query-based setting with large datasets. To this end, we propose to estimate $\mathbf{A}_i$ using Titterington's first order algorithm for computing the minimum volume covering ellipsoid for a set of points (Titterington 1978). Finally, we validate our metric learning approach on a variety of datasets, and compare it to well known algorithms for supervised distance metric learning.

The organization of this paper is as follows. Section 2 briefly overviews the literature on supervised local distance metric learning. Section 3 introduces our main approach

for defining distance metrics based on the set of nearest neighbors falling in the local neighborhood for the point of interest $\mathbf{x}_q$. Section 4 considers the internal details for our proposed method for distance metric learning. In particular, Sect. 4 addresses the motivation for choosing the minimum volume covering ellipsoid (MVCE) as an estimator for the matrix $\mathbf{A}_i$, the MVCE formulation, and a first order efficient algorithm for computing the MVCE estimate for $\mathbf{A}_i$ (Titterington 1978). Section 5 presents our experimental results, and closing remarks with extensions to other research work are given in Sect. 6.

**Notations and setup:** Lower case letters $x, m, i$ denote scalars and indexes. Bold small letters $\mathbf{x}, \mathbf{y}$ are vectors. Bold capital letters $\mathbf{A}, \mathbf{B}$ are matrices. $\mathbf{I}$ is the identity matrix of suitable dimensions. Distributions are denoted in script: $\mathscr{P}, \mathscr{G}$. Calligraphic and double bold capital letters $\mathcal{X}$, $\mathcal{Y}$, $\mathbb{X}$, $\mathbb{Y}$ denote sets and/or special spaces. Symmetric positive definite (PD) and positive semi–definite (PSD) matrices are denoted by $\mathbf{A} \succ 0$ and $\mathbf{A} \succeq 0$, respectively. $\mathrm{tr}(\cdot)$ is the matrix trace and $\det(\cdot)$ is the matrix determinant. The space of $d \times d$ symmetric PD matrices is denoted by $\mathbb{S}_{++}^{d \times d}$. We assume that $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a set of $n$ independent and identically distributed (i.i.d) points sampled from the unknown distribution $\mathscr{P}_{\mathbf{x}}$; this is denoted by $\mathcal{X} \sim \mathscr{P}_{\mathbf{x}}$.

## 2 Literature review

The literature on metric learning can be categorized according to four dimensions (Yang 2006): (*i*) supervised, unsupervised or semi–supervised, (*ii*) local, or global, (*iii*) linear or nonlinear, and (*iv*) using embedding or not. The supervised approach can be further categorized based on the type of labels associated with each data point, which can be in the form of class labels, pairwise distances, or pairwise constraints. The latter constraints are also known as equivalent/inequivalent constraints, $+ve/-ve$ constraints, or side information. If the class membership of the data is partially known (partial labeling), then the algorithm that identifies the distance metric from the data is considered to be a semi–supervised learning algorithm. For our context, we give a brief literature review on supervised local distance metric learning.

The earliest work on supervised local distance metric learning dates back to Short and Fukunaga (1981) with an attempt to minimize the difference between the finite sample nearest neighbor (NN) classification error and the asymptotic NN error (or the twice Bayes error bound). Assuming a smooth posterior and smooth conditional densities around points, the distance between a query point and its neighbors is weighted by the gradient of the posterior probability with respect to the query point, given the class labels of the nearest neighbors. This should give a larger weight to features that are relevant to the classification task (a.k.a. local feature relevance). Friedman (1994) reuses this idea of local feature relevance combined with recursive partitioning of the space, in a similar spirit to decision trees, to achieve a flexible nearest neighbor metric that is adapted to each point and its neighborhood. Hastie and Tibshirani (1996) generalize the work of Short and Fukunaga (1981) by defining local linear discriminant analysis (LDA) for each query point in a neighborhood around it. Their neighborhoods are ellipsoids stretched along decision boundaries between classes. Domeniconi and Gunopulos (2002) use SVMs to compute locally flexible metrics where the maxi-

mum margin of SVMs decides the most discriminating features (or directions) over the query point's neighborhood, and hence provides weights for each feature. In a similar vein, Domeniconi et al. (2002) and Peng et al. (2004) replace SVMs by the Chi–squared distance analysis and quasi-conformal kernels, respectively, to achieve the same objective. By changing class labels to partial side–information, Chang and Yeung (2007) learn a metric through local linear transformations of neighborhoods. The metric is learned independently for each point and its neighborhood through a regularized moving least squares framework with closed form solutions.

Our proposed distance metric learning algorithm, although developed independently in (Abou-Moustafa and Ferrie 2007), is in the same spirit of (Chang and Yeung 2007) since our algorithm defines a distance metric for each point from the local density information surrounding it. However, our work is also different from (Chang and Yeung 2007) in two aspects. First, we consider an unsupervised setting where no class labels nor side information are available for the training data, while Chang and Yeung (2007) assume a semi–supervised setting where such information is available for training. Second, for a query point $\mathbf{x}_q$, our approach uses a first order optimization algorithm that runs only on the samples falling the in the neighborhood of $\mathbf{x}_q$. By contrast, Chang and Yeung (2007) use a second order optimization algorithm that requires a pseudo-inverse for a $d \times d$ matrix, and to define the distance metric for each data point, the optimization algorithm runs on the remaining $n - 1$ data points.

## 3 The minimum volume ellipsoid metric

The GQD metric: $\|\mathbf{x} - \mathbf{y}\|_{\mathbf{A}} = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{A}(\mathbf{x} - \mathbf{y})}$, is a generalization of the Mahalanobis distance which is known as an outlier detector in the robust statistics literature. The Mahalanobis distance exposes outliers by assigning them large distances. This, however, depends on an accurate estimate for the covariance matrix $\mathbf{A}$. To obtain such an estimate, our approach relies on approximating a robust estimator for the covariance matrix $\mathbf{A}$ known as the Minimum Volume Ellipsoid (MVE) estimator (Rousseeuw and Leroy 1987). Our approximation has some desirable properties such as the intuitive geometric meaning, and its formulation as a convex optimization problem with a global unique solution. The MVE estimator and our proposed approximation will be discussed in the following section. The resulting distance metric from the MVE estimator will be called the MVE Metric, or MVEM.

The basic idea underlying the MVEM is that the distance metric is determined independently for each point, should it be a training point, a test point, or a query point, labeled or unlabeled. For $\mathcal{X}^n \sim \mathscr{P}_{\mathbf{x}}$, and for $1 \leq i \leq n$, each point $\mathbf{x}_i \in \mathcal{X}$ defines its own GQD metric: $\|\mathbf{x}_i - \mathbf{y}\|_{\mathbf{A}_i}$, where $\mathbf{y} \in \mathbb{R}^d$. Since the coordinates of $\mathbf{x}_i$ define its location in the input space, the MVEM by definition adapts to this particular location in the input space through the estimate of the covariance matrix $\mathbf{A}_i$. Each matrix $\mathbf{A}_i$ is estimated from the samples falling in the neighborhood of $\mathbf{x}_i$ denoted by $\mathcal{N}(\mathbf{x}_i)$. The covariance matrix $\mathbf{A}_i$ captures the local density information in $\mathcal{N}(\mathbf{x}_i)$ through the variances of the variables and their correlations among each other. Further, it captures the local curvature of the underling manifold in $\mathcal{N}(\mathbf{x}_i)$ and its intrinsic dimensionality; the leading eigenvectors of $\mathbf{A}_i$ are tangent to the underlying data manifold in $\mathcal{N}(\mathbf{x}_i)$,

**Algorithm 1** Learn MVEM: Learns a GQD metric for point $\mathbf{x}_q$ using an approximation to the MVE covariance matrix estimator.

**Require:** The query point of interest $\mathbf{x}_q$; a dataset with $n$ $d$-dimensional points $\mathcal{X}$; and the size of the neighborhood $m$.

1: Define the neighborhood $\mathcal{N}(\mathbf{x}_q)$ by finding the $m$ NN points to $\mathbf{x}_q$ from $\mathcal{X}$ using any of the following:
   - A suitable $p$–norm $||\mathbf{x}_i - \mathbf{x}_q||_p$,
   - Kernel function $K(\mathbf{x}_i, \mathbf{x}_q)$, or
   - A similarity measure based on *a priori* domain knowledge.

2: Compute the estimate $\widehat{\mathbf{A}}(\mathbf{x}_q) \in \mathbb{S}_{++}^{d\times d}$ from the points in $\mathcal{N}(\mathbf{x}_q)$ using a modified version of Titterington's MVCE algorithm (for a fixed mean vector) shown in Algorithm 2. This algorithm will be discussed in the following section. Optionally, compute an eigen decomposition for $\widehat{\mathbf{A}}(\mathbf{x}_q)$ to obtain a low rank estimate $\widehat{\mathbf{A}}_{\text{LR}}(\mathbf{x}_q) \in \mathbb{R}^{d\times r}$, where $r \ll d$.

3: **return** $\widehat{\mathbf{A}}(\mathbf{x}_q)$ or $\widehat{\mathbf{A}}_{\text{LR}}(\mathbf{x}_q)$.

and the number of leading eigenvectors is an estimate for the intrinsic dimensionality of points in $\mathcal{N}(\mathbf{x}_i)$ (Fukunaga 1972). The MVEM does not depend on class labels nor side–information. However, if such information is available, it can be incorporated to define the metric. For instance, one can define neighborhoods using points with the same class label, or using equivalence links.

Algorithm 1 depicts the steps for learning the MVEM for $\mathbf{x}_q$. The algorithm starts by defining the neighborhood $\mathcal{N}(\mathbf{x}_q)$, and in Step 2 it estimates the symmetric PD matrix $\mathbf{A}(\mathbf{x}_q)$ using a modified version of Titterington's Minimum Volume Covering Ellipsoid (MVCE) algorithm (with a fixed mean vector) from the points in $\mathcal{N}(\mathbf{x}_q)$ only. This will be explained in the following section. Step 2 in Algorithm 1 includes an optional step that can provide significant memory savings, as well as speedup in computations. If the dataset is assumed to lie on a low dimensional manifold, then the estimate $\widehat{\mathbf{A}}(\mathbf{x}_q)$ will be low rank. Considering an eigen decomposition for $\widehat{\mathbf{A}}(\mathbf{x}_q)$, its $r \ll d$ leading eigenvectors–eigenvalues yield a low rank approximation to $\widehat{\mathbf{A}}(\mathbf{x}_q)$, denoted by $\widehat{\mathbf{A}}_{\text{LR}}(\mathbf{x}_q)$. Any matrix–vector or matrix–matrix operations with $\widehat{\mathbf{A}}_{\text{LR}}(\mathbf{x}_q)$ can be done in $O(rd)$ and $O(rd^2)$ instead of $O(d^2)$ and $O(d^3)$, respectively.

### 3.1 The neighborhood size and the initial distance metric

Our approach assumes that the input space is locally smooth and hence it can be considered a smooth differentiable manifold that is locally Euclidean. Under this assumption, Euclidean geometry only holds in a neighborhood around each point in the dataset. Here the neighborhood size is defined in terms of the number of NNs for $\mathbf{x}_q$ and denoted by $m$. In principle, the neighborhood size should slowly grow until it circumscribes the region where Euclidean geometry holds. Beyond this size, the local Euclidean assumption will break due to the manifold curvature. If $m$ is too small, the estimate for the local Euclidean subspace will be inaccurate, while if $m$ is too large, the local linear structure will be smoothed out by the influence of distant points.

Note that defining $\mathcal{N}(\mathbf{x}_q)$ using any similarity measure or *a priori* information is considered to be bootstrapping the MVEM since the NNs depend on the choice of the initial similarity measure used. In our experiments in Sect. 5, using the Euclidean dis-

tance to infer the GQD, although this might seem to be a naïve choice, it showed to be a reasonable strategy based upon the performance on various datasets. Alternatively, in high dimensions, $\mathcal{N}(\mathbf{x}_q)$ can be defined using fractional distances to leverage the curse of dimensionality effect (Aggarwal et al. 2001; François et al. 2007). In practice, since most learning algorithms entail minimizing an objective function of the data, optimizing $m$ (and optionally $r$ in Algorithm 1) can be done using a grid search evaluated by the loss values. The objective of optimizing $m$ in this way is to obtain a consistent neighborhood size for the finite dataset that is aligned with the learning task under consideration (through its objective function). Note that different objective functions can result in different optimal values for $m$. For instance, as shown in the Experimental Results section, optimizing $m$ for the $k$-NN classifier is based on minimizing the error rate for the $k$-NN classifier.

The presented local distance metric approach can encourage the reader to question the relation between two covariance matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ for two nearby points $\mathbf{x}_i$ and $\mathbf{x}_j$ in terms of shared information, overlap, and distance. In this setting, the ellipsoids defined by $\mathbf{A}_i$ and $\mathbf{A}_j$ might overlap with each other, and might be aligned in the same orientation as well. The amount of overlap and the degree of alignment will depend on the closeness of $\mathbf{x}_i$ and $\mathbf{x}_j$, the neighborhood size, and the local distribution of points in each neighborhood. In particular, the leading eigenvectors for $\mathbf{A}_i$ and $\mathbf{A}_j$ will tend to be aligned to each other depending on these factors. The amount of information shared between $\mathbf{A}_i$ and $\mathbf{A}_j$ can be quantified using suitable metrics for covariance matrices (Abou-Moustafa and Ferrie 2012). While in this work we do not address this aspect since our concern here is the query-based setting, this aspect is thoroughly discussed in an extension of this research work (Abou-Moustafa et al. 2013).

In the following section, we consider the details for estimating $\mathbf{A}(\mathbf{x}_q)$ from the points in $\mathcal{N}(\mathbf{x}_q)$ using the minimum volume covering ellipsoid algorithm.

## 4 A reliable estimate for covariance matrices

The Minimum Volume Ellipsoid (MVE) estimator is a well known robust estimator for location (mean) and scatter (covariance matrix) (Rousseeuw and Leroy 1987). It is the generalization of the least median of squares (LMS) estimator in high dimensions with the extra property of being equivariant to translation, scaling, orthogonal projection and affine transformations.

The MVE estimator assumes that points are not necessarily normally distributed and may contain a proportion $\alpha = k/n$ of outliers, where $0 < \alpha \leq 0.5$, $n$ is the sample size, and $k < n$ is the number of outliers in the sample. The MVE estimator finds the minimum volume ellipsoid that covers (at least) $h < n$ points of the set $\mathcal{X}$, where $[n/2] + 1 \leq h < n$. Note that for any fixed $h$, the possible number of such sets is $C_h^n$, where $C_h^n$ denotes $n$ choose $h$ and $[\cdot]$ is the rounding operator. The center and shape of the ellipsoid are defined by the robust estimates for the mean and the covariance matrix respectively.

The MVE estimation procedure (or algorithm), known as MINIVOL (Rousseeuw and Leroy 1987), entails an implicit combinatorial problem with cardinality $C_h^n$, which is prohibitive even for small experimental datasets encountered in machine learning

and pattern recognition applications. Therefore, MINOVOL relies on a repeated sampling procedure to find a subset of $h$ samples out of $n$ for which there exists a minimal covering ellipsoid. Observe here the two objectives implicit in the MINIVOL procedure: (*i*) Find all the subsets of at least $h$ samples (i.e. in terms of cardinality) and, (*ii*) Select the subset which has the minimal volume covering ellipsoid. Thus, the objective of MINIVOL is to find the smallest and most compact ellipsoid that encloses a set of $h$ points. The caveat when using MINVOL, however, is that the proportion of outliers $\alpha$ should be known in a priori. If $\alpha$ is not known, one can guess a reasonable value for $\alpha$ or assume a worst case scenario and set $\alpha = 49\%$.

Unfortunately, this approach for estimating the MVE is not feasible in our context for a few reasons. First, the proportion of outliers $\alpha$ is usually unknown, and assuming a predefined value such as 49% has no justification. Second, there is a practical inefficiency in running a sampling–based procedure such as MINIVOL for each data point. Third, considering our local learning context, large deviating samples might not necessarily be outliers; for instance samples from other classes or clusters cannot be considered as outliers. Therefore, a different approach is needed for computing the MVE estimate for $\mathbf{A}$.

### 4.1 Approximating the MVE estimate

The objective of MINIVOL is to find the smallest and most compact ellipsoid that encloses a set of $h$ points. In our context, the sampling procedure for selecting $h$ out of $n$ samples can be avoided since estimating $\mathbf{A}_q$ only requires the points in $\mathcal{N}(\mathbf{x}_q)$. Recall that $\mathcal{N}(\mathbf{x}_q)$ contains a small number of points that are close (in terms of the initial distance) or similar to $\mathbf{x}_q$. This is known as "locality", since in the neighborhood $\mathcal{N}(\mathbf{x}_q)$, the underlying distribution is assumed to smoothly vary around $\mathbf{x}_q$. In this setting, and assuming that $d$ is large, it is expected that $m \ll d$. If $\mathbf{A}_q$ is estimated for $\mathcal{N}(\mathbf{x}_q)$ using a maximum likelihood estimation (MLE) procedure then the result will be a crude low rank estimate for $\mathbf{A}_q$. In the same spirit of MINIVOL, and to avoid an unreliable estimate for $\mathbf{A}_q$, we propose to estimate $\mathbf{A}_q$ using the Minimum Volume Covering Ellipsoid (MVCE) for the set $\mathcal{N}(\mathbf{x}_q)$ (Atwood 1969; Titterington 1978). In other words, we approximate the functionality of the MINIVOL estimation procedure using (*i*) locality (Bottou and Vapnik 1992) and (*ii*) the MVCE of $\mathcal{N}(\mathbf{x}_q)$. The MVCE of $\mathcal{N}(\mathbf{x}_q)$ has interesting properties and will be explained in the following section.

### 4.2 Formulation of the MVCE

Let $\mathcal{N} = \mathcal{N}(\mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be the set of NNs for $\mathbf{x}_q$. The MVCE of $\mathcal{N}$, known as the *Löwner–John Ellipsoid*, is denoted by $\mathcal{E}_{\text{LJ}}$ and parameterized by $\mathbf{A} \in \mathbb{S}_{++}^{d \times d}$ and $\mathbf{u} \in \mathbb{R}^d$ as follows (Boyd and Vandenberghe 2004):

$$\mathcal{E}_{\text{LJ}} = \{\mathbf{x} \mid (\mathbf{x} - \mathbf{u})^\top \mathbf{A}^{-1} (\mathbf{x} - \mathbf{u}) \leq d, \forall \mathbf{x} \in \mathcal{N}\}, \tag{1}$$

where $\mathbf{u}$ is the center of the ellipsoid. The problem of computing the MVCE of $\mathcal{N}$ can be expressed as:

$$(\widehat{\mathbf{A}}^*, \widehat{\mathbf{u}}^*) = \arg\min_{\mathbf{A},\mathbf{u}} \quad \log\det(\mathbf{A}) \tag{2}$$

$$\text{s.t.} \quad \|\mathbf{A}^{-\frac{1}{2}}\mathbf{x}_j - \mathbf{b}\|_2^2 \leq d, \quad 1 \leq j \leq m, \tag{3}$$

where $\mathbf{b} = \mathbf{A}^{-\frac{1}{2}}\mathbf{u}$, and $\mathbf{A} \in \mathbb{S}_{++}^{d\times d}$. The optimization problem in (2) is a convex optimization problem since the objective function is convex in $\mathbf{A}$, and the squared norm constraints are convex quadratic inequalities in $\mathbf{A}$ and $\mathbf{b}$.

### 4.3 An algorithm for computing the MVCE

Computing the MVCE was studied in various research disciplines such as optimal experimental design (Atwood 1969; Titterington 1978), operations research (Sun and Freund 2004; Damla et al. 2008), numerical optimization (Boyd and Vandenberghe 2004), and computer science (Kumar and Yildirim 2005). The resulting algorithms fall into two main categories: (*i*) first order gradient based methods (Atwood 1969; Titterington 1978; Kumar and Yildirim 2005), and (*ii*) second order methods that rely on convex optimization and interior point methods (Boyd and Vandenberghe 2004; Sun and Freund 2004). Although interior point methods are considered to be an important advancement in optimization, their computational complexity per iteration is usually high (Todd 2006). In addition, in terms of convergence, Damla et al. (2008) showed the linear convergence of simple first order algorithms (Atwood 1969; Kumar and Yildirim 2005), thereby favoring first order algorithms over second order interior point methods.

For the purpose of MVEM, we use a first order method that was proposed by Titterington (1978). In particular, Titterington (1978) showed that the problem of computing the MVCE for a dataset can be regarded as the dual of a problem in optimal design for parameter estimation in linear regression, with the dataset as the design space. Consider again the set $\mathcal{N}(\mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, and let $\mathbf{w} = [w_1, \ldots, w_m]^\top$. Following Titterington (1978) and Dolia et al. (2006), a regularized version of dual problem in (2) can be expressed as:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \quad \log\det(\mathbf{A}(\mathbf{w})) \tag{4}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} w_j = 1, \text{ and } w_j \geq 0,$$

where

$$\mathbf{A}(\mathbf{w}) = \sum_{j=1}^{m} w_j(\mathbf{x}_j - \mathbf{u}(\mathbf{w}))(\mathbf{x}_j - \mathbf{u}(\mathbf{w}))^\top + \varepsilon\mathbf{I}\,, \qquad \mathbf{u}(\mathbf{w}) = \sum_{j=1}^{m} w_j\mathbf{x}_j\,,$$

from which $\mathbf{A}$ was eliminated to yield an optimization problem in the dual variable $\mathbf{w}$. The term $\varepsilon\mathbf{I}$, $0 < \varepsilon \ll 1$, is a regularization term that prevents the ellipsoid from collapsing to zero volume in high-dimensional spaces. The dual of problem (4) in terms of $\mathbf{A}$ and $\mathbf{u}$ is thus given by:

**Algorithm 2** Modified Titterington's MVCE algorithm (for a fixed center $\mathbf{x}_q$).

**Require:** The query point: $\mathbf{x}_q$; the set of NNs for $\mathbf{x}_q$: $\mathcal{N}(\mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$; maximum number of iterations: $t_{\max}$; a stopping threshold for the norm of difference in $\mathbf{w}$ vectors: $\tau > 0$; and the regularization parameter $\varepsilon$ to keep a minimum volume for the ellipsoid.

$t = 0$
For $1 \leq j \leq m$, set $w_j(t) = 1/m$
$\mathbf{w}(t) = [w_1(t) \quad \cdots \quad w_m(t)]$
**for** $t = 0$ to $t_{\max}$ **do**
  $\mathbf{A}(t) = \sum_{j=1}^{m} w_j(t)(\mathbf{x}_j - \mathbf{x}_q)(\mathbf{x}_j - \mathbf{x}_q)^\top + \varepsilon \mathbf{I}$
  **for** $j = 1$ to $m$ **do**
    $\delta_j = (\mathbf{x}_j - \mathbf{x}_q)\mathbf{A}^{-1}(t)(\mathbf{x}_j - \mathbf{x}_q)^\top$
    $w_j(t+1) = \frac{w_j(t)\delta_j}{d}$
  **end for**
  if $\|\mathbf{w}(t) - \mathbf{w}(t+1)\| \leq \tau$ then break
**end for**
**return** $\widehat{\mathbf{A}}^* = \mathbf{A}(t)$

$$(\widehat{\mathbf{A}}^*, \widehat{\mathbf{u}}^*) = \underset{\mathbf{A},\mathbf{u}}{\arg\min} \quad \log\det(\mathbf{A}) + d + \varepsilon \mathrm{tr}(\mathbf{A}^{-1}) \tag{5}$$
$$\text{s.t.} \quad \|\mathbf{A}^{-\frac{1}{2}}\mathbf{x}_j - \mathbf{b}\|_2^2 \leq d, \quad 1 \leq j \leq m \ ,$$

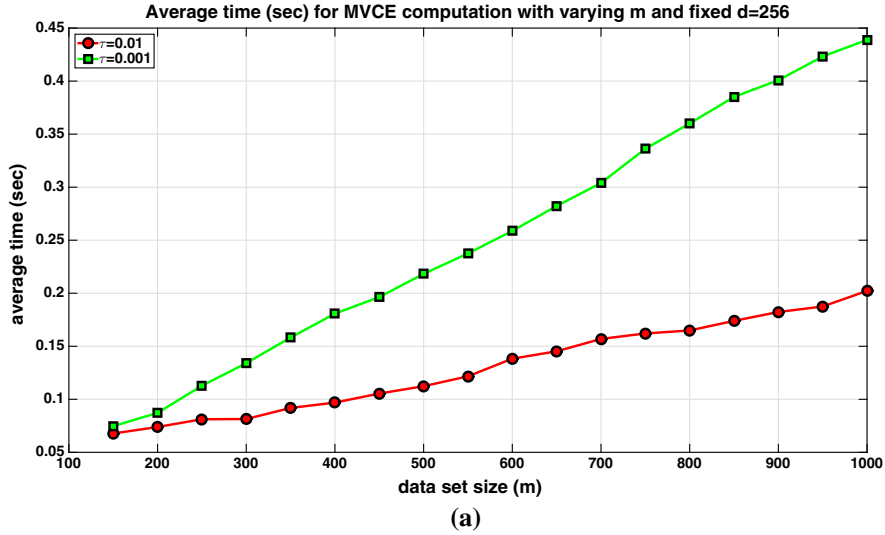
where $\mathbf{b} = \mathbf{A}^{-\frac{1}{2}}\mathbf{u}$, and $\mathbf{A} \in \mathbb{S}_{++}^{d \times d}$. To see how problem (5) is different from the MLE for $\mathbf{A}$, consider the following optimization problem:

$$\widehat{\mathbf{A}}_{\text{MLE}} = \underset{\mathbf{A}}{\arg\min} \quad \log\det(\mathbf{A}) + \mathrm{tr}(\mathbf{A}^{-1}\mathbf{R}) \tag{6}$$
$$\text{s.t.} \quad \mathbf{A} \in \mathbb{S}_{++}^{d \times d} \ ,$$

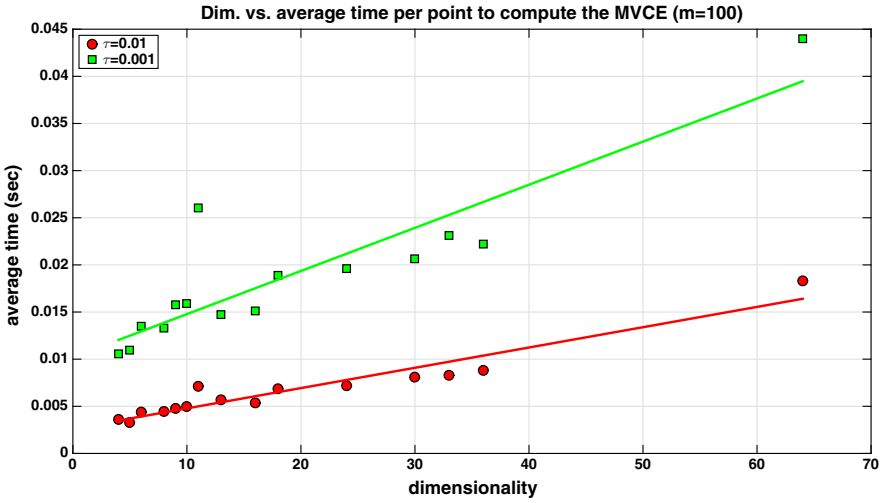
which estimates the covariance matrix $\mathbf{A}$ for a set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ with zero mean, and $\mathbf{R} = \sum_{j=1}^{m} \mathbf{x}_j \mathbf{x}_j^\top$ is the sample covariance matrix. With no further constraints on $\mathbf{A}$, $\widehat{\mathbf{A}}_{\text{MLE}} = \mathbf{R}$. Thus the difference between (5) and (6) is in the regularization term, and the $m$ linear matrix inequality constraints that define the MVCE.

From problem (4), it can be seen that $\widehat{\mathbf{A}}^*$ is a weighted average of rank one covariance matrices where the weights are restricted to form a convex combination. Setting all the weights $w_j = 1/m$ for $1 \leq j \leq m$, yields the maximum likelihood estimate of the mean of $\mathbf{u}$ and covariance matrix $\mathbf{A}$. Note that $w_j$'s are estimated from the samples such that outer samples are assigned larger weights than inner ones.

Titterington's first order algorithm for solving problem (4) was proposed in (Titterington 1978). Algorithm 2 depicts a modified version of this algorithm in which the ellipsoid center is given as an input to the algorithm. Note that in our context, we only need to estimate the covariance matrix $\mathbf{A}$ and not the mean vector, or the ellipsoid center $\mathbf{u}$. In our context, the mean vector is the point of interest, $\mathbf{x}_q$, which is fixed and does not change. The implementation includes the inversion of a symmetric PD matrix which can be efficiently done using Cholesky factorization. The computational complexity per iteration is $O(d^3 + 4m(d^2 + d))$ flops and is dominated by the Cholesky factorization with a complexity of $O(d^3)$ flops. This factorization is independent from the number of points since it is computed in the outer $t$ loop of Algorithm 2. Accuracy




**(a)**




**(b)**

**Fig. 1** **a** The effect of the dataset size ($m$) on the average time for computing the MVCE using our implementation of Titterington's algorithm. The average time is computed over 100 randomly sampled points from the training set of the USPS dataset ($d = 256$). For each random sample, the neighborhood size $m$ varied from 150 to 1000 NNs. **b** The effect of the dataset dimensionality ($d$) on the average time per point for computing the MVCE using our implementation of Titterington's algorithm. The average time is computed over 100 samples picked randomly from the following UCI datasets: iris, new-thyroid, bupa, pima, glass, pageblocks, vowel, wine, housevotes, lymphography, german, WDBC, Ionosphere, satimages, and optdigits. The experiments were carried on a Dell compute server with two Intel Xeon quad-core processors (E5345 @ 2.33 GHz) with 16 GB RAM

and speed of convergence, depicted in Fig. 1a, b respectively, depend on parameter $\tau$ which in turn is affected by the size and dimensionality of the dataset. Note that when $t = 0$, the initial weights for $\mathbf{w}$ in step 2 lead to the regularized sample covariance matrix in step 5.

**Table 1** Attributes of the twenty UCI datasets used in our experiments; number of classes ($c$), size ($n$) and number of features ($d$)

| Dataset | $c$ | $n$ | $d$ | Dataset | $c$ | $n$ | $d$ |
|---|---|---|---|---|---|---|---|
| Balance | 3 | 625 | 4 | Monks-3 | 2 | 554 | 6 |
| Bupa | 2 | 345 | 6 | NewThyroid | 3 | 215 | 5 |
| German | 2 | 1000 | 24 | Pageblocks | 5 | 5473 | 10 |
| Glass | 7 | 214 | 9 | Pima | 2 | 768 | 8 |
| HouseVotes | 2 | 341 | 16 | SatImage | 6 | 6435 | 36 |
| Ionosphere | 2 | 350 | 33 | Segment | 7 | 2086 | 18 |
| Iris | 3 | 150 | 4 | Spam | 2 | 4601 | 57 |
| Lymphography | 4 | 148 | 18 | WDBC | 2 | 569 | 30 |
| Monks-1 | 2 | 556 | 6 | Wine | 3 | 168 | 13 |
| Monks-2 | 2 | 601 | 6 | Yeast | 10 | 1484 | 6 |

**Table 2** Attributes of the MNIST and USPS datasets

| Dataset | $c$ | Train | Test | $d$ | PCA |
|---|---|---|---|---|---|
| MNIST | 10 | 19997 | 10000 | $(24 \times 24)$ 576 | Yes (120) |
| USPS | 10 | 7291 | 2007 | $(16 \times 16)$ 256 | No |

## 5 Experimental results

We validated the performance of the MVEM by running experiments on standard benchmark datasets with different size ($n$) and number of features ($d$). Our experiments were conducted on twenty datasets from the UCI machine learning repository (Newman et al. 1998), shown in Table 1, and two handwritten digits datasets, MNIST (LeCun 1998) and USPS (Keysers 1998), shown in Table 2. The performance on the twenty UCI datasets was evaluated in terms of the average error rate (with standard deviation) on the test sets from a 10–folds double cross validation. The performance on MNIST and USPS was evaluated using the classifier's error rate on the test set of both datasets.

### 5.1 Experimental setup

Due to the large size and the high-dimensionality of MNIST, it was preprocessed with some basic operations as follows. First, since the digits are relatively well centered with respect to the image boundaries, the images were cropped by two pixels from each side to form new images of $24 \times 24$ pixels. Next, similar to DeCoste and Schölkopf (2002), all images were smoothed with a 2D Gaussian kernel of width $\sigma = 0.75$. Finally, principal component analysis (PCA) was used to reduce the dimensionality of the dataset to 120 dimensions, retaining 99% of the total variance. To reduce the size of the training set, only one third of the samples in each digit class were randomly selected and used as the training set, resulting in a training set size of 19997 samples. Principal

components were obtained from the new training set of MNIST. No preprocessing was applied to all other dataset.

The MVEM—denoted here by RMVEM due to its internal regularization taking place—was compared with the Euclidean distance (EUC), the large margin nearest neighbor classifier (LMNN) (Weinberger et al. 2006), relevant component analysis (RCA) (Bar-Hillel et al. 2005), Xing's metric learning algorithm (XING) (Xing et al. 2003), $k$–Local Hyperplanes classifier (Vincent and Bengio 2002) (KLHP) and a local GQD distance metric based on regularized MLE estimate for $\mathbf{A}$ (RMLE) (see (6)). That is, RMLE and RMVEM are both GQD metrics but with different estimates for the covariance matrix. Note that RMVEM is unsupervised when defining the neighborhood for each point, while LMNN, RCA, XING, and KLHP are all supervised learning algorithms.

LMNN is a supervised and discriminative metric learning algorithm that is specifically designed to minimize the $k$-NN error, while RCA and XING are semi–supervised metric learning algorithms that use pairwise constraints or similarity information during their training. While XING's algorithm uses $+ve$ and $-ve$ constraints, RCA uses only $+ve$ constraints. In our experiments, both XING and RCA were provided with full pairwise constraints, i.e. they were used in a supervised setting. Note that XING and RCA are global metric learning algorithms with global constraints on the data while LMNN is a global metric learning algorithm with only local constraints. KLHP (Vincent and Bengio 2002), on the other hand, is a supervised local learning algorithm for classification tasks. Assuming a $c$-class problem, and for a query point, KLHP finds the $k$-NN from each class and constructs $c$ local hyperplanes for the $k$ neighboring samples from each class. The correct class of the query point is the one with the minimum distance between the query point and the $c$ hyperplanes. Since $k$ is a crucial parameter for KLHP, we do not restrict its value to $\{1, 3, 5\}$; rather, it is optimized to minimize the training error of each split of the dataset.

### 5.2 Training and testing procedures for the $k$-NN classifier using the MVEM

The test error in our context is based on a $k$-NN classifier with different values of $k$. Note that we do not optimize $k$ to obtain the best error rate, rather the $k$-NN classifier is run on each dataset with the following $k$ values: $\{1, 3, 5\}$. Our hypothesis is that under a "good" distance metric, for any value of $k$, the $k$-NN error rate should be consistently as good as or smaller than the error rate under the Euclidean distance for the same value of $k$.

The MVEM was applied on all datasets using the $k$-NN classifier as follows. For each split of the data into training and test sets, the training phase searches for $m^*$ (and optionally $r^*$) that minimizes the $k$-NN error rate on the training set using the following procedure.
For a fixed $k$ for the $k$-NN classifier do:

1. For each value of $m$ in the range $[m_1, \ldots, m_v]$ do:
   (a) For each point $\mathbf{x}_i$ in the training set, $1 \leq i \leq n$, find its $m$ NNs from the training set to form the set $\mathcal{N}(\mathbf{x}_i)$ and apply Algorithm 2 to obtain the estimate $\widehat{\mathbf{A}}_i$.

(b) For each point $\mathbf{x}_i$, use the GQD parameterized by $\mathbf{x}_i$ and $\widehat{\mathbf{A}}_i$ to find the $k$-NN samples for $\mathbf{x}_i$ from the given training set.
(c) Compute the error rate for the $k$-NN classifier on the training set.

2. Return $m^*$ that yields the lowest $k$-NN error rate on the training set.

Once $m^*$ is obtained for a training set, a new point $\mathbf{x}_q$ from the test set can be classified using the MVEM as follows:

1. Find the $m^*$ NNs for $\mathbf{x}_q$ from the given training set to form the set $\mathcal{N}(\mathbf{x}_q)$, and apply Algorithm 2 to obtain the estimate $\widehat{\mathbf{A}}_q$.
2. Use the GQD parametrized by $\mathbf{x}_q$ and $\widehat{\mathbf{A}}_q$ to find the $k$-NN for $\mathbf{x}_q$ from the given training set.
3. Classify $\mathbf{x}_q$ according to the $k$-NN rule using the NNs obtained in the previous step.

Note that in this procedure, $m$ is larger than $k$, and optimizing $m$ is done in a supervised manner since it is selected to minimize the $k$-NN error on the training set.

Titterington's MVCE algorithm requires two parameters: $\tau$ and $\varepsilon$. The algorithm is not sensitive to $\varepsilon$ and it was fixed in all our experiments to 0.0001. Convergence to an accurate estimate of the robust covariance matrix, however, is affected by $\tau$, especially in high dimensions. Therefore, $\tau$ needs to be small to obtain an accurate convergence for the covariance matrix. Convergence is also affected by the size and the dimensionality of the dataset as shown in Fig. 1a, b respectively. We found that $\tau$ values of 0.01, 0.001 or 0.0001 worked well for our experiments.

### 5.3 Results on the UCI datasets

Consider Figs. 2, 3, 4, 5, 6, 7 and 8 which compare the average test error (with standard deviation) for the $k$-NN classifier on all the datasets using the seven metrics/algorithms. For the UCI datasets, MVEM is consistently better than the Euclidean metric, and in most of the cases, it is consistently better than RCA, XING and KLHP.

Out of 60 cases, and using a $z$–test (for two proportions) with 0.05% significance level, MVEM was significantly better than EUC, LMNN, RCA, XING, KLHP and RMLE in 42, 30, 44, 38, 46, and 22 cases, respectively.[1] On the other hand, EUC, LMNN, RCA, XING, KLHP and RMLE were significantly better than MVEM in 11, 23, 14, 15, 13, and 16 cases, respectively. In the remaining cases, all algorithms are similar. The overall performance on the twenty UCI datasets is summarized in Fig. 7. It can be seen that RMVEM, on average, has the lowest average test error across all datasets and all $k$ values.

### 5.4 Results on MNIST and USPS datasets

In the MNIST and USPS experiments, different observations can be made from Fig. 8. KLHP was the best performer on MNIST while not competitive with EUC, LMNN,

[1] The 60 cases are: 20 (datasets) × 3 ($k$ values).

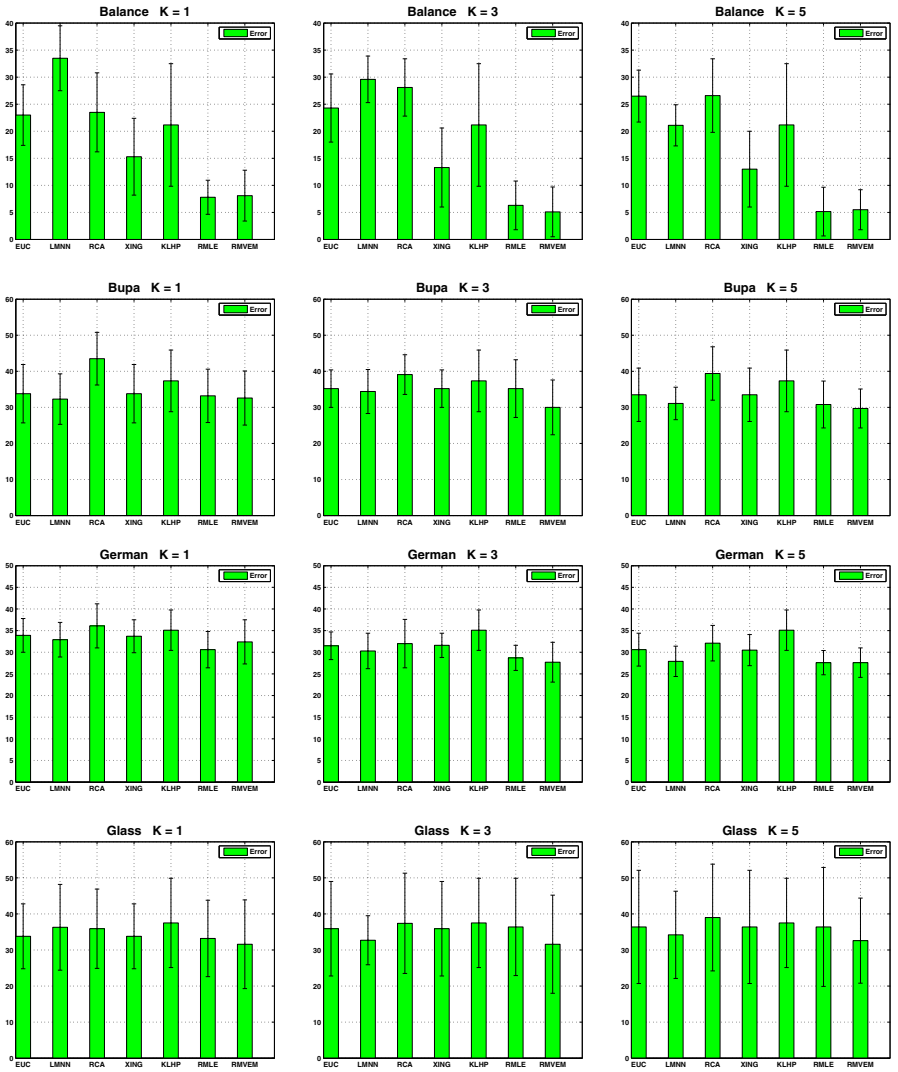



**Fig. 2** Average test error for the $k$-NN classifiers the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM) on Balance, Bupa, German and Glass datasets. The $y$-axis shows the error rate

XING, RCA and RMVEM on USPS. By contrast, LMNN was the best performer on the USPS dataset, but not competitive with EUC, RCA, XING and RMVEM in the MNIST case. RMVEM and XING performed similarly to EUC on both datasets. Before running any experiments, we expected that the performance of all metrics would be close to EUC in the MNIST case since it is an uncorrelated space (due to the preprocessing step using PCA). This hypothesis proved to be true for RMVEM and XING and false for all other metrics on MNIST and USPS datasets. This shows that both RMVEM and XING were able to learn a metric that is close to the Euclidean

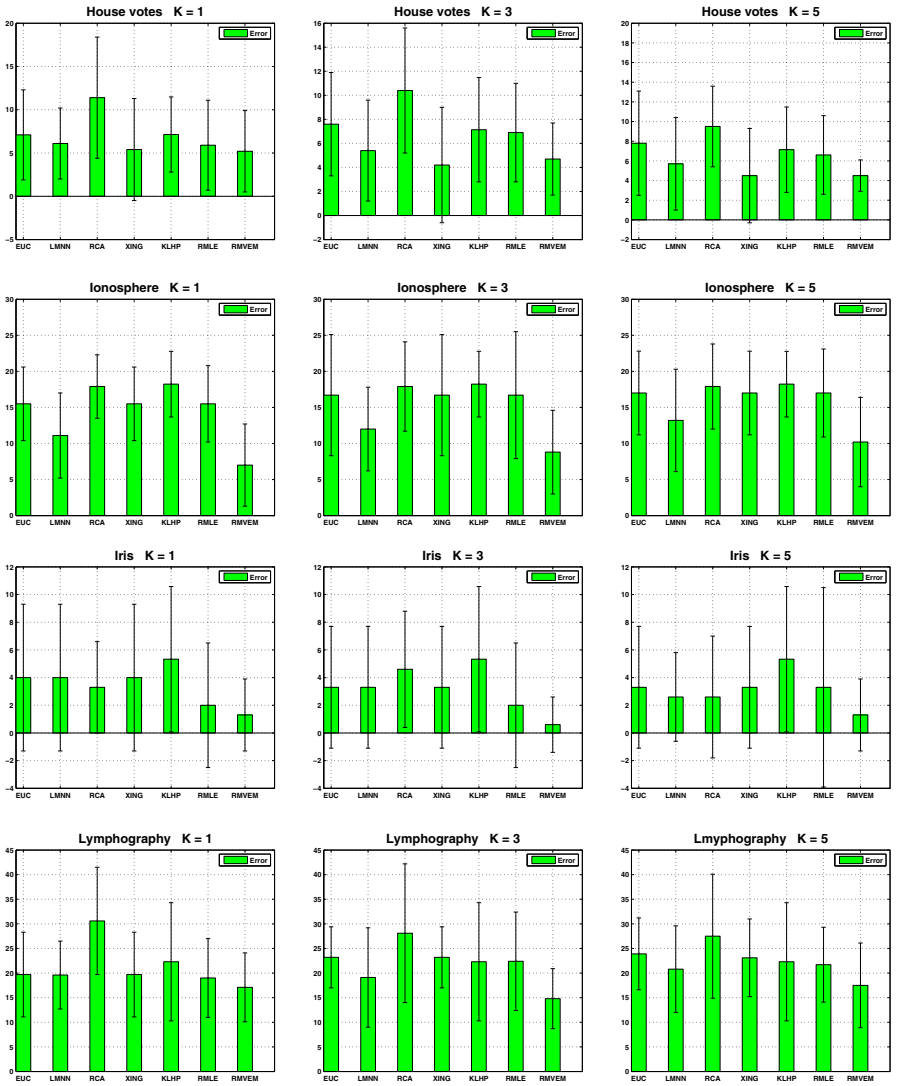



**Fig. 3** Average test error for the $k$-NN classifiers using the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM) on HouseVotes, Ionosphere, Iris and Lymphography datasets. The $y$-axis shows the error rate

distance from the data. On the other hand, LMNN, RCA and RMLE were not able to detect the uncorrelated space of MNIST and performed worse than EUC.

### 5.5 General remarks

Comparing the performance of each metric learning algorithm against the Euclidean metric, we note the following. Only RMVEM, LMNN and XING showed a consistent

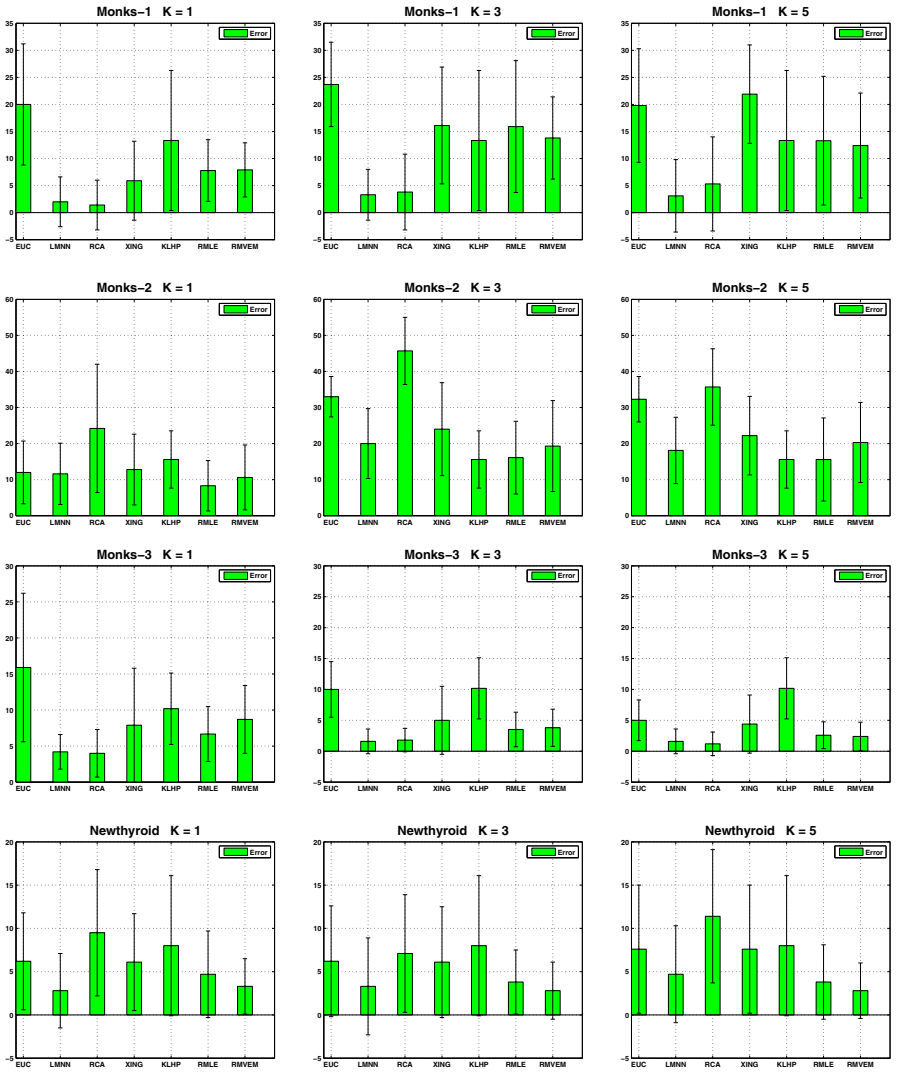



**Fig. 4** Average test error for the $k$-NN classifiers using the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM) on Monks-1, Monks-2, Monks-3 and New-thyroid datasets. The $y$-axis shows the error rate

improvement over the Euclidean metric. Moreover, we noticed from all datasets that the MVEM is more likely to be consistently better than, or at least as good as, the Euclidean distance. While KLHP was on average better than the Euclidean distance, it did not demonstrate such consistent behavior. This might be further improved by using better regularization in its implementation. RCA had less satisfactory behavior in that regard which suggests that it might be more useful in a semi–supervised setting for clustering as reported in (Bar-Hillel et al. 2005).

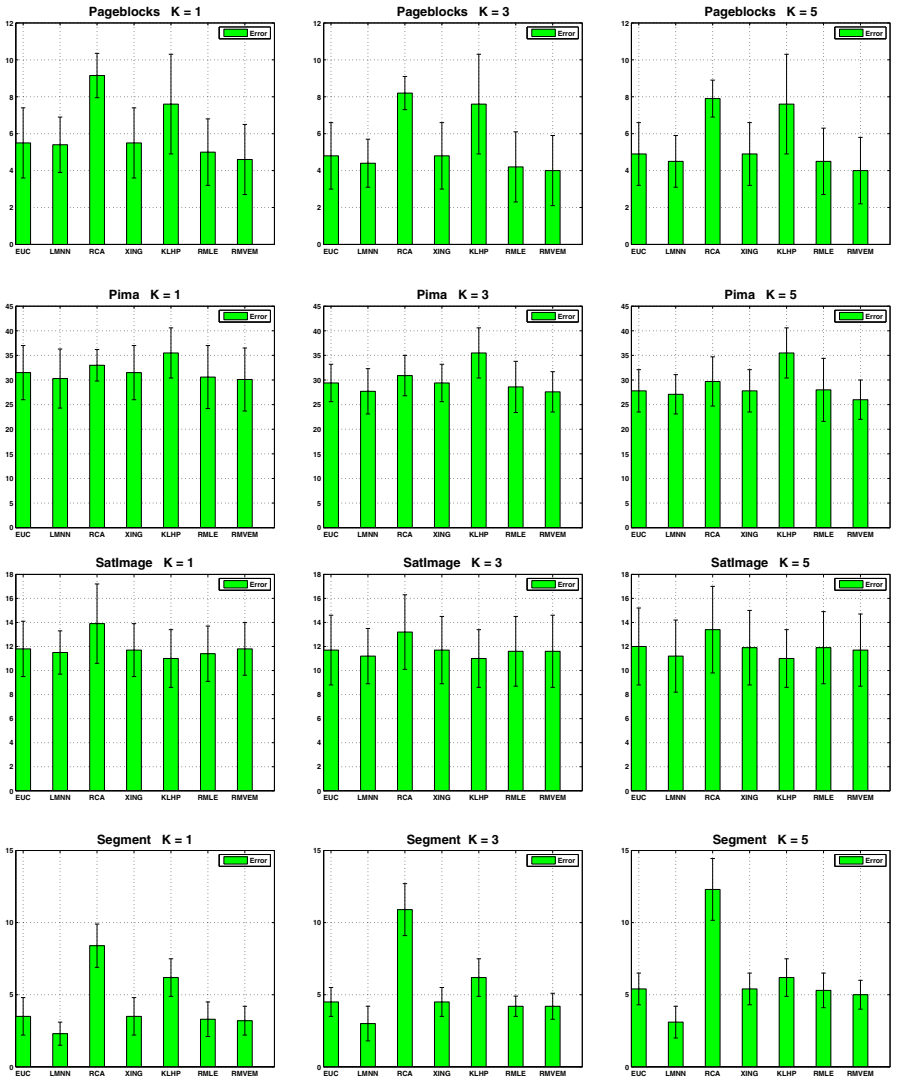



**Fig. 5** Average test error for the $k$-NN classifiers using the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM) on Page blocks, Pima, SatImage and Segment datasets. The $y$-axis shows the error rate

In general, for the $k$-NN classification setting, global metric learning with $+ve$ and $-ve$ constraints seems to perform better than global metric learning with $+ve$ constraints only. The former type (e.g. XING's algorithm) has a more difficult and slower learning since the two types of global constraints can work against each other. By exchanging global constraints with local ones, as with LMNN, performance improves significantly. RMVEM depends more on locality by first learning a metric for each point from the neighborhood information surrounding it and then fine tuning it with

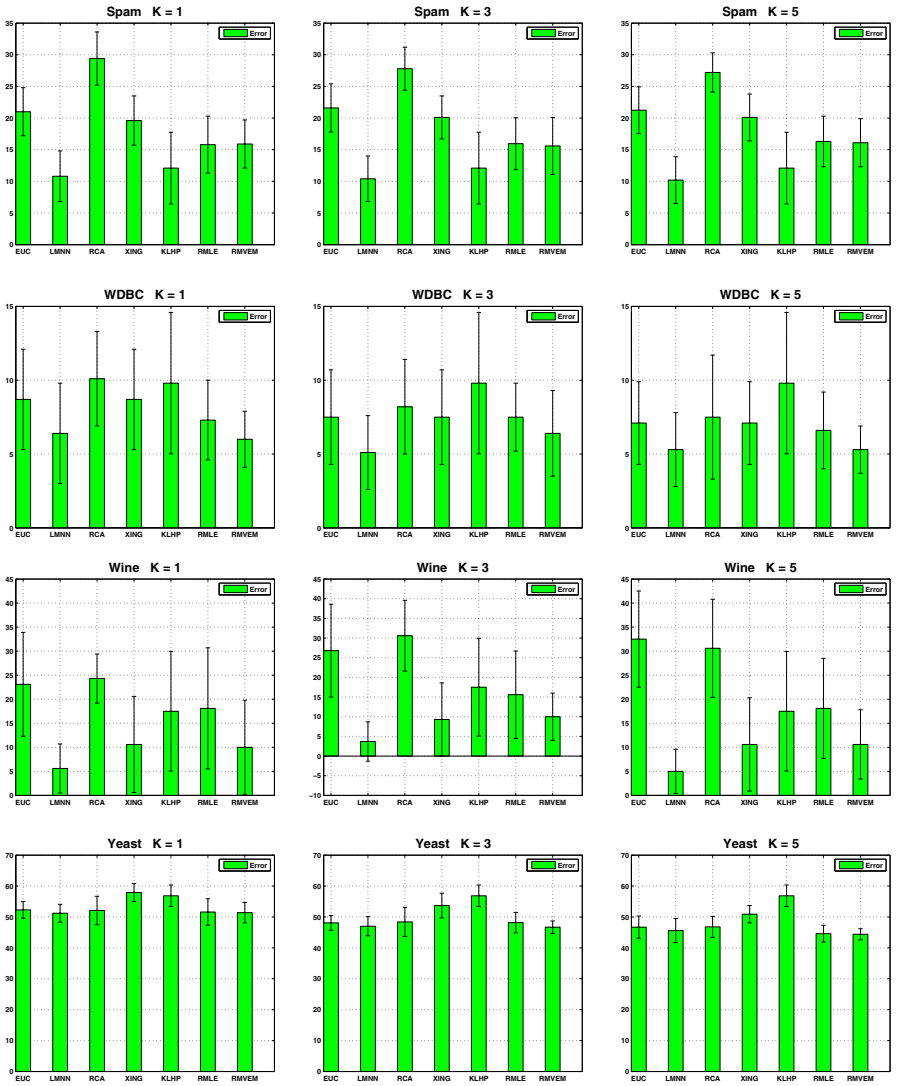



**Fig. 6** Average test error for the $k$-NN classifiers using the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM) for Spam, WDBC, Wine and Yeast datasets. The $y$-axis shows the error rate

parameters $m$ and $r$ that have to be consistent across the dataset. This resulted in a promising performance for RMVEM when compared to supervised learning algorithms that rely on class labels and side information. Our results suggest that while various supervised learning algorithms have been proposed for learning distance metrics to improve subsequent analysis, a simple unsupervised local learning approach for a distance metric can yield results that are as good as, or better than supervised and sophisticated techniques.

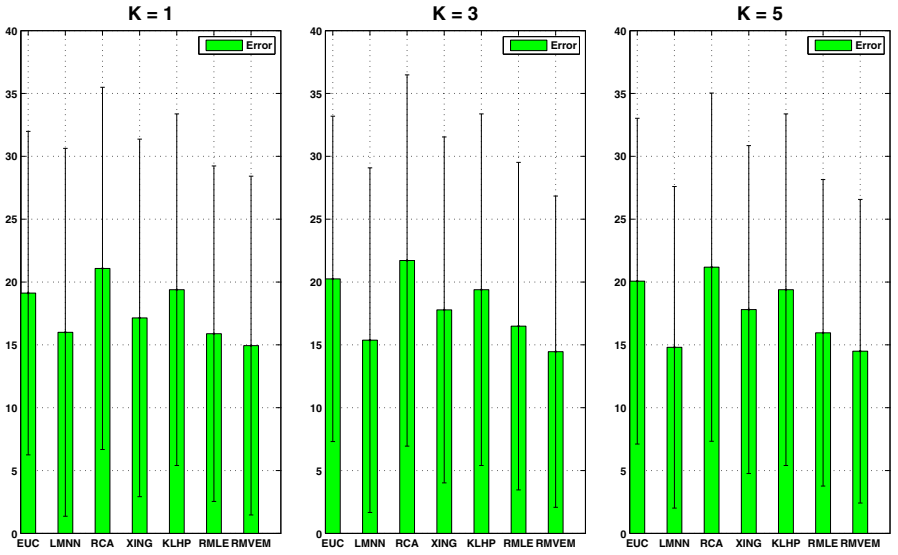



**Fig. 7** The average test error for each metric/algorithm on all UCI datasets. The $y$-axis shows the error rate

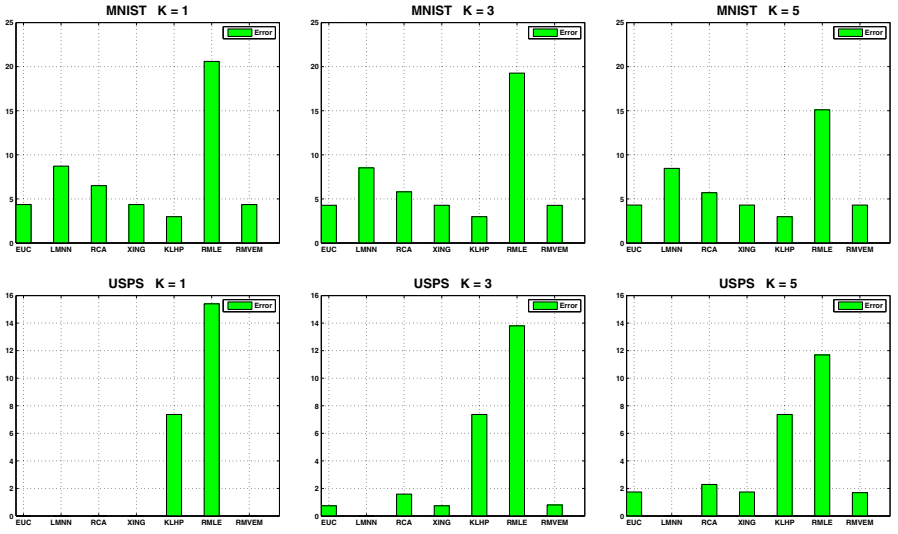



**Fig. 8** Error rates on the test sets of MNIST and USPS for the $k$-NN classifier using the seven metrics/algorithms (EUC, LMNN, RCA, XING, KLHP, RMLE, RMVEM). The $y$-axis shows the error rate

## 6 Concluding remarks

In this work, we propose a new approach for learning a data-dependent distance metric function that can find accurate nearest neighbors for a query point of interest. Our approach defines local distance functions based only on the set of nearest neighbors

falling in the neighborhood for the query point of interest. Such local distance metrics are modeled as quadratic distance functions parameterized by symmetric PD matrices. The proposed local based distance model allows the generalized quadratic distance function to change slowly, so that nearby query points generally have distance metrics that are similar. To obtain reliable estimates for the symmetric PD matrices, the local learning mechanism together with the Minimum Volume Covering Ellipsoid algorithm allowed us to approximate the functionality of the Minimum Volume Ellipsoid estimator, which is known as a robust estimator for covariance matrices. Our experimental results show that the MVEM is a promising direction for defining suitable metrics for such query-based operations. Further, the distance function is flexible enough to be adapted and optimized according to the learning task under consideration through its objective function.

Future research directions can address the algorithmic aspects of our approach, as well as new application domains. On the algorithmic side, faster and more efficient computation of the minimum volume covering ellipsoid is an important direction especially for high-dimensional data. On the applications side, the MVEM can be applied to image retrieval, object recognition, and appearance–based matching. The proposed local distance functions can be also used as inputs for other distance-based algorithms. For instance, this approach was used for constructing neighborhood graphs for spectral clustering and manifold learning algorithms for nonlinear dimensionality reduction (Abou-Moustafa et al. 2013).

**Acknowledgements** We would like to thank our Coordinating Editor and our anonymous Reviewers for their rigorous comments that improved various sections of the manuscript. We also would like to thank Michael Smith, Prasun Lala, Catherine Laporte, and Mathew Toews for reading earlier versions of this manuscript. The authors also acknowledge the supported of the Natural Sciences and Engineering Research Council of Canada, under Discovery Grant RGPIN-2016-04638.

## References

Abou-Moustafa K, Ferrie F (2007) The minimum volume ellipsoid metric. In: LNCS 4713, 29th Symp. of the German Association of Pattern Recognition (DAGM), Springer, Heidelberg, pp 335–344

Abou-Moustafa K, Ferrie F (2012) A note on metric properties of some divergence measures: the Gaussian case. JMLR W&CP 25:1–15

Abou-Moustafa K, Schuurmans D, Ferrie F (2013) Learning a metric space for neighbourhood topology estimation. Appl Manifold Learn JMLR W&CP 29:341–356

Aggarwal C, Hinneburg A, Keim D (2001) On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche J, Vianu V (eds) Database Theory, vol 1973., Lecture Notes in Computer Science, Springer, pp 420–434

Atwood C (1969) Optimal and efficient design of experiments. Ann Math Stat 40:1570–1602

Bar-Hillel A, Hertz T, Shental N, Weinshall D (2005) Learning a Mahalanobis metric from equivalence constraints. J Mach Learn Res 6:937–965

Belkin M, Niyogi P (2003) Laplacian eigenmaps and spectral techniques for data representation. Neural Comput 15:1373–1396

Bottou L, Vapnik V (1992) Local learning algorithms. Neural Comput 4(6):888–900

Boyd S, Vandenberghe L (eds) (2004) Convex Optimization. Cambridge University Press

Chang H, Yeung DY (2007) Locally smooth metric learning with application to image retrieval. In: IEEE Proceedings of ICCV, pp 1–7

Coifman R, Lafon S (2006) Diffusion maps. Appl Comput Harmonic Anal 21:5–30

Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27

Damla S, Sun P, Todd M (2008) Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids. J Optim Methods Softw 23:5–19
DeCoste D, Schölkopf B (2002) Training invariant support vector machines. J Mach Learn 46(1–3):161–190
Ding C, Li T (2007) Adaptive dimension reduction using discriminant analysis and K–means clustering. In: ACM Proceedings of the 24th ICML
Dolia A, Bie TD, Harris C, Shawe-Taylor J, Titterington D (2006) The minimum–volume covering ellipsoid estimation in kernel–defined feature spaces. In: Proceedings of the 17th ECML, Springer
Domeniconi C, Gunopulos D (2002) Adaptive nearest neighbor classification using support vector machines. In: NIPS 14
Domeniconi C, Peng J, Gunopulos D (2002) Locally adaptive metric nearest neighbor classification. IEEE Trans PAMI 24(9):1281–1285
Dornaika F, El Traboulsi Y (2015) Learning flexible graph-based semi-supervised embedding. IEEE Trans Cybern 46(1):206–218
François D, Wertz V, Verleysen M (2007) The concentration of fractional distances. IEEE Trans Knowl Data Eng 19:873–886
Friedman J (1994) Flexible metric nearest neighbor classification. Department of Statistics, Stanford University, Technical report
Fukunaga K (ed) (1972) Introduction to statistical pattern recognition. Academic Press, Cambridge
Hastie T, Tibshirani R (1996) Discriminant adaptive nearest neighbor classification. IEEE Trans PAMI 18(6):607–615
Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. IEEE Trans Syst Man Cybern C Appl Rev 41(6):797–819
Keysers D (1998) United States Postal Service (USPS) data set. http://www-i6.informatik.rwth-aachen.de/keysers/usps.html
Kruskal J (1964) Nonmetric multidimensional scaling: a numerical method. Psychometrika 29:115–129
Kulis B (2013) Metric learning: a survey. Found Trends Mach Learn 5(4):287–364
Kumar P, Yildirim E (2005) Minimum-volume enclosing ellipsoids and core sets. J Optim Theory Appl 126(1):1–21
LeCun Y (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/
Macleod J, Luk A, Titterington DM (1987) A re-examination of the distance-weighted k-nearest neighbor classification rule. IEEE Trans Syst Man Cybern 17(4):689–696
Newman D, Hettich S, Blake C, Merz C (1998) UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/~mlearn/MLRepository.html
Peng J, Heisterkamp D, Dai H (2004) Adaptive quasi-conformal kernel for nearest neighbor classification. IEEE Trans PAMI 26(5):656–661
Rousseeuw P, Leroy A (eds) (1987) Robust regression and outlier detection. Wiley, New York
Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding (LLE). Science 290(5500):2323–2326
Schultz M, Joachims T (2004) Learning a distance metric from relative comparisons. In: Thrun S, Saul LK, Schölkopf PB (eds) NIPS 16, MIT Press
Shepard R (1962) The analysis of proximities: multidimensional scaling with an unknown distance function. Psychometrika 27:219–246
Short R, Fukunaga K (1981) The optimal distance measure for nearest neighbor classification. IEEE Trans Inf Theory 27(5):622–627
Sun P, Freund R (2004) Computation of minimum-volume covering ellipsoids. Op Res 52:690–706
Tenenbaum J, de Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
Titterington D (1978) Estimation of correlation coefficients by ellipsoidal trimming. J Appl Stat 27(3):227–234
Todd M (2006) On minimum–volume ellipsoids. From John and Kiefer–Wolfowitz to Khachiyan and Nesterov–Nemirovski. Slides presented in the ABEL Symposium, http://people.orie.cornell.edu/~miketodd/ublndeta.pdf
Vincent P, Bengio Y (2002) K–Local hyperplane and convex distance nearest neighbor algorithms. In: Dietterich TG, Becker S, Ghahramani Z (eds) NIPS 14, The MIT Press, pp 985–992
Weinberger K, Blitzer J, Saul L (2006) Distance metric learning for large margin nearest neighbor classification. In: Weiss Y, Schölkopf PB, Platt JC (eds) NIPS 18, MIT Press, pp 1473–1480

Xing E, Ng A, Jordan M, Russell S (2003) Distance metric learning with application to clustering with side–information. In: Becker S, Thrun S, Obermayer K (eds) NIPS 15, MIT Press, pp 505–512

Yang L (2006) Distance metric learning: A comprehensive review. Technical report, Department of Computer Science and Engineering, Michigan State University