# Divergence Based Graph Estimation for Manifold Learning

Karim T. Abou–Moustafa
Dept. of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
aboumous@cs.ualberta.ca

Frank Ferrie
Centre of Intelligent Machines
McGill University
Montréal, QC H3A 0E9, Canada
ferrie@cim.mcgill.ca

Dale Schuurmans
Dept. of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
dale@cs.ualberta.ca

***Abstract*—Manifold learning algorithms rely on a neighbourhood graph to provide an estimate of the data's local topology. Unfortunately, current methods for estimating local topology assume local Euclidean geometry and locally uniform data density, which often leads to poor embeddings of the data. We address these shortcomings by proposing a framework that combines local learning with parametric density estimation for local topology estimation. Given a data set $\mathcal{D} \subset \mathcal{X}$, we first estimate a new metric space $(\mathbb{X}, d_{\mathbb{X}})$ that characterizes the varying sample density of $\mathcal{X}$ in $\mathbb{X}$, and then use $(\mathbb{X}, d_{\mathbb{X}})$ as a new (pilot) input space for manifold learning. The proposed framework results in significantly improved embeddings, which we demonstrated objectively by assessing clustering accuracy.**

***Index Terms*—Manifold learning, divergence measures, neighbourhood graphs, graph topology estimation, divergence based graphs.**

## I. Introduction

Manifold learning algorithms have recently played a crucial role in unsupervised learning tasks such as clustering and nonlinear dimensionality reduction [15], [13], [5], [3], [6]. A common aspect of these algorithms is that they rely on a neighbourhood graph constructed from the input data[1] $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ with the points $\mathbf{x}_i \in \mathbb{R}^d$ as its vertices. Such a graph provides an estimate for the topology of an underlying low dimensional manifold that (approximately) encapsulates the data. A manifold learning algorithm then tries to "unfold", or "flatten" this manifold—while preserving some local information—to partition the graph (e.g. as in clustering), or to redefine some metric information (e.g. as in dimensionality reduction). The standard distance used to construct this data graph—whether it be a fully connected, $\epsilon$-ball, or $k$-nearest neighbours graph—is the Euclidean distance. Unfortunately, the Euclidean distance creates severe inaccuracy problems for graph estimation, and consequently for the manifold learning process. In this paper, we show how to overcome these inaccuracies by introducing a new manifold learning framework that mitigates the liability incurred by using Euclidean geometry on real data.

One reason for inaccurate topology estimates is the finite nature of data, which means that low probability regions will be poorly sampled and hence poorly represented in $\mathcal{D}$. This results in an *uneven sample distribution* in the input space $\mathcal{X}$ [4]. In practice, the situation is exacerbated by noise, nonlinearity, and the high dimensionality of the data. Unfortunately, the Euclidean metric cannot accommodate any of these factors. First, the Euclidean distance, by definition, is constant over the entire input space $\mathcal{X}$, and hence does not take the varying sample distribution into consideration. To see this note that for any data of the form $\mathcal{D}$, the Euclidean distance enforces an identity covariance matrix $\mathbf{I}$ to measure pairwise distances between points. By expanding the squared norm of $\|\mathbf{x}-\mathbf{y}\|^2$ to $(\mathbf{x}-\mathbf{y})^\top \mathbf{I}(\mathbf{x}-\mathbf{y})$, one directly obtains a special case of the generalized quadratic distance (GQD) $d(\mathbf{x}, \mathbf{y}; \mathbf{A}) = \sqrt{(\mathbf{x}-\mathbf{y})^\top \mathbf{A}(\mathbf{x}-\mathbf{y})}$ which itself, generalizes the Mahalanobis[2] distance for any SPD matrix $\mathbf{A}$. If $\mathbf{A} = \mathbf{I}$, then the Euclidean distance enforces a unit variance for all variables in the data with zero correlation among them. Second, similar remarks apply for the GQD if $\mathbf{A}$ is the inverse of the data's global covariance matrix, or it is learned via a metric learning algorithm [19], [18] that might impose some local and/or global constraints on distances (based on labels or side information). For most metric learning algorithms $\mathbf{A}$ is constant over $\mathcal{X}$ and hence, it is still not a faithful modelling for the varying density in $\mathcal{X}$. More importantly, such metric learning algorithms are either supervised or semi-supervised, and hence they cannot be used in the unsupervised setting discussed here. These factors impart serious inaccuracy in the data graph construction, which in turn yields erroneous estimates for the manifold topology and increases uncertainty of point locations in the lower dimensional subspace. A manifestation of these effects is topological instability of manifold learning and sensitivity to noise [2].

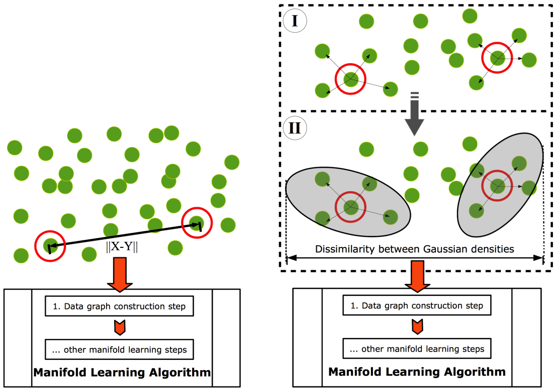
In this paper we propose an algorithmic framework that overcomes these liabilities by inferring a new input space for the manifold learner, such that the new input space, denoted $\mathbb{X}$, characterizes the varying sample density in the original input space $\mathcal{X}$. In particular, we integrate the concept of local learning algorithms [4], with parametric density estimation to learn from $\mathcal{D}$ a new metric space[3] $(\mathbb{X}, d_{\mathbb{X}})$ that becomes the (pilot) input space for the manifold learner. The proposed framework, depicted in Figure 1, redefines the proximity between two points in $\mathcal{D}$ based on the divergence between the local density surrounding each of the two points, then passes this proximity to the manifold learner. The set $\mathbb{X}$ contains all the parameters that define the local density for each point in $\mathcal{D}$, while the new proximity information, characterized by the divergence measure $d_{\mathbb{X}}$, defines the metric space $(\mathbb{X}, d_{\mathbb{X}})$.

Recent work in this direction has focused on patching some of the problems caused by the Euclidean distance when used in the neighbourhood graph construction. Due to space limitations, we refer to [10] and [7] as recent examples on such approaches. Unlike this research direction, our approach redefines a new input space with a different geometry and distance between points and anchor it to the manifold learner.

[1] Notations: Bold small letters $\mathbf{x}, \mathbf{y}$ are vectors. Bold capital letters $\mathbf{A}, \mathbf{B}$ are matrices. Calligraphic and double bold capital letters $\mathcal{X}$, $\mathcal{Y}$, $\mathbb{X}$, $\mathbb{Y}$ denote sets and/or spaces. Symmetric positive definite (SPD) and semi-definite (SPSD) matrices are denoted by $\mathbf{A} \succ 0$ and $\mathbf{A} \succeq 0$ respectively. $\mathrm{tr}(\cdot)$ is the matrix trace and $|\cdot|$ is the matrix determinant.

[2] In this case $\mathbf{A}$ is the inverse of the data's covariance matrix.

[3] A metric space is an ordered pair $(\mathcal{X}, d)$ such that $\mathcal{X}$ is a non-empty abstract set (of any elements, whose nature is left unspecified), and $d$ is a distance function, or a metric, defined as: $d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, and the following axioms hold for all $a, b, c \in \mathcal{X}$: (i) $d(a, b) \geq 0$, (ii) $d(a, a) = 0$, (iii) $d(a, b) = 0$ iff $a = b$, (iv) Symmetry: $d(a, b) = d(b, a)$, and (v) The triangle inequality: $d(a, c) \leq d(a, b) + d(b, c)$. Semi-metrics satisfy axioms (i), (ii), and (iv) only. The axiomatic definition of metrics and semi-metrics, in particular axioms (i) and (ii), produce the positive semi-definiteness of $d$. Hence metrics and semi-metrics are positive semi-definite (PSD).


||X-Y||
1. Data graph construction step
... other manifold learning steps
Manifold Learning Algorithm
I
II
Dissimilarity between Gaussian densities
1. Data graph construction step
... other manifold learning steps
Manifold Learning Algorithm

(a)

(b)

Fig. 1. (a) The common used distance for the data graph construction step in manifold learning is Euclidean distance, which is the straight line between the points $\mathbf{x}$ and $\mathbf{y}$. (b) In the proposed framework (dashed black box), the distance between any two points is defined in two steps: I. Each point defines a local neighbourhood by finding its $m$ nearest neighbours. II. A Gaussian distribution with a regularized full covariance is fitted to each local neighbourhood. The distance between the two points is then the divergence between these two Gaussian distributions that is then conveyed to the graph construction step of manifold learning.

## II. The Elements of the Set $\mathbb{X}$

We assume that the input space $\mathcal{X}$ 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 small neighbourhood $\mathcal{N}$ around each point $\mathbf{x} \in \mathcal{X}$. In the finite sample setting, for each $\mathbf{x}_i \in \mathcal{D}$, the neighbourhood $\mathcal{N}(\mathbf{x}_i)$, or $\mathcal{N}_i$ for short, is the set of neighbouring points for $\mathbf{x}_i$, which can be defined using an $\epsilon-$ball or the $m$ nearest neighbours (NNs) of $\mathbf{x}_i$. In principle, the neighbourhood size should slowly grow until it circumvents the region where Euclidean geometry holds, and going beyond that size, the local Euclidean assumption will break due to the manifold curvature. Hence, if $m$ is too small, the estimate for the local Euclidean subspace will be poor and inaccurate, while if $m$ is too large, the local linear structure will be smoothed out by the influence of far away points. In practice, $m$ can be set either by using cross validation, grid search, or the method of [5]. See [4], [5], [3] for examples on using local neighbourhoods for manifold learning.

For large data sets with high dimensionality, finding the NNs for each $\mathbf{x}_i$ can be time consuming. In this case, approximate neighbours can be found using methods based on random projections. For example, locality-sensitive hashing can be used [9], where for $h$ hash tables, each defined using $k$ hash functions, the time complexity for finding $m$ approximate nearest neighbours for all query points $\mathbf{q}_i$ is reduced to guaranteed to be $O\left(nmh(k\tau + dn\epsilon)\right)$. Here $\tau$ is the time to evaluate the hash function on point $\mathbf{x}$, and $\epsilon << 1$ is the probability that the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, is greater than a predefined threshold. Since $\tau$ is usually small, this complexity is substantially less than the $O(mn^2d^2)$ required for brute force search.

### A. Local learning for manifold estimation

Local learning algorithms overcome a nonuniform data distribution by introducing a local adjustment mechanism with control parameters that limit its impact to individual regions of the input space [4]. The Euclidean distance and the GQD do not have such a control mechanism, and hence do not take the varying sample density into consideration. We propose to introduce such a control mechanism to manifold learning in the following three steps:

1) Estimate the density for $\{\mathcal{N}_i\}_{i=1}^n$. The parameters of these densities will define the elements of $\mathbb{X}$.
2) Define $d_{\mathbb{X}}$ as a dissimilarity measure on $\mathbb{X}$ (§III).
3) Use $d_{\mathbb{X}}$ to construct the neighbourhood graph.

Formally, let $\mathcal{N}_i \equiv \mathcal{N}(\mathbf{x}_i) = \{\mathbf{x}_1^i, \dots, \mathbf{x}_m^i\}$, where $\mathbf{x}_j^i \in \mathcal{D}$, and $1 \leq i \leq n$. A reasonable density model for $\mathcal{N}_i$ under the local smoothness assumption is the Gaussian. Note that such a local density model does not impose any constraints or assumptions on the global density for the data. Let $\mathcal{G}_i(\mathbf{x}_i, \mathbf{\Sigma}_i)$ be the Gaussian density centered at $\mathbf{x}_i$ for neighbourhood $\mathcal{N}_i$, where $\mathbf{\Sigma}_i \in \mathbb{S}_{++}^{d \times d}$ is the sample covariance with respect to the mean $\mathbf{x}_i$, given by $\mathbf{\Sigma}_i = m^{-1}\sum_{\mathbf{x} \in \mathcal{N}_i}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^\top$, and $\mathbb{S}_{++}^{d \times d}$ is the space of symmetric positive definite matrices. Since in practice it might be that $d \gg m$, the sample covariance $\mathbf{\Sigma}_i$ will be a poor estimate for the true covariance, and generally rank deficient. Therefore, $\mathbf{\Sigma}_i$ can be replaced with the regularized estimate $\mathbf{R}_i \in \mathbb{S}_{++}^{d \times d}$ which shall be formally discussed in §II-B.

In terms of local learning, $\boldsymbol{\mu}_i$ and $\mathbf{R}_i$ are the local parameters that provide the means for coping with the uneven sample distribution. Ideally, each $\mathcal{G}_i$ defines a local neighbourhood around the point $\mathbf{x}_i$ with axes defined by the eigenvectors of $\mathbf{\Sigma}_i$, while its eigenvalues indicate the amount of data variance along each axis direction. If the data manifold is locally linear in the vicinity of $\mathbf{x}_i$, then all but the $d_0$ dominant eigenvalues will be very close to zero, while their associated leading eigenvectors will constitute the optimal variance preserving local coordinate system. In an ideal setting, a local maximum likelihood procedure will naturally capture this structure. However, to leverage the cases when $\mathbf{\Sigma}_i$ is degenerate for the above mentioned reasons, $\mathbf{\Sigma}_i$ is replaced with $\mathbf{R}_i$. This local Gaussian assumption at each point $\mathbf{x}_i$ is in the same spirit various of various manifold learning algorithms [5], [17],

Given the set $\{\mathcal{N}_i\}_{i=1}^n$, their corresponding local densities $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$ characterize the varying sample density for $\mathcal{X}$ according to the parameters $\boldsymbol{\mu}$ and $\mathbf{R}$. Since all local densities have the same parametric form, we define the set of 2-tuples $\{(\boldsymbol{\mu}_i, \mathbf{R}_i)\}_{i=1}^n \subset \mathbb{X}$, where $\mathbb{X} \subset \mathbb{R}^d \times \mathbb{S}_{++}^{d \times d}$. Note that $\mathcal{N}(\mathbf{x})$, $\boldsymbol{\mu}$, and $\mathbf{R}$ are defined in an unsupervised manner. However, if auxiliary information is available in the form of labels or side information, then the proposed approach can be extended to supervised and semi-supervised learning.

### B. Handling high dimensional data

For real world data sets, it is possible that $d$ is large thereby making $d$ larger than the number of samples $m$ in $\mathcal{N}(\mathbf{x})$. In this setting, standard assumptions of classical statistics can be easily violated causing conventional estimators to behave poorly. This problem of *"large $d$ small $m$"* is more serious when estimating a covariance matrix from a small number of samples. Indeed, accurate estimation of a covariance matrix from high dimensional data is a fundamental problem in statistics, since the number of parameters to be estimated grows quadratically with the number of variables. Learning a model under these conditions can be easily prone to overfitting, and yield poor generalization to out-of-sample points. In addition, computational tractability becomes another problem for handling such large matrices.

In this work, we consider shrinkage estimators that regularize the covariance matrix by shrinking it towards a symmetric target matrix such as a scaled version of the identity matrix: $\mathbf{R}_i = (1-\gamma)\mathbf{\Sigma}_i + \gamma d^{-1}\mathrm{tr}(\mathbf{\Sigma}_i)\mathbf{I}$, or the diagonal entries of $\mathbf{\Sigma}_i$: $\mathbf{R}_i = (1-\gamma)\mathbf{\Sigma}_i + \gamma\mathrm{diag}(\mathbf{\Sigma}_i)$, where $\gamma \in (0,1)$ is the mixing (or shrinkage) intensity coefficient. These regularized estimates are known to be statistically efficient, well conditioned, avoid local overfitting, and reduce the influence of outliers especially on the manifolds' boundaries.

Note that due to the structure of $\mathbf{R}_i$ and to the rank deficiency of $\mathbf{\Sigma}_i$, the computational efficiency (in space and time) can be greatly improved as follows. First, since $\mathbf{\Sigma}_i$ is rank deficient, it can be written as $\mathbf{V}_i\mathbf{\Lambda}_i\mathbf{V}_i^\top$, where $\mathbf{V}_i \in \mathbb{R}^{d\times m}$ and $\mathbf{\Lambda}_i \in \mathbb{R}^{m\times m}$ are respectively its eigenvector and eigenvalue matrices. Since $\mathbf{\Lambda}_i$ is diagonal and $m \ll d$, the storage required for $n$ $\mathbf{R}_i$'s, will be $n(dm + m + 2)$, which is significantly less than $nd(d + 1)/2$ as originally required. This in turn speeds up any matrix-matrix and matrix-vector computations since it is rarely required to form $\mathbf{R}_i$. Second, this decomposition improves computational efficiency since it leverages the need for explicitly computing the inverse of $\mathbf{R}_i$. Finally, note that for all $\mathbf{R}_i$'s, the spectral properties for $\mathbf{\Sigma}_i$ are not changed since the regularization only affects its eigenvalues but not its eigenvectors.

## III. The Divergence Measure $d_{\mathbb{X}}$

The measure $d_{\mathbb{X}}$ conveys the dissimilarity between two local densities, $\mathcal{G}_i$ and $\mathcal{G}_j$, which describe $\mathcal{N}_i$ and $\mathcal{N}_j$ around points $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. Note that $d_{\mathbb{X}}$ measures the difference between two local coordinate systems located at $\mathbf{x}_i$ and $\mathbf{x}_j$ in terms of : (*i*) the location, specified by $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$, and (*ii*) the scaling and orientation, specified by the eigenvectors of $\mathbf{R}_i$ and $\mathbf{R}_j$, which define the axes of each local coordinate system. It is important, therefore, to understand the properties of $d_{\mathbb{X}}$, how these properties can affect the graph topology estimation, and how they affect the final embedding.

Since the elements of $\mathbb{X}$ are the parameters for $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$, a natural measure for the dissimilarity between two densities is the divergence. For $\mathcal{G}_i$ and $\mathcal{G}_j$, some well known divergence measures with closed form expressions are: (1) The symmetric KL divergence:

$$d_J(\mathcal{G}_i, \mathcal{G}_j) = \tfrac{1}{2}\mathbf{u}^\top\mathbf{\Psi}\mathbf{u} + \tfrac{1}{2}\mathrm{tr}\{\mathbf{R}_i^{-1}\mathbf{R}_j + \mathbf{R}_j^{-1}\mathbf{R}_i\} - d, \qquad (1)$$

where $\mathbf{\Psi} = (\mathbf{R}_i^{-1} + \mathbf{R}_j^{-1})$, and $\mathbf{u} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$. (2) The Bhattacharyya distance $d_B$:

$$d_B(\mathcal{G}_i, \mathcal{G}_j) = \tfrac{1}{8}\mathbf{u}^\top\mathbf{\Gamma}^{-1}\mathbf{u} + \tfrac{1}{2}\ln\left[|\mathbf{R}_i|^{-\frac{1}{2}}|\mathbf{R}_j|^{\frac{-1}{2}}|\mathbf{\Gamma}|\right], \qquad (2)$$

where $\mathbf{\Gamma} = (\frac{1}{2}\mathbf{R}_i + \frac{1}{2}\mathbf{R}_j)$. (3) The Hellinger distance $d_H = \sqrt{2[1-\rho(\mathcal{G}_i,\mathcal{G}_j)]}$, where $\rho$ is the Bhattacharyya coefficient: $\rho(\mathcal{G}_i,\mathcal{G}_j) = |\mathbf{\Gamma}|^{-\frac{1}{2}}|\mathbf{R}_i|^{\frac{1}{4}}|\mathbf{R}_j|^{\frac{1}{4}}\exp\{-\frac{1}{8}\mathbf{u}^\top\mathbf{\Gamma}^{-1}\mathbf{u}\}$. (4) The Jeffreys-Riemann metric [1]:

$$d_{J\mathcal{R}}(\mathcal{G}_i, \mathcal{G}_j) \;\; = \;\; (\tfrac{1}{2}\mathbf{u}^\top\mathbf{\Psi}\mathbf{u})^{\frac{1}{2}} + d_{\mathcal{R}}(\mathbf{R}_i, \mathbf{R}_j), \qquad (3)$$

where $d_{\mathcal{R}}(\mathbf{R}_i,\mathbf{R}_j) = \mathrm{tr}\{\log^2\mathbf{\Lambda}(\mathbf{R}_i,\mathbf{R}_j)\}^{\frac{1}{2}}$ is the Riemannian metric for SPD matrices, and $\mathbf{\Lambda}(\mathbf{R}_i,\mathbf{R}_j) = \mathrm{diag}(\lambda_1,\ldots,\lambda_d)$ is the generalized eigenvalue matrix for the generalized eigenvalue problem.

The divergence *Div* between any two probability distributions, $P_1$ and $P_2$ say, has the following properties: $Div(P_1, P_2) \geq 0$, and $Div(P_1, P_2) = 0$ iff $P_1 = P_2$. Therefore, by definition, *Div* satisfies axioms (i), (ii), and (iii) of metrics, and in general, is not symmetric nor does it satisfy the triangle inequality (see footnote 3). However, for the above divergence measures, they are all are symmetric, and hence axiom (iv) is also satisfied. Unfortunately, the triangle inequality does not hold for the KL divergence $d_J$, nor does it for the Bhattacharyya distance $d_B$. However, for $d_H$ and $d_{J\mathcal{R}}$, they all satisfy the triangle inequality as well [1].

Although $d_{\mathbb{X}}$ can be any of the above divergence measures, it turns out that the metric properties for these measures are intimately related to the positive semi-definiteness of the affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ extracted from the data's neighbourhood graph adjacency matrix.

TABLE I
Data sets used in our experiments with their size $n$, number of features $d$, and number of classes $c$.

| Data Set | $n$ | $d$ | $c$ | Data Set | $n$ | $d$ | $c$ |
|---|---|---|---|---|---|---|---|
| Balance | 625 | 4 | 3 | Lymphography | 148 | 18 | 4 |
| Bupa | 345 | 6 | 2 | NewThyroid | 215 | 5 | 3 |
| Glass | 214 | 9 | 6 | Wine | 178 | 13 | 3 |
| Ionosphere | 351 | 33 | 2 | Corel | 500 | 36 | 5 |
| Letter | 20000 | 16 | 26 | SatImage | 6453 | 36 | 6 |
| Spam | 4601 | 57 | 2 | | | | |

In [1] we have studied the metric properties of these divergence measures and how they can impact the final embedding for two different manifold learning algorithms: classical multidimensional scaling (cMDS) and Laplacian eigenmaps (LEM) [3].

For LEM, for instance, the affinity matrix $\mathbf{A}$ is defined as $\mathbf{A}_{ij} = K(\mathcal{G}_i, \mathcal{G}_j)$, $\forall i, j$, where $K$ is a SPSD kernel. Since for LEM $\mathbf{A}$ has to be SPSD, then from Mercer theorem it is known that $\mathbf{A}$ will be SPSD *if and only if* $K$ is SPSD as well. If $K(\mathcal{G}_i,\mathcal{G}_j) = \exp\{-\frac{1}{\sigma}d_{\mathbb{X}}(\mathcal{G}_i,\mathcal{G}_j)\}$, where $\sigma > 0$ is a scale parameter, then it suffices for $K$ to be SPSD that $d_{\mathbb{X}}$ is a semi-metric (see footnote 3). Therefore for LEM, $d_{\mathbb{X}}$ can be $d_J$, $d_B$, $d_H$, or $d_{J\mathcal{R}}$. Note that LEM is different from cMDS for instance which requires $d_{\mathbb{X}}$ to be a metric such as $d_H$ or $d_{J\mathcal{R}}$ in order for $\mathbf{A}$ to be SPSD [1].

### *A. How does $d_{\mathbb{X}}$ affect the graph topology estimation?*

We will make a slight abuse of the notation and let: $d_{\mathbb{X}} \equiv d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j) = Div(\mathcal{G}_i, \mathcal{G}_j)$, to imply that querying the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ with respect to the space $\mathbb{X}$ returns the divergence between their respective local densities, where *Div* is $d_J$, $d_B$, $d_H$, and $d_{J\mathcal{R}}$. The expressions for *Div* in Equations(1), (2), and (3) are summations of two terms; the first term is for the difference in means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ weighted by a symmetric positive definite matrix, and the second term is for the discrepancy between $\mathbf{R}_i$ and $\mathbf{R}_j$. If $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j = \boldsymbol{\mu}$ (or $\boldsymbol{\mu}_i \approx \boldsymbol{\mu}_j$), the first term in (1), (2), and (3) will be zero (or very small), and $d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j)$ will be mainly determined by the dissimilarity in the covariances. If $\mathbf{R}_i = \mathbf{R}_j = \mathbf{R}$ (or $\mathbf{R}_i \approx \mathbf{R}_j$), the second term in (1), (2), and (3) will be zero (or very small) and $d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j)$ reduces to the Mahalanobis distance. Further, if $\mathbf{R}_i = \mathbf{R}_j = \mathbf{I}$, then $d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j)$ reduces to the Euclidean distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$.

Any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{D}$ (equivalently two nodes on the graph) will be close to each other, *if and only if* $\boldsymbol{\mu}_i \approx \boldsymbol{\mu}_j$ and $\mathbf{R}_i \approx \mathbf{R}_j$. That is, it is not sufficient that $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ is small. This new meaning for the distance between points is more restrictive and different from the Euclidean distance and the GQD, which are special cases from $d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j)$. Note that $d_{\mathbb{X}}(\mathbf{x}_i, \mathbf{x}_j)$ has an effect only on $\mathbf{x}_i$ and $\mathbf{x}_j$, but not on any other points in $\mathcal{D}$. This is due to the nature of local learning employed to learn $(\mathbb{X}, d_{\mathbb{X}})$, together with the nature of $d_{\mathbb{X}}$ as a divergence measure.

## IV. Experiments

We performed a series of experiments to test the validity and efficacy of the input space $(\mathbb{X}, d_{\mathbb{X}})$ on two different manifold learning methods; cMDS and LEM. Following various works on spectral clustering and dimensionality reduction [8], [11], we assess the benefit of $(\mathbb{X}, d_{\mathbb{X}})$ for manifold learning via the accuracy of $k$-Means clustering. More specifically, we quantify the impact of the new input space via comparing the average clustering accuracy for the data in the embedding space obtained before and after using $(\mathbb{X}, d_{\mathbb{X}})$. For each learning method, say LEM, and for each data set, we

TABLE II
CLUSTERING ACCURACY (WITH STANDARD DEVIATION) FOR $k$-MEANS IN THE ORIGINAL INPUT SPACE (EUC.) AND FOR THE EMBEDDINGS BY PCA, LEM, $(\mathbb{X}, d_{\mathbb{X}})$+LEM, CMDS, AND $(\mathbb{X}, d_{\mathbb{X}})$+CMDS.

| Data set | Euc. | PCA | LEM | $(\mathbb{X}, d_{\mathbb{X}})$+LEM |
|---|---|---|---|---|
| Balance | 51.1 (3.2) | 51.1 (4.4) | 56.8 (1.8) | **61.7** (4.5) |
| Bupa | 55.1 (0.1) | 55.3 (0.04) | 56.6 (0.1) | **66.6** (0.04) |
| Glass | 51.3 (3.2) | 52.8 (1.8) | 52.1 (3.1) | **56.4** (3.5) |
| Ionosphere | 70.6 (1.6) | 71.1 (0.1) | 69.8 (0.05) | **79.3** (2.6) |
| Lymphography | 47.4 (6.5) | 49.5 (5.3) | 54.9 (4.2) | **57.3** (5.47) |
| NewThyroid | 79.7 (8.9) | 79.9 (8.1) | 86.0 (1.4) | **91.1** (0.05) |
| Wine | 67.7 (5.1) | 68.5 (4.4) | 73.2 (0.5) | **92.1** (0.06) |
| Corel | 45.2 (3.6) | 45.9 (3.5) | 51.1 (3.01) | **64.9** (2.9) |
| Letter | 26.8 (0.7) | 26.8 (0.6) | 33.7 (2.4) | **38.4** (2.6) |
| SatImage | 62.7 (1.1) | 65.3 (1.1) | 72.2 (3.2) | **76.3** (2.7) |
| Spam | 63.5 (0.3) | 63.5 (0.3) | 69.2 (1.03) | **72.5** (4.5) |

| Data set | Euc. | PCA | cMDS | $(\mathbb{X}, d_{\mathbb{X}})$+cMDS |
|---|---|---|---|---|
| Balance | 51.1 (3.2) | 51.1 (4.4) | 53.2 (1.6) | **65.3** (1.8) |
| Bupa | 55.1 (0.1) | 55.3 (0.03) | 55.3 (0.01) | **59.7** (0.1) |
| Glass | 51.3 (3.2) | 52.8 (1.8) | 52.8 (2.5) | **55.9** (1.0) |
| Ionosphere | 70.6 (1.6) | 71.1 (0.1) | 71.1 (0.1) | **80.8** (0.4) |
| Lymphography | 47.4 (6.5) | 49.5 (5.3) | 49.6 (6.4) | **57.6** (8.6) |
| NewThyroid | 79.7 (8.9) | 79.9 (8.1) | 80.9 (7.2) | **94.4** (0.03) |
| Wine | 67.7 (5.1) | 68.5 (4.4) | 68.5 (4.4) | **89.8** (0.05) |
| Corel | 45.2 (3.6) | 45.9 (3.5) | 46.6 (2.9) | **59.4** (2.3) |
| Letter | 26.8 (0.7) | 26.8 (0.6) | 26.9 (0.7) | **35.6** (2.0) |
| SatImage | 62.7 (1.1) | 65.3 (1.1) | 64.4 (0.8) | **77.8** (0.5) |
| Spam | 63.5 (0.3) | 63.5 (0.3) | 63.5 (0.3) | **72.8** (2.3) |

tune the hyperparameters for the learning algorithm to maximize a specific clustering accuracy measure. Next, the input space $(\mathbb{X}, d_{\mathbb{X}})$ is combined with manifold learning—denoted $(\mathbb{X}, d_{\mathbb{X}})$+LEM—and for the same data set, the algorithm is optimized to yield an embedding that maximizes the same clustering accuracy measure. Our hypothesis is that the clustering accuracy in the embedding space obtained by $(\mathbb{X}, d_{\mathbb{X}})$+LEM and $(\mathbb{X}, d_{\mathbb{X}})$+cMDS will always be higher than the accuracies obtained by LEM and cMDS alone.

For the purpose of these experiments, we used twelve data sets, shown in Table (I), from the UCI Machine Learning Repository [12]. The experimental setup proceeded as follows. For each data set $\mathcal{D}$, we obtain different low dimensional embeddings using the following algorithms: (i) Principal component analysis (PCA). (ii) LEM with a Gaussian kernel. (iii) $(\mathbb{X}, d_{\mathbb{X}})$+LEM, (iv) cMDS, and (v) $(\mathbb{X}, d_{\mathbb{X}})$+cMDS, where $d_{\mathbb{X}} = \{d_J, d_B, d_H, d_{J\mathcal{R}}\}$. We used $k$–Means for clustering the data in each embedding space, and the number of clusters was assumed to be known. Since $k$–Means usually converges to local minima, the algorithm was run for 30 times with different initializations. For each run, the clustering accuracy was measured using the the Hungarian score and the final accuracies shown in Table (II) are the average Hungarian scores over the 30 different runs[4].

It can be seen from Table (II) that the clustering accuracy under $(\mathbb{X}, d_{\mathbb{X}})$, regardless of the divergence measure used, is significantly better than the accuracy under Euc., PCA, LEM, and cMDS. Note that in the context of current results, we had a fixed value for the regularization parameter $\gamma$, and we did not consider how to choose which divergence measure to use for a particular data set with LEM since this a question of model selection. Nevertheless, our results show that $(\mathbb{X}, d_{\mathbb{X}})$ helps the manifold learner to better characterize the latent class structure in the data. This also shows that there is still a room for improvement if we consider optimizing $\gamma$, or using more efficient regularized low rank covariance estimators.

## V. CONCLUDING REMARKS

We proposed an algorithmic framework for manifold learning algorithms that overcomes the liabilities of Euclidean geometry when dealing with real data sets. The framework integrates local learning with parametric density estimation to learn, in an unsupervised manner, the metric space $(\mathbb{X}, d_{\mathbb{X}})$ which becomes a pilot input space for manifold learning. $(\mathbb{X}, d_{\mathbb{X}})$ characterizes the varying sample density in the original input space $\mathcal{X}$, and reorganizes the proximity between points in $\mathcal{D} \subset \mathcal{X}$ based on $d_{\mathbb{X}}$ which now takes the varying sample density into consideration. The measure $d_{\mathbb{X}}$ conveys this new proximity information to the manifold learner, through the neighbourhood graph (and its adjacency matrix), to better characterize regions with high density (clusters). Future research work will consider the analysis of our approach based on the results of [6], [16], [14].

**Acknowledgment** We thank our Reviewers for pointing us to further references related to the analysis of our approach. This research was partially supported by FQRNT postdoctoral fellowship.

## REFERENCES

[1] K. Abou-Moustafa and F. Ferrie, "A note on metric properties of some divergence measures: The Gaussian case," *Proc. of the 4th Asian Conf. on Machine Learning, JMLR W&CP*, vol. 25, pp. 1–15, 2012.

[2] M. Balasubramanian, E. Schwartz, J. Tenenbaum, V. de Silva, and J. Langford, "The Isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, p. 7, 2002.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for data representation," *Neural Computation*, vol. 15, 2003.

[4] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Computation*, vol. 4, no. 6, pp. 888–900, 1992.

[5] M. Brand, "Charting a manifold," in *NIPS 15*, 2003.

[6] R. Coifman, S. Lafon, A. Lee, M. Maggioni, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," in *Proc. National Academy of Sciences*, vol. 102, no. 21, 2005, pp. 7426–7431.

[7] S. I. Daitch, J. A. Kelner, and D. A. Spielman, "Fitting a graph to vector data," in *Proc. of the 26th ICML*. ACM, 2009, pp. 201–208.

[8] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and K–means clustering," in *Proc. 24th ICML*, 2007.

[9] T. D. G. Shakhnarovich and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. The MIT Press, 2006.

[10] M. Gashler and T. Martinez, "Robust manifold learning with cyclecut," in *ACM Proc. of ICML*, 2011.

[11] U. v. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[12] D. Newman, S. Hettich, C. Blake, and C. Merz, "UCI Repository of Machine Learning Databases," 1998, `www.ics.uci.edu/~mlearn/MLRepository.html`.

[13] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[14] A. Singer and R. Coifman, "Non-linear independent component analysis with diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 25, no. 2, pp. 226 – 239, 2008.

[15] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, November 2000.

[16] D. Ting, L. Huang, and M. Jordan, "An analysis of the convergence of graph laplacians," in *Proc. 27th ICML*, 2010.

[17] L. van der Maaten and G. Hinton, "Visualizing data using t–SNE," *JMLR*, vol. 9, pp. 2579–2605, Nov. 2008.

[18] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbour classification," in *NIPS 18*, 2006.

[19] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side–information," in *NIPS 15*, 2002.

[4] The hyperparameters were tuned as follows. For PCA, the number of retained components constituted 98% of the total data variance. For LEM with a Gaussian kernel, there are two hyperparameters; $\sigma$ the kernel width, and the number of NNs used to construct the neighbourhood graph. Three values were considered for $\sigma$; from all the pairwise similarities distribution, we selected the median, the 0.25 and the 0.75 quantiles. The NNs was allowed to vary from 3 to 15. The dimensionality $d_0$ for LEM was fixed to the number of classes in the data [11]. For $(\mathbb{X}, d_{\mathbb{X}})$, $m$ the size of neighbourhoods varied from 3 to 15 NNs, and in all our experiments we used a regularized covariance of the form: $\mathbf{R}_i = \mathbf{\Sigma}_i + \gamma\mathbf{I}$, $\gamma = 1.0e-4$.