Out-of-sample generalizations for supervised manifold learning for classification

Elif Vural, Christine Guillemot

Out-of-sample generalizations for supervised manifold learning for classification

Elif Vural and Christine Guillemot

***Abstract*—Supervised manifold learning methods for data classification map data samples residing in a high-dimensional ambient space to a lower-dimensional domain in a structure-preserving way, while enhancing the separation between different classes in the learned embedding. Most nonlinear supervised manifold learning methods compute the embedding of the manifolds only at the initially available training points, while the generalization of the embedding to novel points, known as the out-of-sample extension problem in manifold learning, becomes especially important in classification applications. In this work, we propose a semi-supervised method for building an interpolation function that provides an out-of-sample extension for general supervised manifold learning algorithms studied in the context of classification. The proposed algorithm computes a radial basis function (RBF) interpolator that minimizes an objective function consisting of the total embedding error of unlabeled test samples, defined as their distance to the embeddings of the manifolds of their own class, as well as a regularization term that controls the smoothness of the interpolation function in a direction-dependent way. The class labels of test data and the interpolation function parameters are estimated jointly with a progressive procedure. Experimental results on face and object images demonstrate the potential of the proposed out-of-sample extension algorithm for the classification of manifold-modeled data sets.**

***Index Terms*—Manifold learning, supervised learning, out-of-sample extensions, pattern classification.**

I. Introduction

THE recovery of low-dimensional structures in data sets not only allows understanding the data but also provides useful representations for their treatment in several problems. Data classification is among the applications that benefit from the identification of low-dimensional structures in data. Unlike unsupervised manifold learning methods such as [1], [2], [3], which only take the geometric structure of data samples into account when learning a low-dimensional embedding, many recent supervised manifold learning methods seek a representation that not only preserves the manifold structure in each class, but also enhances the separation between different class-representative manifolds in the learned embedding. Meanwhile, an important problem in data classification with supervised manifold learning is the generalization of the learned embedding to novel data samples. In this work, we address the problem of constructing a continuous mapping between the high-dimensional original data space and the low-dimensional space of embedding for data classification applications.

Supervised manifold learning methods can be categorized into two groups as linear and nonlinear algorithms. Linear methods such as [4], [5], [6], [7], and [8] learn a linear projection that maps data into a lower-dimensional space such that the proximity of neighboring samples from the same class is preserved, while the distance between samples from different classes is increased. Nonlinear methods such as [9] have a similar classification-driven objective, while the new coordinates of data samples in the low-dimensional space are computed with a nonlinear learning process based on a graph representation of data. As linear dimensionality reduction methods compute a linear projection, they have the advantage that the generalization of the embedding for initially unavailable data samples is immediate and given by the learned linear operator. However, with linear methods samples from different classes are not linearly separable in the learned embedding, unless they are already linearly separable in the original high-dimensional space, which is rarely the case. The separation between different classes is an important factor that influences the performance of classification. Nonlinear dimensionality reduction methods achieve a better separation as a result of their relative flexibility in learning the coordinates. In fact, nonlinear methods such as [9], or nonlinear adaptations of the above linear methods, typically learn data representations where different classes become even linearly separable. However, one difficulty of using nonlinear methods is that they compute an embedding only in a pointwise manner, i.e., data coordinates in the low-dimensional domain are computed only for the initially available training data and are not generalizable to the test data in a straightforward way. Hence, an important issue that needs to be addressed in order to benefit from nonlinear manifold learning methods in classification is the generalization of the embedding to novel data samples.

The generalization of the learned embedding to new samples is referred to as the out-of-sample extension problem in manifold learning. Several previous works have addressed the out-of-sample problem. The study in [10] focuses on the extension of manifold learning methods that compute data coordinates in the form of the eigenvectors of a data kernel matrix. It is shown that in such a setting the Nyström method can be used to compute eigenfunctions that coincide with the eigenvectors on the training samples and generalize them to the continuous domain. In fact, the out-of-sample extension with the Nyström formula as proposed in [10] can also be derived from the kernel ridge regression framework, by removing the regularization term and imposing the constraint that the data coordinates of training samples be given by the eigenvectors of the data kernel matrix [11]. Next, several out-of-sample extension algorithms rely on the construction of an interpolation function between the high- and low-dimensional domains. Some families of interpolation functions used in manifold learning extensions

Elif Vural and Christine Guillemot are with Centre de recherche INRIA Rennes - Bretagne Atlantique, France (elif.vural@inria.fr, christine.guillemot@inria.fr).

are polynomials [12], sparse linear combinations of functions in a reproducing kernel Hilbert space (RKHS) [11], and sparse grid functions [13]. In [14], the out-of-sample extension of general manifold learning methods is achieved by computing a local projection of the high-dimensional space to the low-dimensional domain with a similarity transformation of the local PCA bases. There are also some extension methods designed for particular manifold learning algorithms. The study in [15] proposes an out-of-sample generalization of the multidimensional scaling (MDS) method, which is based on an interpretation of MDS as a least squares problem. Similarly, the method proposed in [16] presents a generalization for maximum variance unfolding [17].

Meanwhile, all of the above methods address the out-of-sample extension problem in an unsupervised setting, i.e., no class label information of input data samples is used. In a classification problem, on the other hand, different classes are often assumed to lie on different manifolds, e.g., in a face recognition problem, the face images of each individual form a different manifold, and supervised manifold learning methods map these class-representative manifolds to different manifolds in the low-dimensional domain. Therefore, class labels of data samples and the fact that different classes are concentrated around different low-dimensional structures should be taken into account when constructing an out-of-sample extension for classification applications. Besides this, many of the above unsupervised extension methods are even not applicable in the supervised setting. For instance, the popular Nyström extension [10] considers embeddings given by the eigenvectors of a symmetric kernel matrix. Then, in order to embed a novel point, the kernel is evaluated between the novel point and each training point. Meanwhile, in supervised manifold learning, the value of the kernel depends not only on data sample pairs but also on the class labels of the samples. The kernel usually takes positive values for sample pairs from the same class and negative values for those from different classes, e.g., as in [9]. Hence, the Nyström method does not have a straightforward generalization for supervised manifold learning.

In this paper, we propose a method for constructing out-of-sample generalizations of supervised manifold learning algorithms for classification. In order to extend the embedding (learned with any supervised algorithm) to novel points, we compute a radial basis function (RBF) interpolation function from the high-dimensional space to the low-dimensional one. We optimize the parameters of the interpolation function such that it maps initially unavailable test points as close as possible to the embeddings of the manifolds of their own class in the low-dimensional domain. This is achieved with a progressive estimation of the class labels of test points while gradually updating the parameters of the interpolation function at the same time. As the proposed method makes use of test points in the construction of the interpolation function, it can be considered as a semi-supervised solution for obtaining an out-of-sample extension. Another criterion that is taken into account in the optimization of the parameters of the interpolation function is the regularity of the interpolation function. We find that the regularity of the interpolation function can be adjusted by optimizing its scale parameters to minimize a regularization objective, which controls the magnitude of the interpolation function gradient, while allowing sharp directional derivatives to occur only along the class separation boundaries in order to attain an effective separation between different classes. Experimentation on several image data sets shows that the proposed method can be effectively used in the classification of data of intrinsic low dimension. The proposed out-of-sample extension method is general and can be coupled with any supervised manifold learning algorithm.

The rest of the paper is organized as follows. In Section II we briefly overview some supervised manifold learning methods and formulate the out-of-sample extension problem. In Section III we describe the proposed method for classification-driven out-of-sample extensions for supervised manifold learning. In Section IV we discuss some aspects of the proposed algorithm, where we analyze its complexity and interpret it within the context of regression. In Section V we present some experimental results and in Section VI we conclude.

II. Overview of manifold learning

A. Manifold learning for classification

Given a set of data samples $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$ that reside in a high-dimensional space $\mathbb{R}^n$, manifold learning computes a new representation $y_i \in \mathbb{R}^d$ in a lower-dimensional domain $\mathbb{R}^d$ for each data sample x_i. Manifold learning methods generally assume that the samples $\{x_i\}$ come from a model of low intrinsic dimension and search for an embedding that significantly reduces the dimension of the data ($d \ll n$) while preserving certain geometric properties. Different methods target different objectives in computing the embedding. The ISOMAP method computes an embedding such that Euclidean distances in the low-dimensional domain are proportional to the geodesic distances in the original domain [1], while LLE looks for an embedding that preserves local reconstruction weights of data samples in the original domain [2]. The Laplacian eigenmaps algorithm [3] first constructs a graph from the data samples where nearest neighbors are typically connected with an edge. The graph Laplacian matrix is given by $L = D - W$, where W is the $N \times N$ weight matrix whose entries are usually computed based on a kernel $W_{ij} = K(x_i, x_j)$, and D is a diagonal degree matrix given by $D_{ii} = \sum_j W_{ij}$. The embedding with Laplacian eigenmaps is then learned by solving

$$\min_{Y \in \mathbb{R}^{N \times d}} \operatorname{tr}(Y^T L Y) \quad \text{s.t.} \quad Y^T D Y = I$$

where I is the identity matrix. The solution to this problem is given by the d eigenvectors corresponding to the smallest nonzero eigenvalues of the generalized eigenvalue problem $Lz = \lambda Dz$, where the coordinate vector y_i for each data sample x_i is given by the i-th row of Y. Intuitively, such an embedding seeks data coordinates that have a slow variation on the data graph, i.e., two neighboring points on the graph are mapped to nearby coordinates. There exist linear versions of the Laplacian eigenmaps method as well. The above problem is solved under the constraint that Y be given by a linear projection of X onto $\mathbb{R}^d$ in [18], which is applied to face recognitions problems in [19] and [20].

Recently, many extensions have been proposed for manifold learning for classification. Most of these methods are supervised adaptations of the Laplacian eigenmaps algorithm. In order to achieve a good separation between the classes, an embedding is sought where data coordinates vary slowly between neighboring samples of the same class and change rapidly between neighboring samples of different classes. The algorithm proposed in [9] formalizes this idea by defining two graphs that respectively capture the within-class and between-class neighborhoods. Denoting the weight matrices of these two graphs by W_w and W_b, and the corresponding Laplacians by L_w and L_b, the method seeks an embedding that solves

$$\min_{Y\in\mathbb{R}^{N\times d}} \operatorname{tr}(Y^T L_w Y) - \mu \operatorname{tr}(Y^T L_b Y) \quad \text{s.t.} \quad Y^T D_w Y = I \tag{1}$$

where $\mu > 0$. The method proposed in [21] employs an alternative Fisher-like formulation for the supervised manifold learning problem where the embedding is obtained by solving

$$\max_z \frac{z^T L_b z}{z^T L_w z}. \tag{2}$$

However, the problem is solved under the constraint $z^T = v^T X$ in order to obtain a linear embedding, where $X = [x_1 \dots x_N]$ is the $n \times N$ data matrix and $v \in \mathbb{R}^{n\times 1}$ defines a projection. Variations over this formulation can be found in several other works such as [4], [5], [6], [7], [22] and [8].

B. Out-of-sample extensions

While most manifold learning methods learn the coordinates of only initially available data samples, in many applications including classification, the generalization of the learned embedding to the whole data space is an important problem. Given a set of data samples $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$ in the high-dimensional ambient space and their corresponding coordinates $\{y_i\}_{i=1}^N \subset \mathbb{R}^d$ in a low-dimensional space, the out-of-sample extension problem consists of constructing a mapping $f : \mathbb{R}^n \to \mathbb{R}^d$ such that f gives the learned embedding $f(x_i) = y_i$ on the available data samples while generalizing the embedding to all points in $\mathbb{R}^n$.

A popular out-of-sample generalization algorithm is presented in [10], based on the Nyström formula. This method proposes a generalization for manifold learning algorithms that compute the coordinates based on an eigenvalue problem $My^k = \lambda_k y^k$, where the symmetric matrix M is given by a data-dependent kernel $M_{ij} = \tilde{M}(x_i, x_j)$ and y^k is the kth eigenvector of M, which defines the kth dimension of the data coordinates. The exact expression of the kernel matrix M as a function of the weight matrix W depends on the manifold learning algorithm to be generalized, as different algorithms target different objectives. The out-of-sample extension proposed in [10] is then given by the function $f(x) = [f^1(x) \dots f^d(x)]$, where

$$f^k(x) = \frac{1}{\lambda_k} \sum_{i=1}^{N} y_i^k \tilde{M}(x, x_i) \tag{3}$$

and $y_i = [y_i^1 \dots y_i^d]$ are the coordinates of the embedding of x_i in $\mathbb{R}^d$. This defines a continuous function that coincides with the embedding at the initially available points $f(x_i) = y_i$. While this popular method provides straightforward generalizations of many manifold learning algorithms such as ISOMAP, LLE, and Laplacian eigenmaps, it cannot be used with most supervised manifold learning methods. The reason is that, although the data kernel matrix M is assumed to be a general symmetric matrix (not necessarily positive semi-definite) in [10], the entries of this matrix in supervised methods are not only dependent on the data samples x_i, but also on their class labels. For instance, in (1), the kernel matrix M is a normalized version of the matrix $L_w - \mu L_b$, which is determined with respect to data class labels. In this case, the Nyström formula (3) cannot be applied as $\tilde{M}(x, x_i)$ is not priorly known for a test sample of unknown class.

Several out-of-sample extension methods such as those based on fitting a particular type of interpolation function as in [11], [12], and [13] can be applied for generalizing supervised embeddings by fitting a function f to the priorly learned (x_i, y_i) pairs. However, as this gives a generalized embedding based only on an approximation objective that does not take into account the class information of data, its classification performance is likely to be suboptimal.

In this paper, we propose to learn an interpolation function in an application-aware manner. The proposed method not only makes use of the initially available training samples (x_i, y_i), but also exploits the test samples of unknown class in the learning, by jointly estimating the interpolation function parameters and the class labels of test samples. We describe this method in Section III.

III. Out-of-sample extensions for classification

A. Formulation of the out-of-sample problem

We begin with a formalization of the classification-based out-of-sample extension problem. Let $\mathcal{M}_1, \mathcal{M}_2, \dots \mathcal{M}_M \subset \mathbb{R}^n$ be M compact manifolds representing M different classes in the original ambient space $\mathbb{R}^n$. Let $\mathcal{E}$ be an embedding of the manifolds $\{\mathcal{M}_m\}$ in a lower-dimensional space $\mathbb{R}^d$

$$\mathcal{E} : \bigcup_m \mathcal{M}_m \to \mathbb{R}^d$$

such that each manifold $\mathcal{M}_m \subset \mathbb{R}^n$ is mapped to $\mathcal{E}(\mathcal{M}_m) \subset \mathbb{R}^d$. The restriction of $\mathcal{E}$ to each manifold is assumed to be continuous and the embeddings of different manifolds are assumed to be disjoint. We consider that the data samples are drawn from a probability measure ν on $\mathbb{R}^n$ such that the samples of each class m are concentrated around the manifold $\mathcal{M}_m$. Let ν_m denote the probability measure of class m having a support region S_m in $\mathbb{R}^n$, where $\mathcal{M}_m \subset S_m$. We denote by $P_{\mathcal{M}_m}(x)$ a projection of the point x onto the manifold $\mathcal{M}_m$, which is a point on $\mathcal{M}_m$ of minimal distance to x

$$\|x - P_{\mathcal{M}_m}(x)\| = \min_{x' \in \mathcal{M}_m} \|x - x'\|.$$

Here $\|\cdot\|$ denotes the usual ℓ_2-norm in the Euclidean space.

As for the solution set of interpolation functions, we consider a compact set $\mathcal{H}$ of differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$, where $f : \mathbb{R}^n \to \mathbb{R}^d$ belongs to $\mathcal{H}^d$ given by the d-dimensional Cartesian product of $\mathcal{H}$. An interpolation function that is suitable for classification should map points x from

class m as close as possible to the set $\mathcal{E}(\mathcal{M}_m)$, so that they can be correctly classified with respect to their representation in $\mathbb{R}^d$. We thus define the embedding error of f with respect to its deviation from the embedding of the projection of a point onto the manifold of its own class. The total embedding error of a function f over all classes is then given by

$$E(f) := \sum_m \int_{S_m} \|f(x) - \mathcal{E}\left(P_{\mathcal{M}_m}(x)\right)\|^2 \, d\nu_m(x). \qquad (4)$$

The distributions ν_m are usually not explicitly known in practice. In order to avoid overfitting to training data, it is useful to enforce some regularity properties for the interpolation function f. A smoothness constraint can be imposed by controlling the total gradient magnitude. Meanwhile, nonlinear supervised manifold learning methods, whose extensions are targeted in this paper, typically learn a representation where different classes are likely to become linearly separable in the learned embedding. The coordinates defining the embedding are orthogonal when given by the eigenvectors of a symmetric kernel, or "nearly orthogonal" when given by the generalized eigenvalue problem (1). Different groups of classes are then expected to become separable along different dimensions of the learned embedding, which is also easy to confirm experimentally (see, e.g., Figure 3(a)). Thus, when learning an interpolation function f, in order to enhance the separation between different classes, it is desirable to have sufficiently strong derivatives along the directions defining the boundaries of the distributions of different classes in the ambient space, especially for the components f^k of f for which the considered classes are separable at dimension k of $\mathbb{R}^d$. This is illustrated in Figure 1. Given a dimension $k \in \{1, \ldots, d\}$, let

$$I^k = \{(m,p) : \max \mathcal{E}^k(\mathcal{M}_m) < \min \mathcal{E}^k(\mathcal{M}_p) \\ \text{or } \max \mathcal{E}^k(\mathcal{M}_p) < \min \mathcal{E}^k(\mathcal{M}_m)\}$$

denote the set of indices of manifold pairs whose embeddings are separable at dimension k, where $\mathcal{E}^k(\mathcal{M}_m)$ denotes the kth dimension of the embedding $\mathcal{E}(\mathcal{M}_m)$. Let $\nabla_v f^k$ denote the directional derivative of f^k along the direction v. For a point x from class m, let

$$u_p(x) := \frac{x - P_{\mathcal{M}_p}(x)}{\|x - P_{\mathcal{M}_p}(x)\|}$$

denote the unit vector corresponding to the direction of projection of x onto the manifold $\mathcal{M}_p$ of class p, where $p \neq m$. Then, we would like to learn an interpolation function f such that the directions along which f^k has the strongest derivatives coincide with the directions of the projections of points onto the manifolds of other classes. The total derivative magnitude along the directions of projection onto other classes, normalized by the average derivative magnitude is given by

$$D(f^k) := \sum_{(m,p)\in I^k} \int_{S_m} \frac{\left\|\nabla_{u_p(x)} f^k(x)\right\|}{\mathbb{E}_v \|\nabla_v f^k(x)\|} \, d\nu_m(x) \qquad (5)$$

where f^k is assumed to have nowhere vanishing gradient and $\mathbb{E}_v\|\nabla_v f^k(x)\|$ denotes the mean directional derivative magnitude, induced from the overall distribution of data over all classes.[1] The normalization of the directional derivative by the average derivative aims to measure the derivative magnitude along separation boundaries relatively to the mean derivative magnitude.

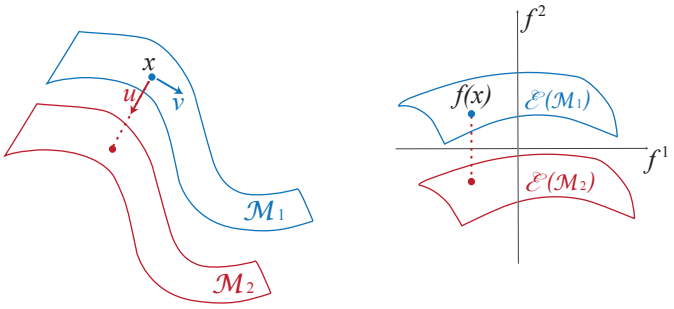


Fig. 1. Illustration of supervised manifold learning and out-of-sample interpolation. Manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ representing two different classes are embedded in a lower-dimensional domain such that they are separable along dimension $k = 2$, but not along dimension $k = 1$. The second component $f^2(x)$ of the interpolation function $f(x) = [f^1(x)\, f^2(x) \ldots f^k(x)]$ should then have a sufficiently strong directional derivative $\nabla_u f^2(x)$ along direction u at x, in order to reinforce the separation achieved by the supervised embedding, while it should vary smoothy along direction v. Meanwhile, the first component $f^1(x)$ of the interpolation function should have a slow variation along both directions u and v as the embeddings $\mathcal{E}(\mathcal{M}_1)$ and $\mathcal{E}(\mathcal{M}_2)$ are not separable along dimension $k = 1$.

While the presence of sufficiently strong directional derivatives along separation boundaries is expected to enhance the separation between classes with the learned function, it is useful to control the smoothness of the interpolation function by preventing it from attaining arbitrarily high gradient magnitudes. We thus define the total gradient magnitude

$$G(f^k) := \sum_m \int_{S_m} \frac{\|\nabla f^k(x)\|}{\mathbb{E}_v \|\nabla_v f^k(x)\|} \, d\nu_m(x) \qquad (6)$$

which is also normalized by the average directional derivative so that it is comparable to the term in (5).

From (5) and (6), one can define an overall regularization objective R that increases with the total gradient magnitude G and decreases with the directional derivative magnitude along separation boundaries D. One way to define the regularization objective R is as

$$R(f) = \sum_{k=1}^{d} \left(G(f^k) - \lambda D(f^k)\right)$$

where $\lambda > 0$.

Finally, combining the embedding error in (4) and the above regularization term, we formulate the search of the

[1] The mean directional derivative can be formally defined as follows. Let p_ν denote the probability density function corresponding to the probability measure ν, and let $S_t(x) \subset \mathbb{R}^n$ denote the n-dimensional sphere of radius t centered at x. The expected value of the directional derivative of f^k at x is then given by

$$\mathbb{E}_v \|\nabla_v f^k(x)\| = \lim_{t\to 0} \frac{\int_{S_t(x)} p_\nu(x+tv)\, \|\nabla_v f^k(x)\| dS}{\int_{S_t(x)} p_\nu(x+tv)\, dS}$$

where v denotes the unit surface normal in the direction of the surface element dS.

interpolation function f as the optimization of the following problem

$$f = \arg\min_{h\in\mathcal{H}^k} \Big(\sum_m \int_{S_m} \|h(x) - \mathcal{E}\left(P_{\mathcal{M}_m}(x)\right)\|^2 \, d\nu_m(x) + \alpha R(h) \Big) \tag{7}$$

where $\alpha > 0$. A solution to the above problem exists as $\mathcal{H}^k$ is compact and the objective function is continuous.

In a real setting, the distributions ν_m of data are often not explicitly known and one has access to a set of samples $\mathcal{X} = \{x_i\}_{i=1}^Q$ drawn from these distributions. Let $C_i \in \{1, \ldots, M\}$ denote the class label of the data sample x_i, and $\mathcal{N}(x_i)$ be the set of nearest neighbors of x_i in $\mathcal{X}$ (which can be chosen for instance as the K-nearest neighbors of x_i with respect to the Euclidean distance in $\mathbb{R}^n$). Let

$$\overline{n}(x_i) = \left\{ \frac{x_i - x_j}{\|x_i - x_j\|} : x_j \in \mathcal{N}(x_i) \right\}$$

denote the set of unit vectors that indicate the directions of the neighbors of x_i, and

$$\overline{n}_p(x_i) = \left\{ \frac{x_i - x_j}{\|x_i - x_j\|} : x_j \in \mathcal{N}(x_i), C_j = p \right\}$$

denote the set of unit directions given by the nearest neighbors of x_i within class p. We can then define the empirical embedding error $\hat{E}(f)$ as

$$\hat{E}(f) = \sum_m \sum_{i:\, C_i = m} \|f(x_i) - \mathcal{E}\left(P_{\mathcal{M}_m}(x_i)\right)\|^2$$

and the empirical counterpart $\hat{R}(f)$ of the regularization term $R(f)$ as

$$\hat{R}(f) = \sum_{k=1}^d \left(\hat{G}(f^k) - \lambda \hat{D}(f^k) \right) \tag{8}$$

where

$$\hat{G}(f^k) := \sum_i \frac{\|\nabla f^k(x_i)\|}{|\overline{n}(x_i)|^{-1} \sum_{v\in\overline{n}(x_i)} \|\nabla_v f^k(x_i)\|} \tag{9}$$

$$\hat{D}(f^k) := \sum_{(m,p)\in I^k} \sum_{i:\, C_i = m} \frac{1}{|\overline{n}_p(x_i)|} \sum_{u\in\overline{n}_p(x_i)} \frac{\left\|\nabla_u f^k(x_i)\right\|}{|\overline{n}(x_i)|^{-1} \sum_{v\in\overline{n}(x_i)} \|\nabla_v f^k(x_i)\|}. \tag{10}$$

In the above expressions, $|\cdot|$ denotes the cardinality of a set, and the mean directional derivative $\mathbb{E}_v \|\nabla_v f^k(x_i)\|$ is approximated by the average derivative of $f^k(x_i)$ along the directions of the neighbors $\overline{n}(x_i)$ of x_i. In the definition of $\hat{D}(f^k)$, we approximate the derivative $\nabla_{u_p(x_i)} f^k(x_i)$ along the direction of the projection of x_i onto $\mathcal{M}_p$ with the average derivative along the directions of the nearest neighbors of x_i within class p, where $\overline{n}_p(x_i)$ is assumed to be non-empty for all x_i and p.

Now, having defined the objective function in the empirical setting, we come back to the actual manifold learning problem. In practice, the manifolds $\mathcal{M}_m$ are usually not explicitly known, and manifold learning methods compute an embedding for only the initially available training samples. Let us denote by $\mathcal{X}_T = \{x_i\}_{i=1}^N \subset \mathcal{X}$ the set of training samples with known class labels (where $N \leq Q$), for which an embedding $\mathcal{Y}_T = \{y_i\}_{i=1}^N$ is priorly computed with a supervised manifold learning algorithm. We assume that there exist embeddings $\mathcal{E}(\mathcal{M}_m)$ of the manifolds $\mathcal{M}_m$ such that the embeddings $\{y_i\}$ of the samples of each class m are concentrated around $\mathcal{E}(\mathcal{M}_m)$. Although the training samples available in practice are not guaranteed to lie exactly on a manifold in general (due to noise, imprecise measurements, or several sources of deviation from the assumed model), we make the following approximations for a sample $x_i \in \mathcal{X}_T$ of class m for the simplicity of computations:

$$P_{\mathcal{M}_m}(x_i) \approx x_i, \qquad \mathcal{E}(P_{\mathcal{M}_m}(x_i)) \approx y_i.$$

The embedding error $\hat{E}(f)$ can then be decomposed as

$$\hat{E}(f) = \hat{E}_T(f) + \hat{E}_O(f)$$

where

$$\hat{E}_T(f) = \sum_{i=1}^N \|f(x_i) - y_i\|^2$$

is the embedding error of the training samples $\mathcal{X}_T$ and

$$\hat{E}_O(f) = \sum_{i=N+1}^Q \|f(x_i) - \mathcal{E}\left(P_{\mathcal{M}_m}(x_i)\right)\|^2 = \sum_m \sum_{\substack{i=N+1 \\ C_i = m}}^Q \|f(x_i) - \mathcal{E}\left(P_{\mathcal{M}_m}(x_i)\right)\|^2$$

is the embedding error of the other samples than the training samples (test samples in $\mathcal{X} \setminus \mathcal{X}_T$).

In the generalization of an embedding, one may wish to strictly preserve the learned coordinates of the training data $f(x_i) = y_i$. We can thus formulate the out-of-sample extension problem for supervised manifold learning as follows:

$$f = \arg\min_{h\in\mathcal{H}^k} \hat{E}_O(h) + \alpha \hat{R}(h) \quad \text{s.t.} \quad \hat{E}_T(h) = 0. \tag{11}$$

In the above problem, if there are observations in $\mathcal{X}$ with unknown class labels, one needs to estimate the class labels C_i for $N < i \leq Q$. In the rest the paper, we focus on this general case. In Section III-B, we describe an algorithm that computes an interpolation function with a joint and progressive estimation of the function parameters and the class labels of data.

B. Construction of the interpolation function

In this study, we select the set $\mathcal{H}$ of interpolation functions for the out-of-sample extension problem as the radial basis functions (RBFs)

$$\mathcal{H} = \left\{ g : g(x) = \sum_{l=1}^L c_l \, \phi\left(\frac{\|x - a_l\|}{\sigma_l} \right) \right\}$$

where $\phi : \mathbb{R} \to \mathbb{R}^+$ is a differentiable kernel. The coefficients c_l, the kernel centers a_l, and the scale parameters σ_l are assumed to lie in some compact domains in $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{R}^+$,

respectively. The Gaussian function $\phi(t) = e^{-t^2}$ is a common choice for the RBF kernel due to its desirable properties such as its smoothness and rapid decay, which we also adopt in this work.

In our problem, we look for a function $f = [f^1(x) \dots f^d(x)] : \mathbb{R}^n \to \mathbb{R}^d$ such that each dimension f^k of f is given by

$$f^k(x) = \sum_{l=1}^{L} c_l^k \, \phi\left(\frac{\|x - a_l^k\|}{\sigma_l^k}\right). \tag{12}$$

The construction of the interpolation function f is thus equivalent to the determination of the parameters c_l^k, a_l^k, σ_l^k, and the number of terms L.

In the optimization problem in (11), the evaluation of $\hat{E}_O(h)$ requires the knowledge of the class labels C_i of x_i for $i = N+1, \dots, Q$, which are unavailable in the beginning. We propose to solve this problem with an iterative algorithm that progressively estimates the class labels and constructs a sequence of interpolation functions $f_1, \dots, f_r, \dots f_R$ in an alternating scheme as described below.

In iteration r of the algorithm, we construct a function f_r with L_r terms. When fitting an RBF interpolation function to data, it is common practice to assign kernel centers as data points. In iteration r, we select the kernel centers $a_l^k = x_{r_l}$ as a subset of data samples $\{x_{r_l}\}_{l=1}^{L_r} \subset \mathcal{X}$, where the index sequence $\{r_l\}_{l=1}^{L_r}$ depends on the iteration r and denotes the indices of the data samples $\{x_i\}$ chosen as kernel centers. Throughout the iterations, the number of terms L_r is increased gradually such that $N = L_1 < L_2 < ... < L_R = Q$. Once the kernel centers a_l^k are fixed, the interpolation function f_r in iteration r, characterized by the coefficients $\{c_l^k\}$ and the scale parameters $\{\sigma_l^k\}$, $l = 1, \dots, L_r$, $k = 1, \dots, d$, is obtained by solving the problem

$$\min_{\{c_l^k\} \subset B,\, \{\sigma_l^k\} \subset \Lambda} \hat{E}_O^r(f) + \alpha \hat{R}(f) \quad \text{s.t.} \quad \hat{E}_T(f) = 0 \tag{13}$$

where $B \subset \mathbb{R}$ and $\Lambda \subset \mathbb{R}^+$ are compact parameter domains (sufficiently large so that the constraint $\hat{E}_T(f) = 0$ can be satisfied) and

$$\hat{E}_O^r(f) = \sum_{m} \sum_{\substack{l=N+1 \\ C_{r_l} = m}}^{L_r} \|f(x_{r_l}) - \mathcal{E}\left(P_{\mathcal{M}_m}(x_{r_l})\right)\|^2. \tag{14}$$

The problem (13) has a solution as a continuous function over a compact domain attains its minimum.

Before discussing the solution of (13), we first give an overview of the method. In iteration r, once the interpolation function f_r is computed by solving (13), we estimate the class label of each point x_i for $N+1 \le i \le Q$ by assigning it the class label of the training point x_j such that $f_r(x_j)$ is the closest to $f_r(x_i)$ in the low-dimensional domain $\mathbb{R}^d$:

$$C_i = C_j : \quad j = \arg\min_q \|f_r(x_q) - f_r(x_i)\|, \ 1 \le q \le N. \tag{15}$$

At the same time, a confidence score μ_i is assigned to each estimate C_i by comparing the distance of x_i to its nearest neighbor x_j within all classes and to its nearest neighbor $x_{j'}$ among the classes other than C_j:

$$\mu_i = \frac{\|f(x_{j'}) - f(x_i)\|}{\|f(x_j) - f(x_i)\|} : \\ j' = \arg\min_q \|f_r(x_q) - f_r(x_i)\|, \ 1 \le q \le N, C_n \ne C_j. \tag{16}$$

The confidence score μ_i thus decreases with the "ambiguity" in assigning x_i the class label C_i with respect to the nearest-neighbor decision rule in $\mathbb{R}^d$ via f_r.

The confidence scores μ_i obtained in an iteration are then used in the next iteration for the selection of the kernel centers. In iteration r, the kernel centers are determined based on the confidence scores computed in the previous iteration $r-1$ as follows. The first N kernel centers $\{a_l^k\}_{l=1}^N = \{x_{r_l}\}_{l=1}^N$ consist of the training set $\mathcal{X}_T$, i.e., $r_l = l$ for $l = 1, \dots, N$. The remaining kernel centers $\{a_l^k\}_{l=N+1}^{L_r}$ are then set as the first $L_r - N$ points in $\mathcal{X} \setminus \mathcal{X}_T$ of highest confidence scores. The alternating stages of computing f_r and estimating the class labels C_i and obtaining the confidence scores μ_i are repeated until the last iteration R, where all data samples are included in the set of kernel centers $\{a_l^k\}_{l=1}^Q = \mathcal{X}$. The interpolation function f is then given by f_R, and the class labels of the points in $\mathcal{X}$ are obtained by estimating them with the final interpolation function with respect to (15).

We now discuss the solution of the problem (13). First, observe that for any L_r input data pairs $(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}^d$ and any choice of the scale parameters σ_l^k, one can find interpolation functions f^k of $L = L_r$ terms that satisfy $f(x_i) = y_i$ as follows. Setting $a_l^k = x_l$ for $l = 1, \dots, L_r$, the constraints $f^k(x_i) = y_i^k$ yield the linear system

$$\Phi^k c^k = y^k \tag{17}$$

where $c^k = [c_1^k \dots c_{L_r}^k]^T$ is the coefficient vector, $y^k = [y_1^k \dots y_{L_r}^k]^T$ consists of the kth dimensions of $\{y_i\}$, and

$$\Phi_{il}^k = \phi\left(\frac{\|x_i - x_l\|}{\sigma_l^k}\right) \tag{18}$$

is the matrix of RBFs evaluated at data points x_i. The square matrix Φ^k is invertible if the points x_i are distinct and ϕ is chosen as the Gaussian kernel [23]. The system (17) then has a unique solution $c^k = (\Phi^k)^{-1} y^k$, which satisfies $f^k(x_i) = y_i^k$.

In iteration $r = 1$, we have $L_1 = N$ and all kernel centers are training points. In this case the embedding error in (14) is $\hat{E}_O^1(f) = 0$, and the optimization problem is reduced to

$$\min_{\{c_l^k\} \subset B,\, \{\sigma_l^k\} \subset \Lambda} \hat{R}(f) \quad \text{s.t.} \quad \hat{E}_T(f) = 0.$$

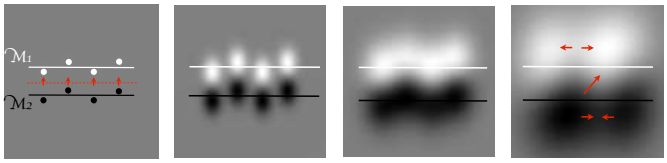
Due to the above discussion, the constraint $\hat{E}_T(f) = 0$ can be satisfied for any choice of scale parameters σ_l^k by setting the coefficients as $c^k = (\Phi^k)^{-1} y^k$. This reduces the problem to the minimization of the regularization term $\hat{R}(f)$ by optimizing the scale parameters $\{\sigma_l^k\}$ under the constraint $c^k = (\Phi^k)^{-1} y^k$

$$\min_{\substack{\{\sigma_l^k\} \subset \Lambda \\ c^k = (\Phi^k)^{-1} y^k}} \hat{R}(f) \tag{19}$$

where the summations in the terms (9) and (10) of $\hat{R}(f)$ run over the set of training samples $\mathcal{X}_T$. The regularization term is a non-convex function of the scale parameters $\{\sigma_l^k\}$ with numerous extrema. Meanwhile, we have experimentally observed that the variation of $\hat{R}(f)$ with σ^k is quite regular when all scale parameters σ_l^k, $l = 1, \ldots, L_1$, in each dimension k are set to a common value σ^k. Moreover, setting all scale parameters to the same value across each dimension also simplifies the optimization problem, as it reduces the number of optimization variables from $L_1 k$ to k. We thus propose to solve the problem (19) under the constraint $\sigma_l^k = \sigma^k$ for $l = 1, \ldots, L_1$. Since the form of $\hat{R}(f)$ in (8) is decomposable into its components in different dimensions, the scale parameter of dimension k is given by

$$\min_{\substack{\sigma^k \in \Lambda \\ c^k = (\Phi^k)^{-1} y^k}} \left(\hat{G}(f^k) - \lambda \hat{D}(f^k) \right).$$

It is difficult to analyze the above function theoretically. Meanwhile, in practice we have observed that $\hat{G}(f^k)$ increases with σ^k monotonically. Moreover, if the underlying embedding obtained with supervised manifold learning provides a "balanced" distribution of the classes across different dimensions while ensuring a sufficient separation, the total directional derivative along class separation boundaries $\hat{D}(f^k)$ first increases at a fast rate with σ^k at small scale values, and then it stagnates or the rate of increase is highly reduced. This is due to the fact that, when the scale parameters are too small, the interpolation function is too localized around kernel centers and its support does not cover well the whole space. Then, it does not have sufficiently strong derivatives along class separation boundaries. As σ^k increases, there typically exists a range for σ^k where the directional derivatives along class separation boundaries are relatively stronger than those along other directions, thanks to the underlying learned embedding that separates different classes and guides the interpolation function via the condition $f^k(x_i) = y_i^k$ imposed on training samples. This range for the scale parameters coincides in general with the interval of scale parameters where a good classification performance is attained. If the scale parameters are increased beyond this range, the gradient of the function f^k increases too much, resulting in an overfitting of the interpolation function, where the advantage of having sufficiently strong directional derivatives along class separation boundaries is lost as strong derivatives appear in other directions as well due to overfitting. This is illustrated with a simple example in Figure 2. Figure 2(a) shows two manifolds $\mathcal{M}_1, \mathcal{M}_2 \subset \mathbb{R}^2$ representing two different classes, and four training samples selected from the distribution concentrated around each manifold. Let us consider a one-dimensional embedding of the manifold samples in $\mathbb{R}$ such that samples from $\mathcal{M}_1$ and $\mathcal{M}_2$ are mapped respectively to 1 and -1. An ideal interpolation function $f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}$ separating the two classes well in $\mathbb{R}$ should have gradients in the directions shown in red in Figure 2(a), which are orthogonal to the class separation boundary. In Figures 2(b)-2(d), an RBF interpolation function f with Gaussian kernel is fitted to the training data, and $f(x)$ is plotted over the displayed region of $\mathbb{R}^2$, where white and black colors correspond respectively to 1 and -1. The scale parameter is chosen as $\sigma = 0.5$, $\sigma = 2$, and $\sigma = 6$ respectively in Figures 2(b)- 2(d). The scale parameter is observed to be too small in Figure 2(b) as the support of f does not cover the manifolds sufficiently. The scale parameter in Figure 2(c) yields an accurate interpolation function that separates the two classes well, where the directions along which f has strong derivatives are close to the directions shown in Figure 2(a). Meanwhile, the selection of a too large value for the scale parameter in Figure 2(d) results in an overfitting of the interpolation function. In particular, strong directional derivatives are observable in directions other than the class separation boundary directions as well due to overfitting, e.g., the directions shown in red in Figure 2(d).



(a) (b) (c) (d)

Fig. 2. Illustration of the effect of the scale parameter on the accuracy of the interpolation function. (a) Manifolds $\mathcal{M}_1, \mathcal{M}_2 \subset \mathbb{R}^2$ representing two different classes and samples chosen from each class. An ideal interpolation function f separating the two classes well in $\mathbb{R}$ should have gradients in the directions shown in red. (b) Function f constructed with $\sigma = 0.5$. The scale parameter is observed to be too small as the support of the function does not cover the manifolds well. (c) Choosing the scale as $\sigma = 2$ yields a good interpolation function. (d) Choosing a too large scale parameter $\sigma = 6$ results in an overfitting of the interpolation function, with large derivatives in the indicated directions.

In optimizing σ^k, we look for an interval where $\hat{D}(f^k)$ is large enough while $\hat{G}(f^k)$ is not too high. We set the weight parameter λ to a value where the effects of both of these terms are visible, often yielding a nonmonotonic variation of the overall regularization term $\hat{G}(f^k) - \lambda \hat{D}(f^k)$, which first decreases with σ^k due to the sharp increase in $\hat{D}(f^k)$ and then increases with σ^k due to the stagnation of $D(f^k)$ and the continuing increase in the first term $\hat{G}(f^k)$. The optimal value of σ^k can then be found easily with a simple descent or line search algorithm by minimizing the one-dimensional regularization term $\hat{G}(f^k) - \lambda \hat{D}(f^k)$. We finally note that other configurations of these two terms $\hat{D}(f^k)$ and $\hat{G}(f^k)$ in a regularization objective $\hat{R}(f)$ (rather than a linear combination) may also be possible, depending on the underlying embedding. This will be discussed in more detail in Section V, as well as the links between the regularization objective $\hat{R}(f)$ and the classification performance.

Having examined the computation of the scale parameters and the coefficients of f_1 in iteration 1, we now discuss the solution of the problem (13) in a general iteration r. Due to the iterative estimation of the class labels and the calculation of the function parameters, the class labels C_{r_l} of the points x_{r_l} contributing to the embedding error (14) are already estimated in the previous iteration. The manifolds $\mathcal{M}_m$ and the embedding $\mathcal{E}$ are not explicitly known in the term $\mathcal{E}\left(P_{\mathcal{M}_m}(x_{r_l})\right)$. However, relying on a locally linear approximation of the manifolds, one can estimate the projection of a point x onto $\mathcal{M}_m$ as a convex combination of its nearest neighbors, which

can then be used to compute $\mathcal{E}\left(P_{\mathcal{M}_m}(x_{r_l})\right)$.[2] Denoting the indices of the K nearest neighbors of x within the training samples of class m as $\{a_i\}_{i=1}^K$, and the set of nearest neighbors as $\mathcal{N}_m(x) = \{x_{a_i}\}_{i=1}^K$, the projection is approximated as

$$P_{\mathcal{M}_m}(x) \approx \sum_{i=1}^{K} w_i\, x_{a_i}$$

where $w = [w_1 \dots w_K]^T$ is the vector of weights given by

$$w = \arg\min_{v} \|x - \sum_{i=1}^{K} v_i\, x_{a_i}\|^2 \quad \text{s.t.} \quad v_i \geq 0,\, \sum_{i=1} v_i = 1 \tag{20}$$

which can be solved with quadratic programming. From the continuity assumption of the embeddings, the embedding $\mathcal{E}\left(P_{\mathcal{M}_m}(x)\right)$ of $P_{\mathcal{M}_m}(x)$ is then estimated as

$$\mathcal{E}\left(P_{\mathcal{M}_m}(x)\right) \approx \sum_{i=1}^{K} w_i\, y_{a_i} \tag{21}$$

where y_{a_i} are the coordinates of x_{a_i} in the learned embedding in $\mathbb{R}^d$.

Letting $y_{r_l} = \mathcal{E}\left(P_{\mathcal{M}_{C_{r_l}}}(x_{r_l})\right)$, the total embedding error is given by

$$\hat{E}^r(f) = \hat{E}_O^r(f) + \hat{E}_T(f) = \sum_{m} \sum_{\substack{l=1 \\ C_{r_l}=m}}^{L_r} \|f(x_{r_l}) - y_{r_l}\|^2.$$

Since in iteration r an interpolation function of L_r terms is constructed, for any choice of the scale parameters $\{\sigma_l^k\}$, fitting the coefficients to the observations as $c^k = (\Phi^k)^{-1} y^k$ yields $\hat{E}^r(f) = 0$, which immediately satisfies the constraint $\hat{E}_T(f) = 0$ on training samples. It then remains to minimize the regularization term by optimizing the scale parameters as in (19).[3] This concludes the description of the proposed method. As the proposed algorithm employs unlabeled test samples in learning an out-of-sample extension, we call it Semi-supervised Out-of-Sample Interpolation (SOSI). The method is summarized in Algorithm 1.

Algorithm 1 Semi-supervised Out-of-Sample Interpolation (SOSI)

1: **Input:**
$\mathcal{X} = \{x_i\}_{i=1}^Q \subset \mathbb{R}^n$: Set of labeled and unlabeled data samples
$\{C_i\}_{i=1}^N$: Class labels of training data $\mathcal{X}_T = \{x_i\}_{i=1}^N \subset \mathcal{X}$, where $N < Q$.
2: **Initialization:** Assign number of iterations R and number of RBF terms $\{L_r\}_{r=1}^R$ in each iteration such that $L_1 = N$, $L_R = Q$ (possibly with equispaced intervals between N and Q)
3: **for** $r = 1$ **do**
4: Set kernel centers $a_l^k = x_l$ for $l = 1, \dots, N$, $k = 1, \dots, d$
5: Optimize scale parameters σ_l^k of f_1 by minimizing $\hat{R}(f)$ subject to the constraints $\sigma_l^k = \sigma^k$, $c^k = (\Phi^k)^{-1} y^k$
6: Estimate class labels C_i and compute confidence scores μ_i for $i = 1, \dots, Q$ by NN classification with f_1 in $\mathbb{R}^d$
7: **end for**
8: **for** $r = 2, \dots, R$ **do**
9: Determine $\{x_{r_l}\}_{l=1}^{L_r}$ such that $\{x_{r_l}\}_{l=1}^N = \mathcal{X}_T$ and $\{x_{r_l}\}_{l=N+1}^{L_r}$ are the points in $\mathcal{X} \setminus \mathcal{X}_T$ with highest confidence scores
10: Set kernel centers as $a_l^k = x_{r_l}$ for $l = 1, \dots, L_r$, $k = 1, \dots, d$
11: Compute the embeddings of the projections of x_{r_l} on the manifolds as in (21) and set $y_{r_l} = \mathcal{E}\left(P_{\mathcal{M}_{C_{r_l}}}(x_{r_l})\right)$
12: Optimize scale parameters σ_l^k of f_r by minimizing $\hat{R}(f)$ subject to the constraints $\sigma_l^k = \sigma^k$, $c^k = (\Phi^k)^{-1} y^k$
13: Update class labels C_i and confidence scores μ_i for $i = 1, \dots, Q$ with NN classification with f_r in $\mathbb{R}^d$
14: **end for**
15: **Output**:
Out-of-sample interpolation function $f = f_R : \mathbb{R}^n \to \mathbb{R}^d$ given by $f^k(x) = \sum_{l=1}^Q c_l^k\, \phi\left(\frac{\|x - a_l^k\|}{\sigma_l^k}\right)$
$\{C_i\}_{i=N+1}^Q$: Class labels of initially unlabeled data samples

IV. Discussion

A. Complexity analysis

We now derive the complexity of the proposed method, which is essentially determined by the complexity of steps 11-13 in the main loop of the algorithm. In step 11, the determination of the nearest neighbors in $\mathcal{X}_T$ for each test image is of complexity $O(nN)$, and the solution of the quadratic program in (20) has a polynomial-time complexity $O(\text{poly}(K))$ in the number of neighbors K [24]. The complexity dK of (21) can be neglected as d is small. Since the embedding of the projection of each point in $\mathcal{X} \setminus \mathcal{X}_T$ is computed only once throughout the algorithm, we get the overall complexity of step 11 as $O(Q\,(\text{poly}(K) + nN)) \approx O(nQN)$.

Next, step 12 requires the evaluation of the regularization term $\hat{R}(f)$ at several σ^k values and the corresponding coefficients $c^k = (\Phi^k)^{-1} y^k$. The computation of the coefficients c^k requires the solution of an $L_r \times L_r$ linear system, whose complexity is between $O({L_r}^2)$ and $O({L_r}^3)$. Then, for a given σ^k and the corresponding c^k, we analyze the evaluation of $\hat{D}(f^k)$. The computation of the gradient $\nabla f^k(x_i)$ is of complexity $O(nL_r)$. Assuming that each training point x_i has around K nearest neighbors in each one of the M classes, the computation of the directional derivative $\nabla_u f^k(x_i)$ for all neighbors of a point x_i is of complexity $O(n(L_r + KM))$. Since this is repeated for all N training points x_i, the complexity of computing $\hat{D}(f^k)$ is of $O(nN(L_r + KM))$. Since the complexity of $\hat{G}(f^k)$ is dominated by that of $\hat{D}(f^k)$, the optimization of σ^k is of $O({L_r}^2 + nN(L_r + KM))$. Performing this optimization for all d dimensions, upper bounding L_r by Q, and repeating this for all R iterations gives the complexity of step 12 throughout the algorithm as $O(dR(Q^2 + nN(Q + KM))) \approx O(dnRN(Q + KM))$. If one omits the reoptimization of σ^k for $r > 1$, the complexity of step 12 is reduced to the optimization of scale parameters at the first iteration $r = 1$ and the update of the coefficients c^k at every iteration, which is of $O(dnN(N + KM) + dRQ^2)$.

Step 13 requires the evaluation of $f(x_i)$ for all $x_i \in \mathcal{X} \setminus \mathcal{X}_T$,

[2] Note that, although the interpolation function of the previous iteration gives an estimate of the embedding of a point x as $f_{r-1}(x)$, it is more reliable to update the embedding by projecting x onto the manifold $\mathcal{M}_m$. This is because the embedding $f_{r-1}(x)$ employs no priors on the class label of x and is indeed used to estimate the class label of x, while the recomputation of the embedding as $\mathcal{E}\left(P_{\mathcal{M}_m}(x)\right)$ uses the estimated class label of x. In fact, $\mathcal{E}\left(P_{\mathcal{M}_m}(x)\right)$ coincides with the value of the updated interpolation function $f_r(x)$ of iteration r for $x = x_{r_l}$ as discussed below.

[3] In practice the optimization of scale parameters can be omitted for $r > 1$ and the scale parameters can be set to the σ^k values obtained in iteration $r = 1$ in order to speed up the algorithm without much change in the performance, as the reoptimization of the scale parameters results in σ^k values in the vicinity of those obtained at iteration $r = 1$ in general.

which is of $O(dnQL_r)$, and the comparison of the function values to those of the training points, which is of $O(dQN)$. The complexity of repeating step 13 throughout R iterations is then of $O(R(dnQ^2 + dQN)) = O(dnRQ^2)$. Finally, combining the complexities of steps 11-13, we get the complexity of the overall algorithm as $O(dnRQ^2)$.

B. Relation to kernel ridge regression

In this section, we discuss how out-of-sample extensions of supervised manifold learning methods with RBF interpolation can be interpreted within the context of kernel ridge regression. Ridge regression is a well-known statistical method that learns a linear function to model the dependency between a set of input training points $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$ and the associated outputs $\{y_i\}_{i=1}^N \subset \mathbb{R}^d$. For each dimension y_i^k of the outputs $y_i = [y_i^1 \dots y_i^d]$, the algorithm looks for a linear model $f^k(x) = w^T x$ that minimizes

$$G(w) = a\|w\|^2 + \sum_{i=1}^{N} (y_i^k - w^T x_i)^2$$

which is a slightly modified version of the least squares method by adding a regularization term representing the squared norm of the fitted linear model. Here $a > 0$ is a parameter adjusting the weight of the regularization term. An alternative formulation of ridge regression is proposed in [25] that is based on a dual version of the above problem. The solution of the dual problem yields the following prediction $f^k(x)$ of the output value for a new input sample x:

$$f^k(x) = (y^k)^T (K + aI)^{-1} v. \tag{22}$$

Here, $y^k = [y_1^k \dots y_N^k]^T$ is the vector of output values for training samples, $K \in \mathbb{R}^{N \times N}$ is the matrix of inner products of input samples whose entries are given by $K_{ij} = \langle x_i, x_j \rangle$, I is the identity matrix, and $v \in \mathbb{R}^{N \times 1}$ is the vector of inner products of x with x_i, whose ith entry is given by $v_i = \langle x, x_i \rangle$.

Since this formulation only involves the inner products between the samples x and $\{x_i\}$ rather than the samples themselves as vectors, it permits a kernel extension of the regression problem, where the samples are mapped to a high-dimensional feature space F via a kernel $\psi : \mathbb{R}^n \to F$. The inner products in K and v are then evaluated in the feature space as $K_{ij} = \langle \psi(x_i), \psi(x_j) \rangle$ and $v_i = \langle \psi(x), \psi(x_i) \rangle$. Translation-invariant kernels are a widely-used family of kernel functions, where the inner product $\langle \psi(x_i), \psi(x_j) \rangle$ in the feature space depends only on the difference $\|x_i - x_j\|$ between the samples in the original space.

Out-of-sample extensions with RBF kernels as in the proposed method are linked to kernel ridge regression in the following way. If the regularization term ($a = 0$) is omitted in (22), the kth dimension of the output vector for the input sample x is given by

$$f^k(x) = (y^k)^T K^{-1} v. \tag{23}$$

If the kernel K_{ij} is set as

$$\langle \psi(x_i), \psi(x_j) \rangle = \phi\left(\frac{\|x_i - x_j\|}{\sigma} \right)$$

with the RBF kernel used in interpolation, one can observe from (18) that the kernel matrix K coincides with the matrix Φ^k when a constant scale parameter σ is chosen for dimension k of the interpolation function. Defining v similarly with the RBF kernel ϕ, the interpolation function in (12) can be written as $f^k(x) = (c^k)^T v$. The coefficients c^k of the interpolation function being given by $c^k = (\Phi^k)^{-1} y^k$, we obtain

$$f^k(x) = (c^k)^T v = (y^k)^T (\Phi^k)^{-1} v$$

which is the same as the result obtained with kernel ridge regression in (23).

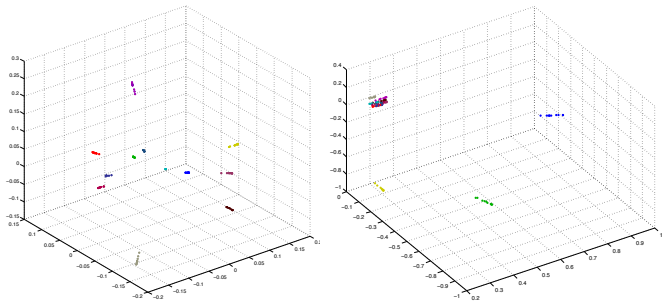
We thus observe that fitting an RBF interpolation function for manifold embeddings is the equivalent of learning a kernel ridge regression model (with no regularization) such that the output values y_i^k are the coordinates of data samples in the computed embedding. Therefore, the studied out-of-sample extension setting can be regarded as a kernel ridge regression adapted particularly to manifold-structured data. Indeed, in the general and traditional regression setting for classification, no assumption is made about the structure of data, and the output vectors y_i are taken as the class labels. Taking y_i's simply as the class labels of data transmits only the class information to the regression algorithm and conveys no information about the geometric properties of data. Meanwhile, first computing an embedding with a supervised manifold learning algorithm and then learning the regression model on the coordinates y_i^k of data in $\mathbb{R}^d$ (instead of taking y_i's directly as class labels) allows the classifier to be guided by the special geometric structure of data samples concentrated around class-representative manifolds. Coordinates learned with supervised manifold learning algorithms reinforce the class information of data by enhancing the separability between the classes, while the manifold structure of data is also preserved in each class.

V. Experimental results

In this section, we evaluate the performance of the proposed method in classification experiments. We apply the presented out-of-sample extension algorithm on two different supervised manifold learning methods. First, we consider the supervised Laplacian eigenmaps algorithm presented in [9], which computes an embedding by solving (1). Next, we evaluate our algorithm on embeddings obtained with the Fisher-like objective function in (2), which is used by methods such as [5], [6], [7], and [21]. However, we compute a nonlinear embedding by removing the linear projection constraint $z^T = v^T X$, so that the out-of-sample extension problem is of interest.

We compare the following methods in the experiments, the first four of which provide out-of-sample extension solutions for manifold embeddings. When testing the out-of-sample extension methods, class labels of test images are assigned with nearest-neighbor classification in the low-dimensional domain of embedding.

- Proposed semi-supervised out-of-sample interpolation method (SOSI)
- RBF fitting: An RBF interpolation function is fitted only to the training samples, which is the equivalent of the interpolation function f_1 computed at the end of iteration



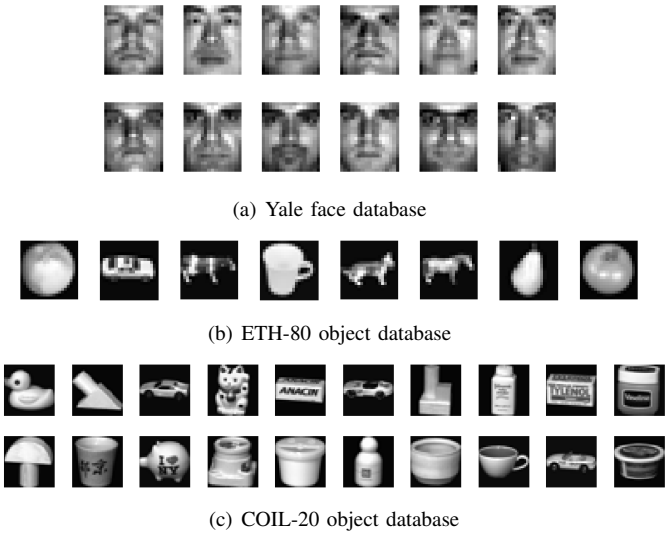
(a) Supervised Laplacian eigenmaps (b) Fisher-based embedding

Fig. 3. Three-dimensional embeddings of the Yale face data set obtained with the two manifold learning methods used in the experiments

$r = 1$ in Algorithm 1. Test images x are then mapped to $\mathbb{R}^d$ via the function $f_1(x)$.

- Locally linear embedding (LLE): Test points in $\mathbb{R}^n$ are mapped to $\mathbb{R}^d$ with an adaptation of the LLE algorithm [2] to the out-of-sample problem. Given a test point $x \in \mathbb{R}^n$, first its approximation is computed as a linear combination of its nearest neighbors in $\mathcal{X}_T$ with weights adding up to 1 as in LLE. The point x is then mapped to $y \in \mathbb{R}^d$ as the linear combination of the embeddings of the same neighbors with the same weights.
- Nyström: The original Nyström formula is not applicable since the data-dependent kernel depends on the class labels as discussed in Section II-B. We thus use a modified version of the Nyström method, where $f^k(x)$ is taken as a linear combination of the embedding coordinates y_i^k weighted by the kernel as in (3). The kernel $\tilde{M}$ in the formula is taken as the same type of kernel (Gaussian kernel) used in the construction of the within-class and between-class weight matrices W_w and W_b, and it is normalized for each test sample so that the kernel values $\tilde{M}(x, x_i)$ sum up to 1.
- Nearest neighbor classification in the original data space $\mathbb{R}^n$
- SVM in the original data space $\mathbb{R}^n$
- Semi-supervised learning (SSL) using Gaussian fields: Since the proposed out-of-sample extension method can be regarded as a building block of a semi-supervised classifier, we also compare our results with those of a semi-supervised classification method. We test the performance of SSL with the algorithm proposed in [26], which is a state-of-the-art semi-supervised classifier based on the computation of a smooth function on the data graph that coincides with the class labels when evaluated at data samples of known class labels.

We first evaluate the proposed method on a data set consisting of the face images of 12 individuals from the extended Yale face database [27], which includes 58 images of each individual taken under different poses and illumination conditions. The images are normalized, converted to grayscale and downsampled to a resolution of 17×20 pixels. A sample image of each subject in the data set is shown in Figure 4(a). The supervised Laplacian eigenmaps and the Fisher-based



(a) Yale face database

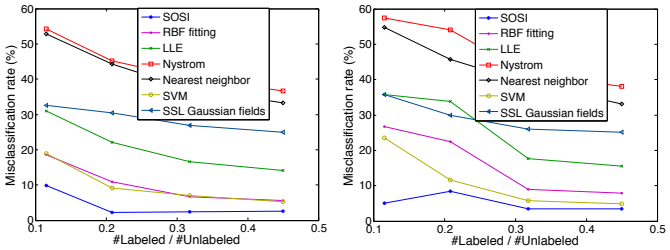
(b) ETH-80 object database

(c) COIL-20 object database

Fig. 4. Sample images from data sets used in the experiments

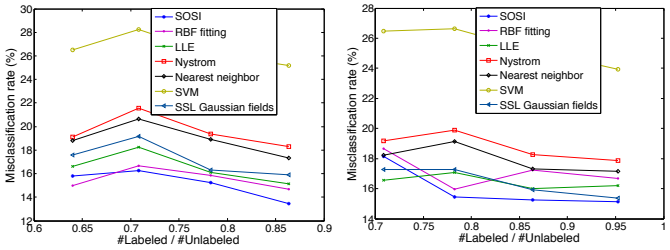
embedding algorithms are used to map the data (17×20-pixel images) to $\mathbb{R}^{20}$. The weight parameter is set as $\mu = 0.01$ in the supervised Laplacian eigenmaps method. Figure 3 shows the embeddings of a subset of the data set containing 10 labeled images of each individual, computed with the supervised Laplacian and the Fisher-based embedding algorithms. Only the first three dimensions of the coordinates are plotted for illustration. It can be observed that both methods compute representations with an enhanced separation between different classes. The supervised Laplacian eigenmaps method yields an even distribution of different classes across different dimensions. Since each dimension of the embedding renders several pairs of classes separable, sufficiently many class pairs contribute to the total directional derivative $\hat{D}(f^k)$ in (10) for each dimension k. This causes the variations of $\hat{D}(f^k)$ and $\hat{G}(f^k)$ with the scale parameter to be as discussed in Section III-B, such that $\hat{G}(f^k)$ increases at a faster rate than $\hat{D}(f^k)$ at large scales due to overfitting. Thus, for the embeddings obtained with supervised Laplacian eigenmaps, we optimize the scale parameters by minimizing the regularization term $\hat{R}(f)$ as in (8).[4] Meanwhile, the embedding computed with the Fisher-based objective yields a more "polarized" representation, where each dimension of the embedding is observed to separate out only one class from the others. When there are not sufficiently many separable class pairs in $\hat{D}(f^k)$, the estimation of the variation of this term with the scale parameter may become unreliable or biased by a particular class in each dimension. We have observed that, when the embedding is computed with the Fisher-based objective, the variation of $\hat{D}(f^k)$ with the scale parameter is closer to that of $\hat{G}(f^k)$ (in comparison with supervised Laplacian eigenmaps). The choice of the regularization term $\hat{R}(f)$ as a linear

[4]Occasionally, the scale parameter σ^k of one dimension or a few dimensions k may diverge from the scale parameters of the rest of the dimensions, which may cause instabilities. In order to avoid this, we bound the final values of the scale parameters to an interval of two standard deviations around their mean value averaged over all dimensions.



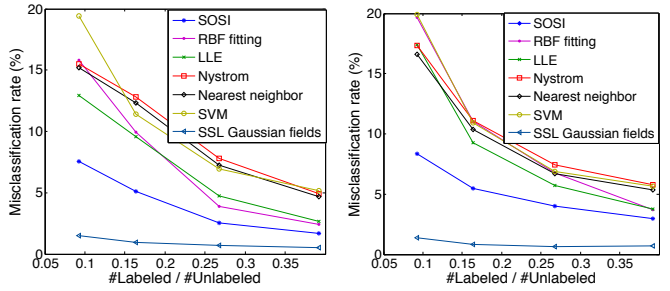
(a) Supervised Laplacian eigenmaps (b) Fisher-based embedding

Fig. 5. Misclassification rates of face images from Yale database



(a) Supervised Laplacian eigenmaps (b) Fisher-based embedding

Fig. 6. Misclassification rates of object images from ETH-80 database



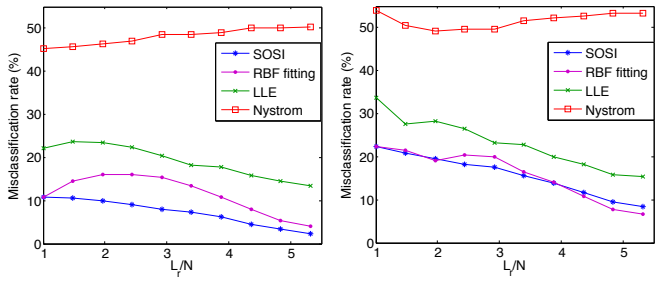
(a) Supervised Laplacian eigenmaps (b) Fisher-based embedding

Fig. 7. Misclassification rates of object images from COIL-20 database

combination of these two terms may then lose its reliability, as it may become a monotonic function of the scale parameter, for instance. Therefore, for the Fisher-based embedding, we apply a slightly modified procedure for optimizing the scale parameters, where we choose a sufficiently large value for the scale parameter in each dimension, which ensures, however, that the $\hat{D}(f^k)/\hat{G}(f^k)$ ratio stays above a certain threshold value. The scale parameters of the RBF fitting method are set as equal to those of the proposed SOSI algorithm. Figure 5 shows the classification errors obtained with all methods for the supervised Laplacian and the Fisher-based embeddings. Each curve displays the misclassification rate (in percentage) of unlabeled images, obtained by varying the ratio between the number of labeled and unlabeled images in the data set. The results are the average of 5 repetitions of the experiment by randomly choosing the labeled samples. An early stopping rule is applied in the SOSI algorithm for the leftmost point of the curve (the labeled/unlabeled ratio of 0.11) due to the relatively high error, where the interpolation function construction is terminated when around 80% of the unlabeled points are added as RBF kernel centers. It is observed that the proposed method outperforms the other out-of-sample extension methods in comparison, as well as the SVM classifier and the semi-supervised graph-based classifier.

We then repeat the same experiment on two different databases of object images captured under varying viewpoints. The first experiment is conducted on the images of 8 objects from the ETH-80 database [28], where 41 images are available for each object (in particular, the images of the first object in each object category are used so that the images in each class belong to the same manifold). A sample image of each object is shown in Figure 4(b). The images are normalized, converted to grayscale, and downsampled to a resolution of 20×20 pixels. An embedding of dimension $d = 15$ is computed with the supervised Laplacian eigenmaps and the Fisher-based manifold learning algorithms. The second experiment is done on the images of 20 objects from the COIL-20 database [29] with 71 images for each object, which are normalized, converted to grayscale, and downsampled to a resolution of 32×32 pixels. Figure 4(c) shows a sample image for each object. The images are embedded in a space of dimension $d = 25$. In both experiments, the optimization of the scale parameters is done as in the previous experiment. The results obtained with the two object data sets are presented in Figures 6 and 7. The misclassification rates of unlabeled samples are plotted with respect to the ratio between the number of labeled and unlabeled samples. The results are the average of 5 random partitionings of the data set. As the classification error of the ETH-80 database is relatively high, an early stopping rule is applied for this data set by terminating the interpolation function construction when around 70% of the unlabeled samples with the highest confidence scores are added as RBF kernel centers. The results show that the proposed method often yields the smallest classification error in the experiment of Figure 6, while it is outperformed only by the semi-supervised learning method in Figure 7. This graph-based semi-supervised learning algorithm performs particularly well on the COIL-20 data set, due to the dense sampling and the regular structure of the object image manifolds.

The overall consideration of these experiments shows that the proposed out-of-sample extension method for supervised manifold learning provides a better performance than the reference out-of-sample extension strategies in comparison, while it can provide an alternative solution for semi-supervised learning when coupled with a supervised dimensionality reduction method and thus regarded as one building block of a semi-supervised classifier. In particular, one can observe in Figures 5-7 that SVM and graph-based SSL may perform very differently in different settings. SVM is based purely on the representation of the data samples in the original ambient space $\mathbb{R}^n$, while graph-based SSL only uses the information of the similarities between neighboring data samples instead of interpreting them as vectors in the high-dimensional space $\mathbb{R}^n$. Meanwhile, the proposed method is expected to find a compromise between these two approaches, as the interpolation function depends both on the coordinates of the data samples in $\mathbb{R}^n$ and the coordinates of the embedding in $\mathbb{R}^d$ learned with a supervised manifold learning algorithm that relies on the



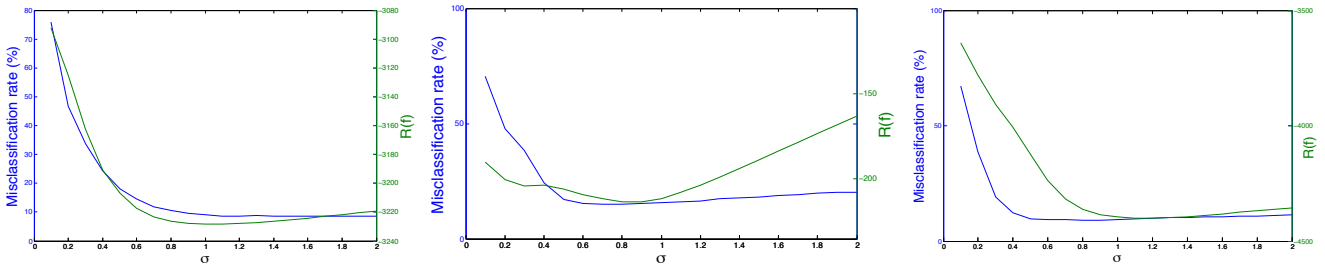
(a) Supervised Laplacian eigenmaps (b) Fisher-based embedding

Fig. 8. Misclassification rates obtained with progressive integration of the test images in the extended training set. Out-of-sample extensions are computed, class labels are assigned, and embeddings are updated with the extended training set in each iteration.

graph representation of data. The experimental results seem to confirm this expectation, as the proposed classification solution attains a reasonably good performance in situations where SVM or graph-based SSL may fail (as in Figures 6 and 5 respectively).

In the experiments of Figures 5-7, the interpolation functions of the out-of-sample methods other than SOSI are constructed using only the training data. The information present in the unlabeled data samples is not exploited in the construction of these interpolation functions, whereas SOSI uses these points to gradually add them as kernel centers of the learned interpolation function. In order to assess the performance of the proposed method in the progressive integration of the unlabeled data samples in the learning process, we do an additional experiment. The proposed SOSI algorithm is used to classify unlabeled test images in an iterative way as described in Algorithm 1. Then, in order to compare SOSI with the other out-of-sample extension methods, for each one of these methods, we carry out an iterative classification procedure as follows. In each iteration, all test images are assigned class labels with nearest-neighbor classification in the low-dimensional domain via the out-of-sample generalization strategies in comparison, and a confidence score is obtained for each test image as in (16). Then in the next iteration, the test images with the highest confidence scores are added to the training set with their estimated class labels and a completely new embedding of this extended training set is computed with the supervised Laplacian eigenmaps and the Fisher-based embedding algorithms (thus new coordinates are assigned to the original training images as well). The out-of-sample extension of this new embedding is then recomputed with the tested strategies in comparison, which are used to reclassify the test images. In each iteration r, the compared methods use the same number L_r of extended training images in $\mathcal{X}$ (same as the number of terms in the interpolation function of SOSI), while the choice of the extended training set varies between the compared methods as a result of the different confidence scores they assign to the test images. This progressive procedure is continued until all test images are included in the extended training set. The results obtained on the face images from the Yale database are presented in Figures 8(a) and 8(b), respectively for the supervised Laplacian eigenmaps and the Fisher-based embedding algorithms. The image set of each subject contains 10 labeled and 48 unlabeled samples in this experiment. The misclassification rates obtained throughout the iterations are plotted with respect to the ratio L_r/N between the size of the extended training set (number of RBF terms for SOSI) and the size of the original training set. The results indicate that the best classification accuracy is achieved by the proposed algorithm most of the time. The misclassification error obtained with the proposed method decreases regularly throughout the iterations as the number of terms in the interpolation function increases, while the evolution of the misclassification error with the other strategies is less regular and the error may even increase throughout the iterations. This is due to the fact that the strategies other than SOSI compute a new embedding of the extended training set from scratch in each iteration. The mislabeled data samples in the extended training set may then significantly influence the computed embedding and consequently the class label assignments of the next iteration, since the embedding given by the eigenvectors of a class-dependent kernel matrix may change dramatically even with small errors in the kernel matrix. The proposed method does not suffer from this problem, since it preserves the original embedding and refines only the interpolation function throughout the iterations, which has a regularizing effect that better tolerates inaccurate assignments of the class labels of test images. We finally note that, among the strategies compared in this experiment, the proposed SOSI algorithm is the only one that provides an out-of-sample solution for manifold learning when $L_r > N$.

Finally, we study the influence of the scale parameters of the interpolation function on the classification performance. As discussed in Section III-B, the proposed method selects the scale parameters by optimizing the regularization term $\hat{R}(f)$. In order to evaluate the effect of this regularization approach on the classification accuracy, we compare the variations of the regularization term $\hat{R}(f)$ and the classification error with the scale parameter. We compute an embedding of the training images with the supervised Laplacian eigenmaps algorithm and then construct an RBF interpolation function, where all scale parameters σ_l^k are set to a common σ value and the coefficients c_l^k are computed to fit the training images and the learned embedding for this choice of the scale parameter (as in the RBF fitting method or the first iteration of SOSI). A sequence of interpolation functions are computed by varying the scale parameter σ, and for each interpolation function, the regularization objective $\hat{R}(f)$ is computed as well as the misclassification rate of the test images. The variations of the regularization cost and the misclassification rate with the scale parameter σ are presented in Figure 9 for all three data sets used in the experiments. The results suggest that the regularization objective $\hat{R}(f)$ has a rather smooth and nonmonotonic variation with the scale parameter, which resembles that of the classification error. Moreover, the interval of scale parameters σ minimizing the regularization objective $\hat{R}(f)$ coincides with the range of σ values where the misclassification rate takes small values. This shows that the proposed regularization objective permits the algorithm to capture the influence of the scale parameters on the performance of learning and can be



(a) Yale face data set (b) ETH-80 object data set (c) COIL-20 object data set

Fig. 9. Variations of the misclassification error and the regularization term $\hat{R}(f)$ with the scale parameter of the RBF kernels

used for optimizing the scale parameters.

VI. Conclusions

We have proposed a method for the out-of-sample extensions of supervised manifold learning algorithms that embed a set of class-representative manifolds residing in a high-dimensional ambient space to a set of manifolds in a lower-dimensional domain. The proposed out-of-sample generalization method is based on the construction of an RBF interpolation function, where the parameters of the interpolation function are optimized to minimize the embedding error over a set of initially unlabeled data samples, whose class labels are estimated progressively along with the parameters of the interpolation function. We have shown that the regularity of the interpolation function can be controlled by optimizing the RBF scale parameters to minimize a regularization objective that controls the total gradient of the interpolation function while encouraging sufficiently strong derivatives along the directions of class separation boundaries in order to ensure an effective separation between different classes. The proposed out-of-sample generalization method outperforms baseline interpolation solutions in classification applications. Experimental results suggest that the proposed algorithm achieves state-of-the-art performance in semi-supervised learning and can be effectively used along with supervised manifold learning methods in the classification of low-dimensional data sets consisting of labeled and unlabelled data samples.

VII. Acknowledgment

The authors would like to thank Pascal Frossard and Alhussein Fawzi for the helpful discussions that contributed to this study.

References

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

[2] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[4] S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, 2007.

[5] Q. Hua, L. Bai, X. Z. Wang, and Y. Liu, "Local similarity and diversity preserving discriminant projection for face and handwriting digits recognition." *Neurocomputing*, vol. 86, pp. 150–157, 2012.

[6] W. Yang, C. Sun, and L. Zhang, "A multi-manifold discriminant analysis method for image feature extraction," *Pattern Recognition*, vol. 44, no. 8, pp. 1649–1657, 2011.

[7] Z. Zhang, M. Zhao, and T. Chow, "Marginal semi-supervised sub-manifold projections with informative constraints for dimensionality reduction and recognition," *Neural Networks*, vol. 36, pp. 97–111, 2012.

[8] Q. Gao, J. Ma, H. Zhang, X. Gao, and Y. Liu, "Stable orthogonal local discriminant embedding for linear dimensionality reduction." *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2521–2531, 2013.

[9] B. Raducanu and F. Dornaika, "A supervised non-linear dimensionality reduction approach for manifold learning," *Pattern Recognition*, vol. 45, no. 6, pp. 2432–2444, 2012.

[10] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering," in *Adv. Neural Inf. Process. Syst.* MIT Press, 2004, pp. 177–184.

[11] G. H. Chen, C. Wachinger, and P. Golland, "Sparse projections of medical images onto manifolds," in *Information Processing in Medical Imaging - 23rd International Conference, IPMI 2013, Asilomar, CA, USA, June 28-July 3, 2013. Proceedings*, 2013, pp. 292–303.

[12] H. Qiao, P. Zhang, D. Wang, and B. Zhang, "An explicit nonlinear mapping for manifold learning," *IEEE T. Cybernetics*, vol. 43, no. 1, pp. 51–63, 2013.

[13] B. Peherstorfer, D. Pflüger, and H. J. Bungartz, "A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with laplacian eigenmaps," in *AI 2011: Advances in Artificial Intelligence - 24th Australasian Joint Conference, Perth, Australia, December 5-8, 2011. Proceedings*, 2011, pp. 112–121.

[14] H. Strange and R. Zwiggelaar, "A generalised solution to the out-of-sample extension problem in manifold learning," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3603

[15] M. W. Trosset and C. E. Priebe, "The out-of-sample problem for classical multidimensional scaling," *Computational Statistics & Data Analysis*, vol. 52, no. 10, pp. 4635–4642, 2008.

[16] T. J. Chin and D. Suter, "Out-of-sample extrapolation of learned manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1547–1556, 2008.

[17] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 77–90, 2006.

[18] X. He and P. Niyogi, "Locality Preserving Projections," in *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press, 2004. [Online]. Available: http://books.nips.cc/nips16.html

[19] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.

[20] D. Cai, X. He, J. Han, and H. Zhang, "Orthogonal Laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.

[21] R. Wang and X. Chen, "Manifold discriminant analysis," in *CVPR*, 2009, pp. 429–436.

[22] D. Xu, S. Yan, D. Tao, S. Lin, and H. Zhang, "Marginal fisher analysis and its variants for human gait recognition and content- based image

retrieval," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2811–2821, 2007.

[23] M. D. Buhmann, *Radial Basis Functions*. New York, NY, USA: Cambridge University Press, 2003.

[24] M. K. Kozlov, S. P. Tarasov, and L. Khachiyan, "The polynomial solvability of convex quadratic programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 5, pp. 223 – 228, 1980.

[25] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 515–521.

[26] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Machine Learning, Proceedings of the Twentieth International Conference, August 21-24, 2003, Washington, DC, USA*, 2003, pp. 912–919. [Online]. Available: http://www.aaai.org/Library/ICML/2003/icml03-118.php

[27] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[28] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, 2003, pp. 409–415.

[29] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep., Feb 1996.