Dimensionality Reduction and Principal Surfaces via Kernel Map Manifolds

Samuel Gerber Tolga Tasdizen Ross Whitaker
Scientific Computing and Imaging Institute
University of Utah, Salt Lake City
sgerber@cs.utah.edu

Abstract

We present a manifold learning approach to dimensionality reduction that explicitly models the manifold as a mapping from low to high dimensional space. The manifold is represented as a parametrized surface represented by a set of parameters that are defined on the input samples. The representation also provides a natural mapping from high to low dimensional space, and a concatenation of these two mappings induces a projection operator onto the manifold. The explicit projection operator allows for a clearly defined objective function in terms of projection distance and reconstruction error. A formulation of the mappings in terms of kernel regression permits a direct optimization of the objective function and the extremal points converge to principal surfaces as the number of data to learn from increases. Principal surfaces have the desirable property that they, informally speaking, pass through the middle of a distribution. We provide a proof on the convergence to principal surfaces and illustrate the effectiveness of the proposed approach on synthetic and real data sets.

1. Introduction

Finding a low dimensional representation of a high dimensional data set is a task with applications in many areas of computer vision. Common approaches are principal component analysis (PCA) or factor analysis if the data is assumed to exhibit a linear structure. In recent years researchers proposed various approaches to tackle the nonlinear case, generally referred to as nonlinear dimensionality reduction or manifold learning. Manifold learning refers to the assumption that the data lies on or near a low dimensional manifold embedded in a high dimensional space.

Several applications of dimensionality reduction require the projection of new data points onto the manifold or the reconstruction of the high-dimensional data from manifold coordinates. For such applications it is natural to quantify the performance of the learned manifold by projection distance, and yet this is seldom considered by state-of-the-art methods. For instance, neither isomap nor Laplacian eigenmaps produce any explicit mappings between the coordinate systems of the manifold (the parametrization), $\mathscr{C}$, and the ambient data space, $\mathscr{D}$. Researchers usually resort to a (weighted) nearest neighbor averaging [24], but there is nothing in the construction of manifolds from these methods that gives any assurances about the quality of these approximations.

Based on these observations we propose an approach to manifold learning that explicitly provides mappings for embedding and reconstruction and specifically optimizes the projection distance. The mappings are computed by kernel regression on manifold coordinates defined on the input data. The coordinates determine the shape of the manifold. Thus we can evaluate and, by adjusting the manifold coordinates, optimize the manifold in terms of projection distance. We call this approach the *Kernel Map Manifolds* (KMMs). In this paper we show that KMMs are, formally, principal surfaces [12]. We show that KMMs can easily be used in conjunction with and improve on existing global methods and readily extend to *n*-dimensional principal surfaces.

2. Background

The research on nonlinear dimensionality reduction is extensive, we give a brief summary of work that is most relevant to the proposed approach.

Much recent research in manifold learning builds on the ideas of isomap [28], Laplacian eigenmaps [3] and local linear embedding [25]. These methods are often used in practice and they offer *global* solutions that do not rely on any initial estimates of the manifold. Many other techniques are closely related to or explicitly build on these three methods [9, 33, 31, 30, 6]. We call these methods *spectral approaches* or *global methods*, because they are based on quantifying the local structure of the input data but solve the problem in a global fashion. Hence these methods have a closed form solution that entails computing the spectral decomposition of a matrix that captures this local information. Out of sample extensions to these methods have been proposed [4] but there is nothing in the construction of manifolds from these methods that guarantees the quality of the manifold fit in terms of projection distance.

Another area of related research is the work on *principal surfaces* [12]. Principal surfaces are a conceptual extension of PCA to the nonlinear case. Intuitively a principal surface passes through the middle of a distribution. There are variations [29, 15] of the original definition proposed by Hastie as well as various heuristically motivated algorithms [20, 26, 16, 19]. For instance, probabilistic principal surfaces (PPS) [7] use generative topographical mapping (GTM) to compute principal surfaces. However none of these approaches comes with theoretical guarantees and few extend beyond one and two dimensional manifolds.

Gaussian process latent variable models (GPLVM) [17] provide a probabilistic generative model of the data. Similar to the proposed method the manifold can be modeled as the mean of a *learned* Gaussian process. However GPLVM do not provide a direct approach to project unseen data points onto the manifold.

Minimizing reconstruction error is also the objective of *auto encoders* [5, 22, 13] from the neural network literature. Obtaining good minima can be difficult and the reliance on neural networks prevents making theoretical connections to principle surfaces [7]. The recently proposed optimization method of Hinton [13], for training deep neural networks, is very promising. However, because KMMs explicitly optimize manifold parameters, we can solve for KMMs using initial estimates from one or more of the global methods.

Manifold learning has been successfully applied to a wide variety of applications in computer vision. Etyngier et al. [10] compute shape manifolds used as segmentation priors. In [18, 11] the manifold structure of posture and viewpoint is exploited to enhance tracking performance. In face recognition manifold based approaches are used to account for pose and illumination [1]. Pless [23] uses isomap to explore video sequences. In medical image analysis manifold learning is used to enhance registration and segmentation tasks [32]. In [17, 27] applications of manifold learning to motion capture data are illustrated.

3. Kernel Map Manifolds

We begin with some notation. We denote an input data point $y \in \mathscr{D}$ where $\mathscr{D} \subset \mathbb{R}^D$ (open and continuous). We let $x \in \mathscr{C}$, where $\mathscr{C} \subset \mathbb{R}^d$, be the low dimensional representation with x_i corresponding to the manifold coordinates of the projection of a data point y_i onto the manifold, and we assume that $d < D$. Because we are also concerned with finding mappings between these spaces, we denote the mapping to the manifold as $f : \mathscr{D} \mapsto \mathscr{C}$, and we call this the *coordinate mapping*. We denote with $g : \mathscr{C} \mapsto \mathscr{D}$ the mapping from manifold coordinates to the higher dimensional data space, and we call this the *reconstruction mapping*. The image of g is, barring degeneracies, a d-dimensional manifold in $\mathscr{D}$, which we denote $\mathscr{M}$.

We model the input data as a set of random samples from a density function $p(y)$ defined on $\mathscr{D}$. Our goal, therefore, is to define a low-dimensional representation of $p(y)$. Principle surfaces require a projection operator, and we can obtain such a projection by a composition of the two mappings $g \circ f$, which maps every point $y \in \mathscr{D}$ onto $\mathscr{M}$. The coordinates $f(y)$ correspond to positions on the manifold $\mathscr{M}$. The definition of principal surfaces [12] requires a specific form for $g(x)$, which must be the expectation of all points y that have position x on the manifold. Thus we have

$$g(x) = \int_{\mathscr{D}} y \delta(f(y) - x) p(y) dy \tag{1}$$

where $\delta(\cdot)$ is the Dirac-delta distribution. Given this, we have only to define $f(y)$, which in order to satisfy the principal surface constraints, must be an orthogonal projection onto manifold $\mathscr{M}$. Our strategy is to represent $f(y)$ using a kernel regression of parameters z that are defined on the input data points.

Let $Y = \{y_i \ \in \mathscr{D}\}$ be a set of random samples, of size N, drawn from $p(y)$, and we assign a set of parameter $Z = \{z_i = \phi(y_i) \in \mathscr{C}\}$ to each input data point, where $\phi : Y \to Z$ is a mapping defined on the discrete set Y. Using kernel regression on these samples we have the coordinate mapping

$$f(y) = \sum_j^N \frac{K_y(y - y_j)}{\sum_k^N K_y(y - y_k)} z_j = \sum_j^N \frac{K_y(y - y_j)}{\sum_k^N K_y(y - y_k)} \phi(y_j) \tag{2}$$

with K_y a kernel function. It is important to notice that the set Z is not the coordinate mapping of y itself, because of the effect of kernel regression. Instead we can think of Z as a finite set of *parameters* that describe the coordinate mapping f. As required in equation (1) we define $g : \mathscr{C} \mapsto \mathscr{D}$ as the conditional expectation of Y given $f(Y) = x$ using Nadaraya-Watson kernel regression

$$g(x) = \sum_j^N \frac{K_x(x - f(y_j))}{\sum_k^N K_x(x - f(y_k))} y_j, \tag{3}$$

Both f and g are examples of Nadaraya-Watson kernel regression, and they converge as $N \to \infty$ to the conditional expectation, $E[\cdot|\cdot]$, of the underlying variables. For g this is an approximation to $E[Y|f(Y) = x]$. Likewise f is an approximation to $E[\phi(Y)|Y = y] = \phi(y)$.

Measuring projection distance is now immediate by $\|g(f(y)) - y\|$. The residual

$$J[f] = \int_{\mathscr{D}} \|g(f(y)) - y\|^2 p(y) dy \tag{4}$$

gives a measure for the quality of the low-dimensional representations in terms of projection distance. In the discrete setting the residual can be approximated by

$$J[f] = \sum_i \|g(f(y_i)) - y_i\|^2. \tag{5}$$

Since f is defined in terms of Z and by expanding g we can write the residual as

$$J(Z) = \sum_i^N \Big\| \sum_j^N \frac{K_x(f(y_i) - f(y_j))}{\sum_k^N K_x(f(y_i) - f(y_k))} y_j - y_i \Big\|^2. \qquad (6)$$

Notice that the use of kernel regression also gives immediate rise to an estimated density in the low as well as in the high dimensional space.

4. Optimizing Kernel Map Manifolds

Minimizing the residual (4) leads to a solution with minimal projection distance. In case of a Gaussian noise model this corresponds to a maximum likelihood estimate of the manifold. Note that (4) is by no means convex and has potentially many local minima. Thus, the effectiveness of the approach depends on the minimization strategy. We propose to use a global scheme such as isomap to initialize Z, and then refine these initial estimates with a gradient descent on the residual.

4.1. Optimization

We optimize the KMM with respect to $Z = \{z_1, \dots, z_N\}$ using a gradient descent on the energy defined on 5. The gradient for a single z_r is

$$\nabla_{z_r} J(Z) = \sum_i^N 2(g(f(y_i)) - y_i)\nabla_{z_r}(g \circ f)(y_i) \qquad (7)$$

where $\nabla_{z_r}(g \circ f)(y_i)$ is the gradient of the function $g \circ f : \mathscr{D} \to \mathscr{M}$ with respect to each component $z_r[k]$ of z_r evaluated at y_i.

$$\nabla_{z_r}(g \circ f)(y_i) = \begin{bmatrix} \frac{\partial (g \circ f)_1(y_i)}{\partial z_r[1]} & \cdots & \frac{\partial (g \circ f)_1(y_i)}{\partial z_r[d]} \\ \vdots & \vdots & \vdots \\ \frac{\partial (g \circ f)_D(y_i)}{\partial z_r[1]} & \cdots & \frac{\partial (g \circ f)_D(y_i)}{\partial z_r[d]} \end{bmatrix} \qquad (8)$$

which is

$$\nabla_{z_r}(g \circ f)(y_i) = \sum_j^N \Big[\frac{\partial K_x(f(y_i) - f(y_j))}{\partial(f(y_i) - f(y_j))} \frac{\partial(f(y_i) - f(y_j))}{\partial z_r} \sum_k^N K_x(f(y_i) - f(y_k)) - K_x(f(y_i) - f(y_k)) \sum_k^N \frac{\partial K_x(f(y_i) - f(y_k))}{\partial(f(y_i) - f(y_k))} \frac{\partial(f(y_i) - f(y_k))}{\partial z_r} \Big] \frac{y_j}{(\sum_k^N K_x(f(y_i) - f(y_k)))2} \qquad (9)$$

with $\frac{\partial K_x(f(y_i) - f(y_j))}{\partial(f(y_i) - f(y_j))}$ the derivative of the kernel function and

$$\frac{\partial(f(y_i) - f(y_j))}{\partial z_r} = \frac{K_y(y_i - y_r)}{\sum_k^N K_y(y_i - y_k)} - \frac{K_y(y_j - y_r)}{\sum_k^N K_y(y_j - y_k)} \qquad (10)$$

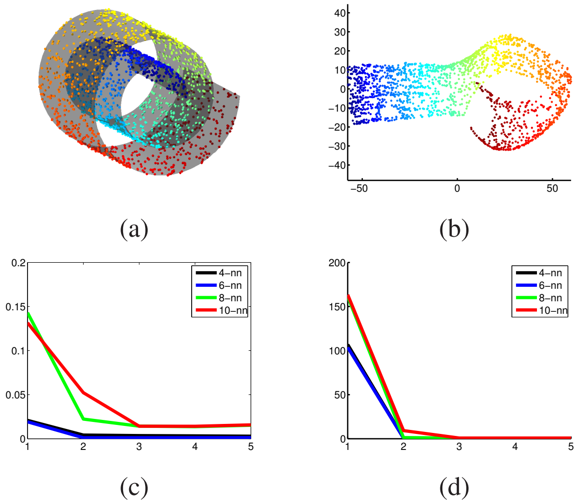
Each iteration of this procedure is $O(n^3 dD)$, where $n = |Y|$ the number of points, and D and d are the dimensions of the data and manifold, respectively. This heavy computational burden stems from the fact that we must compute the gradient for each data point. The gradient computation for each point entails, as equation (7) shows, a summation over all points, where each summand is the matrix vector multiplication $(g(f(y_i)) - y_i)\nabla_{z_r}(g \circ f)(y_i)$, which is itself $O(dD)$. Computing the matrix $\nabla_{z_r}(g \circ f)(y_i)$ for each summand entails again a summation over all points as can be seen in equation (9) (the inner summations can be precomputed). The kernel allows us to reduce the computational cost by only using points that are within in a certain distance, e.g. for a Gaussian kernel with three standard deviations which amounts for more than 99 percent of the probability density. This reduces the number of matrix-vector multiplications as well as for computing the gradient matrix $\nabla_{z_r}(g \circ f)(y_i)$ to a small subset of size c of the original points. The size of c is only dependent on the kernel bandwidth and not the number of data points. Hence the computational cost is reduced to $O(nc^2 2dD)$.

4.2. Initialization

The energy (5) has potentially many local minima. This requires a good initialization scheme in order for gradient descent to find an acceptable solution. Our strategy is to use a *global* manifold learning formulation as an initial estimate of parameters Z. The advantage of the formulation, however, is that we can use the projection error in (5) for selecting from among the various dimensions and parameters of these global approaches. For this discussion, and the subsequent results, we restrict our attention to isomap—but the strategy is applicable with any *global method*.

The isomap algorithm relies on geodesic distances rather than derivatives. Geodesic distances on the manifold are approximated by shortest paths in the k-nearest-neighbor connectivity graph constructed from the input data. This allows to compute all pair wise distances between the data points. Multidimensional scaling [8] (low rank approximation of the centered pair wise distance matrix) on this pairwise distance matrix yields a minimal distortion embedding. The connectivity graph is constructed from nearest neighbors of the high dimensional data y. The shortest path computations can be sensitive to the number of nearest neighbors, k, used in building the graph. This can be prone to create shortcuts in the presence of noise or low sampling density, as described in [2]. Isomap uses residual variance as a measure for the goodness of a low-dimensional representation, a measure of distortion. This measure can indicate a good low dimensional representation for a graph even if the topology of the manifold is not correct, for example if there are shortcuts in the graph.

In essence Isomap is a graph layout technique, and be-



(a) (b)

(c) (d)

Figure 1. A 2-dimensional isomap embedding (b) using 10 nearest neighbors for the noisy swissroll (a), both colored by angle. Shortcuts in the graph lead to folds in the isomap embedding. Residual variances from isomap (c) and residuals from KMM (d) as a function of the number of dimensions for the swissroll in (a) with 4, 6, 8 and 10. The KMM provides an accurate estimate of intrinsic dimensionality even if the isomap embedding has folds.

cause there is no intrinsic relationship between the graph topology and the manifold topology, Isomap can give misleading results. This is demonstrated on the swissroll in figure 1, where the residual variance is not necessarily indicative of a good low dimensional representation. In figure 1 it is easy to asses the quality of the low dimensional representation visually, but this visual assessment is, not feasible beyond 3-dimensional data. However, the residual from KMM provides an independent measure of the quality of the underlying manifold. Thus, we can select the number of nearest neighbors k by constructing KMMs obtained from isomaps with varying k and choose the k that yields a minimal residual KMM. Figure 1 illustrates that the KMM residual is a more reliable indicator for the intrinsic dimensionality of the manifold.

4.3. Kernel Bandwidth and Stopping Criteria

The general formulation for KMMs does not specify the kernels for f and g. In our experiments we use a Gaussian kernel because of its differentiability, although a compact kernel could potentially save some computation.

Virtually all kernels for nonparametric regression include a bandwidth that influences the smoothness of the resulting estimate. For KMMs the bandwidths of kernels used in representing f and g interact. We gain some information by looking at the degenerate cases. For instance, as we let the bandwidth of f go to infinity, the coordinate mapping collapses to a single point. The reconstruction is therefore the mean of the data set, which is the *zero dimensional* principal surface. If the bandwidth of f is finite the coordinates x will not coincide. During optimization f can potentially spread the coordinates x further apart. Because the bandwidth of g is fixed, the corresponding manifold can be more curved. Thus, we see that KMMs could potentially over fit the data. The coordinates x can spread so far apart that g degenerates to a mapping onto the original data points. This can immediately be seen for a zero bandwidth for f, e.g. f is simply the initial discrete mapping ϕ. By pushing the coordinate mapping parameters Z apart we have that $K_x(f(y_i) - f(y_k)) = K_x(z_i - z_k) \rightarrow 0$ except for $i = k$. The solution degenerates to the original point cloud and the residual goes to zero.

To circumvent these dangers, we propose to use cross validation, either a leave-one-out scheme or with a distinct cross validation data set. We stop the optimization before it reaches steady state, i.e. when the projection error on the cross validation data set stops decreasing. We choose a bandwidth for f and g automatically, using the average distance between k-nearest neighbors of the original data and the initial KMM, respectively. For k we use the number of nearest neighbors from the initialization (e.g. isomap) that gives the lowest projection error for an initial set of parameters Z.

5. Kernel Map Manifolds and Principal Surfaces

The formulation of the mappings as a kernel regression leads to some important properties that hold asymptotically. In this section we show that optimal KMMs converge to *principal surfaces* as the number of points goes to infinity.

Definition 1. *Principal Surface [12]: Let Y be a D dimensional random variable and $h : \mathscr{C} \rightarrow \mathscr{D}$ denote a d-dimensional surface, $\mathscr{M}$ in $\mathscr{D}$ parametrized by $x \in \mathscr{C}$. Let $x_h(y) = \max_x\{x : \|y - h(x)\| = \inf_\mu \|y - h(\mu)\|\}$ the coordinate mapping (in [12] called projection index). The principal surfaces of Y are the set $\mathscr{H}$ of functions h that fulfill the self consistency property $E[Y|x_h(Y) = x] = h(x)$. Alternatively h is a principal surface of Y if and only if h is an extremal point of $E[\|Y - h(x_h(Y))\|^2]$.*

Note that the second condition implies $h(x_h(y))$ is an orthogonal projection of y onto h. In particular it is the minimal orthogonal projection. This formalizes the notion of a manifold passing through the middle of a distribution.

We introduce a second weaker formulation:

Definition 2. *Weak Principal Surface. As before let Y be a D dimensional random variable and $h : \mathscr{C} \rightarrow \mathscr{D}$ denote a d-dimensional surface in $\mathscr{D}$ parametrized by $x \in \mathscr{C}$. Let $\tilde{x_h}(y) = \{x : (y - h(x))D(h)(x) = 0\}$ be the coordinate mapping. $D(h)(x)$ denotes the Jacobian matrix of h evaluated at x. The weak principal surfaces of Y is set $\tilde{\mathscr{H}}$ of functions*

h that fulfill the self consistency property $E[Y|\tilde{x}_h(Y) = x] = h(x)$.

A weak principal surface enforces the self consistency but does not require the distance from each point to the manifold to be globally minimal. $\mathscr{H}$ is a strict subset of $\tilde{\mathscr{H}}$. Weak principal surfaces formalize the notion of a manifold passing through the middle of a distribution, but allow for *misassignments* of data to the manifold in cases where the shape of the manifold generates ambiguities in the direction of the projection.

Theorem 5.1. *Local minima of the energy* (5) *are weak principal surfaces as $|Y| \to \infty$.*

Proof. The self consistency follows immediately from our definition of g as Nadaraya-Watson Kernel regression. This estimator g converges to the conditional expectation $g(x) \to E[Y|f(Y) = x]$ as the size of the data goes to infinity [21]. Now if $g(f(y)) - y$ is a orthogonal projection then $g(f(y))$ is a weak principal surface. Here the key is that f is also defined as Nadaraya-Watson kernel regression. Hence,

$$f(y) = \sum_j^N \frac{K_y(y - y_j)}{\sum_k^N K_y(y - y_k)} \phi(y_j) \to E[\phi(Y)|Y = y] = \phi(y). \tag{11}$$

Taking the first variation of the energy (4), with the limit $f(y) \to \phi(y)$, and setting it equal to zero gives

$$\frac{\partial J[\phi]}{\partial \phi} = \int 2(g(\phi(y)) - y) \frac{\partial g(\phi(y))}{\partial \phi(y)} dy = 0. \tag{12}$$

From the fundamental lemma of the calculus of variation this has to hold point wise

$$(g(\phi(y)) - y) \frac{\partial g(\phi(y))}{\partial \phi(y)} = 0, \forall y. \tag{13}$$

This implies that $g(\phi(y)) - y$ has to be orthogonal to the tangent plane at $g(\phi(y))$. Therefore $g(\phi(y))$ is an orthogonal projection onto g. □

Corollary 5.2. *Global minima of Equation* (5) *converge to principal surfaces as $|Z| \to \infty$.*

Proof. This follows directly from equation (4) and the previous proof. The projection distance is locally optimized, because it is globally minimal. Thus we have all of the conditions for the weak principle surface and additionally that the projections are minimal. □

Globally optimal KMMs are in fact principal surfaces with minimal variance projections, and therefore the generalized equivalent of projecting onto the dominant principal components.

6. Experimental Results

First we quantitatively compare isomap [28], probabilistic principal surfaces (PPS) [7] and the proposed approach on synthetic data.

Isomap does not provide a projection method of unseen data points, hence we project by nearest neighbors interpolation. We select the number of nearest neighbors in two ways—by visual inspection of of isomap residual plot (IM) and by cross validation (IM-CV).

For PPS we use the Matlab implementation from the authors [7]. We augmented the code with a isomap initialization and cross validation and stop the optimization, as for the KMM, when the MSE on the cross validation data set stops decreasing.

The PPS initialization with isomap is done by scaling the isomap embedding to the latent node grid. For each latent node the nearest neighbors in the isomap embedding are computed and the PPS output nodes are temporarily initialized by weighted averaging of the input data corresponding to these nearest neighbors. Based on the temporary output nodes assignments the weight matrix W is initialized by solving a linear least squares system to minimize the mapping of the latent nodes by W to the ambient space with respect to the temporary output nodes. The final initialization of the output nodes of the PPS is then resulting from the mapping of the latent nodes with W obtained by linear least squares. Figure 2 shows that it is essential to initialize PPS with a global method to obtain an accurate parametrization of the manifold.

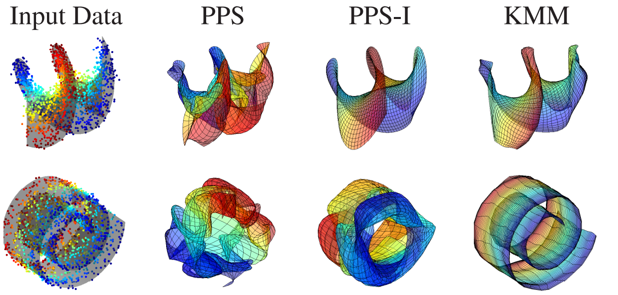


Figure 2. Manifolds obtained by applying PPS, PPS-I and KMM to the input data sampled from the underlying manifold, the corkscrew and the swissroll corresponding to results 6 and 10 in table 1 respectively. The manifolds are colored by the second dimension of the learned parametrization (manifold coordinates) and the original parameter for the input manifold.

We compare against the original (PPS) implementation and the isomap initialized version (PPS-I). We experimented with various settings of the parameters of the PPS, namely the number of latent nodes M, the number of latent basis function L and the α value and report the best results. The number of latent basis functions L effectively constrains the smoothness of the manifold as illustrated in figure 3.

To quantify how well each of the approaches captures the idea of a principal surface we set up a 2 dimensional manifold sampled with added Gaussian noise orthogonal to the manifold. This allows to construct a ground truth data set on the manifold and a test data set by adding Gaussian noise to the ground truth data set. Note that, due to *orthogonal* Gaussian noise, the underlying manifold is truly a principal surface of the distribution the data is sampled from. We then measure how well the principal surface is approximated by the distance between ground truth and the projection of the test data set onto the manifold learned by each approach. This test is performed on the corkscrew and swissroll manifold depicted in the first and second row of figure 2 . The corkscrew is a parametric surface $s(l,h) = (l, \sin(\alpha\pi l)h, \cos(\alpha\pi l)h)^t$, where α controls the number of twists. We set $\alpha = \frac{1}{20}$ and sample l and h uniformly on $[0,40]$. We performed tests with $n = 1000$ and $n = 2000$ samples with different levels of normal $N(0,\sigma^2)$ distributed noise orthogonal to the corkscrew surface, e.g.

$$s_n(l,h) = s(l,h) + N(0,\sigma^2)\frac{\frac{\partial s(h,l)}{\partial l} \times \frac{\partial s(h,l)}{\partial h}}{\|\frac{\partial s(h,l)}{\partial l} \times \frac{\partial s(h,l)}{\partial h}\|} \qquad (14)$$

Similarly the swissroll is the parametric surface $s(r,h) = (r\sin(r), h, r\cos(r))^t$. We sampled r uniformly on $[1,4\pi]$ and h uniformly on $[0,20]$. Again we performed tests with $n = 1000$ and $n = 2000$ samples with different levels of normal $N(0,\sigma^2)$ distributed noise orthogonal to the swissroll surface.

For each configuration we constructed a set of training data consisting of n samples, a cross validation data set of $\frac{n}{2}$ samples, each with $N(0,\sigma^2)$ distributed noise orthogonal to the manifold, a ground truth data set of n samples (without noise), and a test data set of n samples by adding $N(0,\sigma^2)$ distributed noise orthogonal to the ground truth data set.

	IM	IM-CV	PPS	PPS-I	KMM
1.	1.19	0.68	0.31	0.22	**0.18**
2.	3.01	2.13	1.08	**0.57**	0.79
3.	7.30	4.84	2.55	**1.26**	1.63
4.	0.77	0.35	0.12	0.11	**0.10**
5.	1.63	1.67	0.64	0.48	**0.44**
6.	4.66	4.40	2.07	**0.96**	1.24
7.	2.68	1.01	4.73	3.28	**0.41**
8.	2.96	1.33	5.01	3.64	**1.04**
9.	1.01	0.53	2.87	0.45	**0.15**
10.	1.14	0.79	2.82	0.70	**0.19**

Table 1. Projection errors - mean squared error of projected test data to ground truth data. Rows 1 to 3 are corkscrews with $n = 1000$ and $\sigma = 0$,1 and 2. Rows 4 to 6 are corkscrews with $n = 2000$ and $\sigma = 0$,1 and 2. Rows 7 and 8 are swissrolls with $n = 1000$ and $\sigma = 0$ and 0.5. Rows 9 and 10 are swissrolls with $n = 2000$ and $\sigma = 0$ and 0.5.

Table 1 shows the mean of the squared projection errors. Isomap with cross validation degenerates to a single nearest neighbor mapping, which performs quite well in the absence of noise. On the corkscrew the PPS performs slightly better than the KMM in some cases. For the swissroll the KMM performs better than the PPS. The PPS is not capable of finding a good manifold representation of the swissroll, as figure 2 illustrates, because it failed to converge to a smooth surface. The reconstruction error of the PPS can be low, even if the underlying manifold from which the data is sampled is not properly represented, as the results in table 1 in conjunction with figure 2 demonstrates. This example demonstrates the general challenge of evaluating manifold learning algorithms in the absence of visual feedback.

The PPS and the proposed approach, like many other learning algorithms, are prone to over fitting. Due to the noncontinuous projection of the PPS stopping by cross validation is not effective. To avoid over smoothing the number of latent basis functions must be adjusted in a way that depends on the shape of the manifold and the input data. We are not aware of any other option than user input (e.g. visual inspection) for choosing the number of latent basis functions, which is not feasible beyond 3-dimensional input data. Figure 3 shows PPS-I for 3 settings of latent basis functions and KMM with 3 different choices for the number of nearest neighbors which determines the kernel bandwidth as described in section 4.3. For 3 latent basis functions the resulting PPS is too smooth, for 5 an accurate manifold representation is obtained and with 7 the PPS starts to over fit. KMM in conjunction with the proposed optimization strategy is very robust with respect to the choice of the kernel bandwidth.

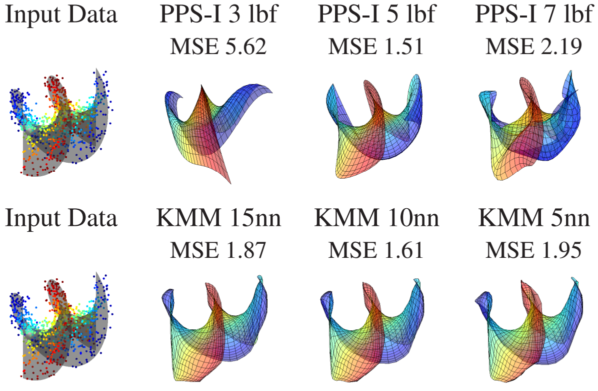


Figure 3. Manifolds for the corkscrew 3 in table 1. 1st row: PPS with isomap initialization and stopped with cross validation for 3, 5 and 7 latent basis functions per dimension. 2nd row: KMM with 15, 10 and 5 nearest neighbors for computing the kernel bandwidth.

In summary the proposed approach compares favorably with the state of the art approaches for manifold learning, including isomap and PPS. PPS can find very accurate manifold representation with a careful tuning of the parameters and initialization with isomap. The proposed approach is more robust with respect to its single input parameter, the number of nearest neighbors for the kernel bandwidth. The kernel bandwidth for the KMM in the coordinate space has a similar effect as the latent basis functions in the PPS, controlling the smoothness of the manifold. By optimizing Z in the coordinate space the KMM approach is adjusting to the smoothness of the underlying manifold — spreading out the free parameters Z in the coordinate space has a virtually the same effect as decreasing the kernel bandwidth. As mentioned in section 4.3 this is prone to over fitting, but the strategy proposed with stopping based on cross validation proves very effective. The theoretical results for KMMs give further assurance that for sufficiently large sample sets the resulting manifolds are good approximations to the data.

Next we applied the KMM to the the Frey faces data set of 1965 images of various facial expressions of the same person. The images are 20×28 pixels with an intensity range from 0 to 255. The isomap residual variance as well as the KMM projection errors indicate that the facial expressions can be captured by approximately 3 degrees of freedom. Therefore we have a 3 dimensional manifold in a 20×28 dimensional space. We split the 1965 images at random into a training set of 1000 images, a cross validation set of 500 images and a test set of 565 images. To each set we add normal distributed noise with zero mean and σ^2 variance. Figure 4 shows projections of the noisy test images onto the learned manifold by KMM.

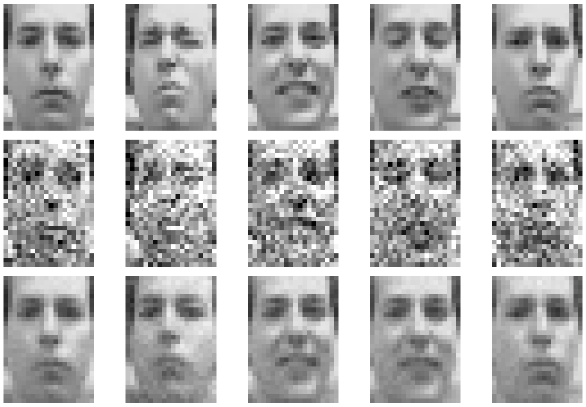


Figure 4. KMM on facial expressions image data set. From top to bottom: original images, noisy images $\sigma = 40$ and images projected by KMM.

Figure 5 shows the coordinate space of the KMM learned from the facial images with $\sigma = 20$ noise added. New images are generated by reconstruction from samples in the coordinate space.

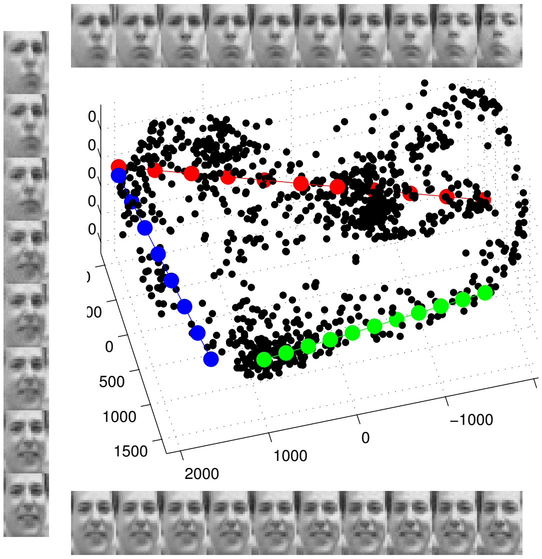


Figure 5. Coordinate space of KMM for Frey faces. Reconstructed images from equally spaced samples along the red line(top), green line (bottom) and blue line (left).

Figure 6 shows an application to motion capture data. The KMM approach is applied to sequences of the different walking styles of subject 132 from the CMU mocap database — a data set of 5000 samples in a 56 dimensional space. Figure 6 shows the coordinate space of the KMM. The KMM accurately captures the different walking styles as illustrated by the reconstructed poses from samples in the coordinate space.

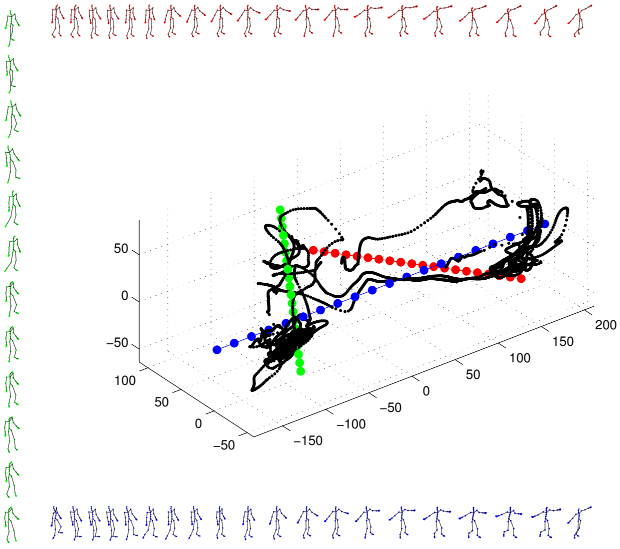


Figure 6. Coordinate space of KMM for motion capture data. Reconstructed poses from equally spaced samples along the red line(top), blue line (bottom) and green line (left).

7. Conclusion

The proposed method for manifold learning, called *kernel map manifolds* provides an explicit formulation of coordinate and reconstruction mapping and allows one to measure projection distance and reconstruction error for a manifold. Many state of the art manifold learning methods often perform validation based on visual inspection. Others validate performance by characterizing classification tasks on the low dimensional representation. The construction of KMMs from an existing representation is straightforward and effective for manifolds of any dimensionality. We furthermore proved that KMMs are in a strict, statistical sense principal surfaces. The experiments show that KMMs are useful in a variety of applications and compare favorable with other methods.

The KMM approach presented provides numerous areas of further improvement. For initialization a recent theorem [14] on parametrization of manifolds by Laplacian eigenfunctions suggest an interesting approach. The theorem states that there exists a set of eigenfunctions of the Laplace-Beltrami operator on a manifold which will in a local neighborhood parametrize the manifold. This result invites to select eigenvectors locally that give low projection error and use those for initialization. In the optimization stage more sophisticated approaches could be employed, for example a simulated annealing related approach with decreasing kernel bandwidth during optimization. Another

Acknowledgments. This work was supported by the NIH/NCBC grant U54-EB005149 and the NSF grant CCF-073222.

References

[1] O. Arandjelović and R. Cipolla. A manifold approach to face recognition from low quality video across illumination and pose using implicit super-resolution. *ICCV*, October 2007.

[2] M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295, Jan 2002.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[4] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *In Advances in Neural Information Processing Systems*, pages 177–184. MIT Press, 2004.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] M. Brand. Charting a manifold. In *NIPS*, 2002.

[7] K.-Y. Chang and J. Ghosh. A unified model for probabilistic principal surfaces. *IEEE TPAMI*, 23(1):22–41, 2001.

[8] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Monographs on Statistics and Applied Probability 59. Chapman and Hall, London, 1994.

[9] D. Donoho and C. Grimes. Hessian eigenmaps: locally linear embedding techniques for high dimensional data. *Proc. of N. A. of Sciences*, 100(10):5591–5596, 2003.

[10] P. Etyngier, F. Segonne, and R. Keriven. Shape priors using manifold learning techniques. *ICCV*, Oct. 2007.

[11] F. Guo and G. Qian. 3d human motion tracking using manifold learning. *ICIP 2007*, 1:I –357–I –360, 16 2007-Oct. 19 2007.

[12] T. Hastie. Principal curves and surfaces. *Ph.D Dissertation*, 1984.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, July 2006.

[14] P. W. Jones, M. Maggioni, and R. Schul. Manifold parametrizations by eigenfunctions of the laplacian and heat kernels. *PNAS*, 105(6):1803–1808, 2008.

[15] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(3):281–297, 2000.

[16] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, 1997.

[17] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *In NIPS*, 2004.

[18] C. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *ICCV*, pages 1–8, 2007.

[19] T. Martinetz and K. Schulten. Topology representing networks. *Neural Netw.*, 7(3):507–522, 1994.

[20] P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal surfaces from unsupervised kernel regression. *IEEE TPAMI*, 27(9):1379–1391, 2005.

[21] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9(1):141–142, 1964.

[22] J. S. Perantonis and v. Virvilis. Dimensionality reduction using a novel neural network based feature extraction method. *IJCNN*, 2:1195–1198, 1999.

[23] R. Pless. Image spaces and video trajectories: Using isomap to explore video sequences. In *ICCV*, page 1433, Washington, DC, USA, 2003. IEEE Computer Society.

[24] R. Pless and I. Simon. Using thousands of images of an object. In *Computer Vision, Pattern Recognition and Image Processing*, 2002.

[25] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(550), 2000.

[26] A. J. Smola, S. Mika, B. Schölkopf, and R. C. Williamson. Regularized principal manifolds. *J. Mach. Learn. Res.*, 1:179–209, 2001.

[27] R. Souvenir and R. Pless. Manifold clustering. In *ICCV*, pages 648–653, 2005.

[28] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(550):2319–2323, 2000.

[29] R. Tibshirani. Principal curves revisited. *Statistics and Computing*, 2:183–190, 1992.

[30] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML*, page 106, New York, NY, USA, 2004. ACM Press.

[31] J. Yang, F. Li, and J. Wang. A better scaled local tangent space alignment algorithm. In *IJCNN*, volume 2, pages 1006–1011, 2005.

[32] Q. Zhang, R. Souvenir, and R. Pless. On manifold structure of cardiac mri data: Application to segmentation. In *CVPR 2006*, pages 1092–1098. IEEE, 2006.

[33] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.*, 26(1):313–338, 2005.