Local Minima Embedding

Minyoung Kim MIKIM@CS.RUTGERS.EDU
Fernando De la Torre FTORRE@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University, PA 15213, USA

Abstract

Dimensionality reduction is a commonly used step in many algorithms for visualization, classification, clustering and modeling. Most dimensionality reduction algorithms find a low dimensional embedding that preserves the structure of high-dimensional data points. This paper proposes Local Minima Embedding (LME), a technique to find a low-dimensional embedding that preserves the local minima structure of a given objective function. LME provides an embedding that is useful for visualizing and understanding the relation between the original variables that create local minima. Additionally, the embedding can potentially be used to sample the original function to discover new local minima. The function in the embedded space takes an analytic form and hence the gradients can be computed analytically. We illustrate the benefits of LME in both synthetic data and real problems in the context of image alignment. To the best of our knowledge this is the first paper that addresses the problem of finding an embedding that preserves local minima properties of an objective function.

1. Introduction

Optimization algorithms occupy a central role within the arsenal of computational methods used for solving problems in the fields of machine learning, statistics, computer vision, and pattern recognition. In particular, many machine learning algorithms can be cast as optimization problems. A major challenge in optimization is to find the global minimum and to understand the structure of local minima of a given problem.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

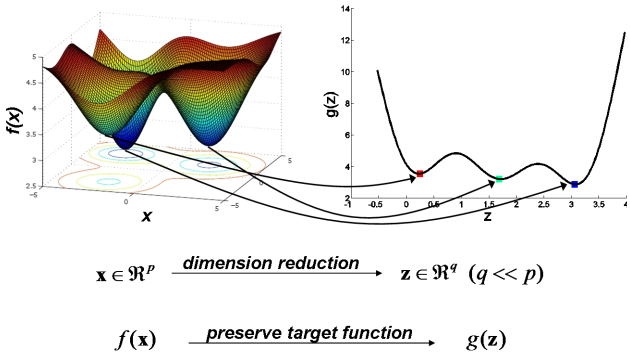


Figure 1. Illustration of local minima embedding (LME). LME finds both an embedding ($\mathbf{z}$) of the input space ($\mathbf{x}$) and a new target function (g($\mathbf{z}$)) defined on the embedded space that preserves the local minima structure of $f(\mathbf{x})$. The three local minima of the original function ($f(\mathbf{x})$) are preserved in $g(\mathbf{z})$.

Motivated by the need of visualizing and understanding the structure of local minima, in this paper, we consider the problem of finding an embedding an error function that preserves local minima properties. Fig. 1 illustrates the main idea of the paper. The objective function ($f(\mathbf{x})$) defined on the 2D parameter space (left plot) contains three local minima. After performing our 1D embedding, the new objective function (right plot) now defined on the lower dimensional embedded space ($\mathbf{z}$), preserves the local minima structure of the original function. We formulate the task of local minima preserving embedding (denoted by LME) as follows: Given a real-valued function $f(\mathbf{x})$ on the input space $\mathbf{x} \in \mathbb{R}^p$, we find the low-dimensional embedding of $\mathbf{x}$, denoted by $\mathbf{z} \in \mathbb{R}^q$ ($q \ll p$), while simultaneously approximating $f(\mathbf{x})$ to a real-valued function $g(\mathbf{z})$ so that it preserves the local minima properties of $f(\mathbf{x})$ as much as possible. LME is useful for visualization as well as doing a gradient search in the low dimensional space (e.g., 1D search), which typically can be done more efficiently than in the input space. Moreover, the embedding can potentially capture underlying redundancies or dependencies that reside in the original variables, providing a chance to explore potentially unobserved local minima points.

The rest of the paper is organized as follows: after reviewing some related work on high-dimensional data embedding in Sec. 2 and defining the notation used throughout the paper in Sec. 3, we propose our LME algorithm in Sec. 4. Experimental results on synthetic and real data are illustrated in Sec. 5.

2. Related Work

A large amount of literature on dimensionality reduction (DR) has been devoted to the classical unsupervised and supervised settings. In the former, discovering a low dimensional structure of the data can often be accomplished by either extracting global statistical information (e.g., principal directions of PCA and kernel PCA (Schölkopf et al., 1998)), or by exploiting the geometric nature of data (e.g., local linear structures of LLE (Roweis & Saul, 2000), geodesic distances of ISOMAP (Tenenbaum et al., 2000), locality preserving projection (LPP) (He & Niyogi, 2003).

In supervised dimensionality reduction, each data point is marked with additional label information that guides the formation of the low-dimensional embedded space. When the label indicates a discrete class membership, such a class label can be exploited to enforce that data points are either grouped together or far apart from one another in the embedded space. The well-known Fisher discriminant analysis and the diverse forms of metric learning algorithms (Globerson & Roweis, 2005; Weinberger et al., 2005; Xing et al., 2002; Yan et al., 2007) fall into this category. On the other hand, in certain applications it is reasonable to regard the label as a smoothly varying real-value. The DR in this setting, often referred to as dimensionality reduction for regression, can be treated within a regression framework where we regress a target (output label) from input data points. The embedding then tries to minimize the loss of information caused by dimensionality reduction in terms of the regression performance, often achieved by enforcing conditional independence between output and input given the embedding (Fukumizu et al., 2004; Kim & Pavlovic, 2008; Li, 1991; Nilsson et al., 2007; Wu, 2008).

A relatively unexplored problem in DR is finding embeddings that preserve local minima of a given objective function. The closest to our work is the structure preserving embedding (Shaw & Jebara, 2009) whose goal is to preserve the nearest neighbor structure of the input graph. They explicitly enforce affinity (neighbors) and repulsive (non-neighbors) constraints to fully respect the original graph topology. Recent related work in computer vision aims to perform feature selection/weighting in generative (Nguyen & de la Torre, 2008) and dicriminative (Wu et al., 2008) models that avoid local minima in image alignment. Unlike previous work on dimensionality reduction on data samples or graph structures, this paper proposes a dimensionality reduction technique to preserve the local minima structure of a given objective function.

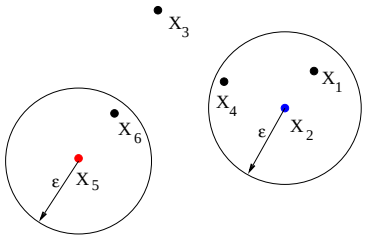


Figure 2. Example illustrating the neighborhood topology.

3. Notation and Setup

We denote by $f(\mathbf{x})$ the target function in the original space. For simplicity and tractability, we assume that we are given a set of n paired samples $\mathcal{D} = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$. We use the boldfaced matrix/vector notation, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$, which are of dimension $(n \times p)$ and $(n \times 1)$, respectively.

We assume that $\mathcal{D}$ contains three types of samples: (i) some of the local minima points of the target function $f(\cdot)$, (ii) their neighborhood points obtained by random sampling around the local minima, and (iii) some other non-neighbor points. These points are all labeled, meaning that we know which data points are local minima, neighbors of local minima, or non-neighbors. We define a *neighbor* as a point $\mathbf{x}$ which is close to a local minimum $\mathbf{x}_*$ in the original space (i.e., $||\mathbf{x}-\mathbf{x}_*|| < \epsilon$ for some small ϵ). The set of neighbors of $\mathbf{x}_*$ within the radius ϵ is denoted by $\mathcal{N}_\epsilon(\mathbf{x}_*)$. By definition, the function value of a neighbor is not smaller than that of the local minimum (i.e., $f(\mathbf{x}_*) \leq f(\mathbf{x})$). For simplicity we restrict ourselves to disjoint neighborhoods, meaning that $\mathcal{N}_\epsilon(\mathbf{x}_*) \cap \mathcal{N}_\epsilon(\mathbf{x}'_*) = \emptyset$ for any two local minima $\mathbf{x}_*$ and $\mathbf{x}'_*$.

Given $\mathcal{D}$ and the neighborhood threshold constant ϵ, we represent the neighborhood topology as the $(n \times n)$ matrix $\mathbf{U}$ which is a 0/1 matrix with $\mathbf{U}_{ji} = 1$ iff $\mathbf{x}_i$ is a local minimum and $\mathbf{x}_j \in \mathcal{N}_\epsilon(\mathbf{x}_i)$. We also define the diagonal matrix $\mathbf{V}$ whose diagonal entries are the row sums of $\mathbf{U}$. For better understanding, consider the synthetic example in Fig. 2 where $\mathcal{D}$ contains 6 points with two local minima $\mathbf{x}_{2,5}$ in blue/red color, their neighbors $\mathbf{x}_{1,4,6}$, and a non-neighbor $\mathbf{x}_3$. In this case, it is easy to see that the neighborhood topology

matrix $\mathbf{U}$ and the row-sum matrix $\mathbf{V}$ are:

$$\mathbf{U} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

$\mathbf{U}$ and $\mathbf{V}$ facilitate expressing the local minima constraints (i.e., $f(\mathbf{x}_*) \leq f(\mathbf{x})$ for $\mathbf{x} \in \mathcal{N}_\epsilon(\mathbf{x}_*)$) compactly as a single linear inequality system, namely $(\mathbf{U} - \mathbf{V})\mathbf{f} \leq 0$.

4. Local Minima Embedding (LME)

LME learns two mappings: (i) A low-dimensional embedding $\mathbf{z} = \mathbf{B}^\top \mathbf{x}$ where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_q]$ is the $(p \times q)$ embedding matrix[1], and $\mathbf{z} \in \mathbb{R}^q$ is the q-dimensional ($q \ll p$) embedded point of $\mathbf{x}$. (ii) A new target function $g(\mathbf{z})$ on the embedded space, which approximates $f(\mathbf{x})$ while preserving the local minima properties of the original function $f(\cdot)$. We assume a linear form $g(\mathbf{z}) = \mathbf{c}^\top \mathbf{z}$ with the parameter $\mathbf{c} \in \mathbb{R}^q$. Note that as a linear function does not entail a local minimum, we *should* kernelize it (now a kernel on the embedded space). Also notice that this roughly corresponds to performing a kernel regression with the data $\{(\mathbf{z}_i, f(\mathbf{x}_i))\}_{i=1}^n$. However, as shown in this section, we do not solve these two problems separately, but optimize them simultaneously in a principled manner.

4.1. Constraints for LME

Given a set of n paired samples $\mathcal{D} = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$, LME finds a low-dimensional embedding that satisfies three criteria:

- **Condition I (Local Structure Preservation)**: The embedding preserves the locality (or proximity) structure of local minima points and their neighborhood points. That is, if a point is close to a local minimum in the original space, so is it in the embedded space.
- **Condition II (Local Minima Preservation)**: Within each neighborhood around a local minimum, the target function $g(\cdot)$ has to satisfy the local minima property. More specifically, when we denote by $\mathbf{x}_*$ ($\mathbf{z}_*$) and $\mathbf{x}$ ($\mathbf{z}$), a local minimum and its neighbor in the original (embedded) space, respectively, one can enforce that $g(\mathbf{z}_*) \leq g(\mathbf{z})$ as well as the stationary point condition $\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\big|_{\mathbf{z}=\mathbf{z}_*} = 0$. We also need to avoid that several local minima collapse in a single point.
- **Condition III (Function Approximation)**: Overall, $g(\mathbf{z})$ should approximate $f(\mathbf{x})$ well.

4.1.1. Enforcing Condition I

To enforce condition I, namely preserving the neighbor structures around the local minima points, LME enforces the following constraint:

$$\begin{aligned} &\text{For every local minimum } \mathbf{x}_*, \\ &\quad \mathbf{x} \in \mathcal{N}_\epsilon(\mathbf{x}_*) \Leftrightarrow \mathbf{z} \in \mathcal{N}_\rho(\mathbf{z}_*), \\ &\quad \text{where } \mathbf{z} = \mathbf{B}^\top \mathbf{x} \text{ and } \mathbf{z}_* = \mathbf{B}^\top \mathbf{x}_*. \end{aligned} \tag{1}$$

Here, the neighborhood set in the embedded space, $\mathcal{N}_\rho(\mathbf{z}_*)$, is defined for a new radius constant ρ. A straightforward way to enforce the above constraint would be to have inequality constraints as follows:

$$\begin{aligned} &||\mathbf{B}^\top \mathbf{x} - \mathbf{B}^\top \mathbf{x}_*||^2 \leq \rho \leq ||\mathbf{B}^\top \mathbf{x}' - \mathbf{B}^\top \mathbf{x}_*||^2, \\ &\quad \text{for all } \mathbf{x} \in \mathcal{N}_\epsilon(\mathbf{x}_*) \text{ and } \mathbf{x}' \notin \mathcal{N}_\epsilon(\mathbf{x}_*). \end{aligned} \tag{2}$$

However, (2) has the form of *differences of quadratic functions*, which can yield non-convex constraints in the optimization of $\mathbf{B}$. Instead, one can consider a soft version of neighborhood preserving by penalizing pairs of data points which have higher proximity in the original space, but lower proximity in the embedded space. This can be expressed as:

$$\min_{\mathbf{B}} \sum_{i,j} w_{ij} ||\mathbf{z}_i - \mathbf{z}_j||^2, \tag{3}$$

where w_{ij} is the measure of affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the original space, having a larger value if $\mathbf{x}_i$ and $\mathbf{x}_j$ are closer. Note that (3) has the same form[2] as that of the *locality preserving projection* (He & Niyogi, 2003) and the *Laplace eigenmap* (Belkin & Niyogi, 2002). There are several diverse ways to construct the affinity matrix $\mathbf{W} = (w_{ij})$. Throughout the paper, we assume that $\mathbf{W} = \mathbf{K}_\mathbf{x}$, the kernel matrix in the original space. Using the matrix notation $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]^\top = \mathbf{XB}$, (3) is equivalent to:

$$\min_{\mathbf{B}} \operatorname{tr}(\mathbf{Z}^\top \mathbf{L} \mathbf{Z}) = \operatorname{tr}(\mathbf{B}^\top \mathbf{X}^\top \mathbf{L} \mathbf{X} \mathbf{B}), \tag{4}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian induced from $\mathbf{W}$, and $\mathbf{D}$ is the diagonal matrix with row-sum entries of $\mathbf{W}$.

[1] One can always consider a kernel extension, in which $\mathbf{B}$ becomes an operator composed of q (nonlinear) functions in the RKHS of $\mathbf{x}$. For expositional convenience, we use the linear representation throughout the paper, which is then turned into a kernel version straightforwardly using the kernel trick.

[2] As is done in LPP, one typically needs to impose certain regularization on $\mathbf{B}$ to avoid the trivial solution of the constant $\mathbf{Z}$ (or $\mathbf{B} = 0$). In our case, however, since this trivial case is automatically discarded by the least-square objective (8), we do not explicitly consider it.

4.1.2. Enforcing Condition II

In condition II, we enforce the local minima criteria for a new target function $g(\cdot)$ within the neighborhoods of local minima points. We consider two conditions: (i) minimality within a neighborhood, and (ii) stationary point condition. The former can be formally stated as:

$$\text{For every local minimum } \mathbf{x}_* \text{ and } \mathbf{x} \in \mathcal{N}_\epsilon(\mathbf{x}_*),$$
$$g(\mathbf{z}_*) \leq g(\mathbf{z}). \tag{5}$$

As $g(\mathbf{z}) = \mathbf{c}^\top \mathbf{z}$, the inequality constraints (5) can be equivalently expressed in a matrix form as:

$$(\mathbf{U} - \mathbf{V})\mathbf{Z}\mathbf{c} \leq 0. \tag{6}$$

Note that (6) forms a set of nonlinear and non-convex constraints as we optimize for both $\mathbf{c}$ and $\mathbf{Z}$.

The latter stationary point condition can be written as: $\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_*} = 0$. As there always exists a numerical error within machine precision, we introduce a slack variable τ, a small positive number. Then

$$\left|\left| \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_*} \right|\right|^2 \leq \tau. \tag{7}$$

Unfortunately, (7) is not a convex constraint in general due to the nonlinearity of $g(\mathbf{z})$ in $\mathbf{z}$.

Sometimes, we have observed that the local minima points (and their neighbors) are collapsed onto one another in their embedding. To address this issue, we additionally enforce the following constraint for every pair of local minima points $(\mathbf{x}_*, \mathbf{x}'_*)$: $||\mathbf{z}_* - \mathbf{z}'_*|| \geq \delta$, where δ is chosen appropriately (e.g., $\delta = 1$).

4.1.3. Enforcing Condition III

Finally, the last condition forces the new target function $g(\cdot)$ to approximate the original function $f(\cdot)$. In the least-squares sense, this can be written as:

$$\min \sum_i ||g(\mathbf{z}_i) - f(\mathbf{x}_i)||^2 = ||\mathbf{Z}\mathbf{c} - \mathbf{f}||^2. \tag{8}$$

It is worth noticing that in (8) we enforce $g(\cdot)$ to be close to $f(\cdot)$ for *all* the training data points, which has an auxiliary effect of imposing the ordering constraints on the function values at the local minima points. For instance, consider two local minima points $\mathbf{x}_*$ and $\mathbf{x}'_*$ where $\mathbf{x}_*$ is the global minimum while $\mathbf{x}'_*$ is not. Then it is desirable to have the function values ordered accordingly, namely $g(\mathbf{z}_*) \leq g(\mathbf{z}'_*)$. This can be enforced either explicitly by a set of inequalities similar to (6) or implicitly by (8). We take the latter approach.

4.2. Energy function for LME

Combining all the constraints from section 4.1, (4), (6), (7), and (8), LME optimizes (with $\mathbf{Z} = \mathbf{X}\mathbf{B}$):

$$\begin{aligned} \min_{\mathbf{B},\mathbf{c}} \quad & ||\mathbf{X}\mathbf{B}\mathbf{c} - \mathbf{f}||^2 + \lambda \text{tr}(\mathbf{B}^\top \mathbf{X}^\top \mathbf{L} \mathbf{X} \mathbf{B}) \\ \text{s.t.} \quad & (\mathbf{U} - \mathbf{V})\mathbf{X}\mathbf{B}\mathbf{c} \leq 0 \\ & \left|\left| \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_*} \right|\right|^2 \leq \tau,\ ||\mathbf{z}_* - \mathbf{z}'_*||^2 \geq \delta. \end{aligned} \tag{9}$$

Here λ is the trade-off parameter that balances two cost terms. As mentioned earlier, it is necessary to kernelize the target function $g(\cdot)$. We consider two RBF kernels, one for the original space and the other for the embedded space:

$$k_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\tfrac{1}{2\sigma_{\mathbf{x}}^2}||\mathbf{x}_i - \mathbf{x}_j||^2\big) \text{ and}$$
$$k_{\mathbf{z}}(\mathbf{z}_i, \mathbf{z}_j) = \exp\big(-\tfrac{1}{2\sigma_{\mathbf{z}}^2}||\mathbf{z}_i - \mathbf{z}_j||^2\big),$$

where the scale parameters $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{z}}$ are assumed fixed. We denote their kernel matrices evaluated on the training data by $\mathbf{K}_{\mathbf{x}}$ and $\mathbf{K}_{\mathbf{z}}$, respectively, which are of dimension $(n \times n)$.

Using the dual representation, the embedding operator $\mathbf{B}$ can be parameterized by the $(n \times q)$ matrix $\boldsymbol{\alpha} = (\alpha_{ij})$, namely $\mathbf{B} = [b_1(\cdot), \ldots, b_q(\cdot)]$, where $b_j(\cdot) = \sum_{i=1}^n \alpha_{ij} k_{\mathbf{x}}(\cdot, \mathbf{x}_i)$ for $j = 1, \ldots, q$. This, by the kernel trick, leads to the nonlinear embedding expressed as $\mathbf{Z} = \mathbf{K}_{\mathbf{x}}\boldsymbol{\alpha}$, while replacing the second term in the objective (9) by $\text{tr}(\boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{x}} \mathbf{L} \mathbf{K}_{\mathbf{x}} \boldsymbol{\alpha})$.

Similarly, the target function $g(\cdot)$ can be represented in a dual form in the embedded space as:

$$g(\mathbf{z}) = \sum_{i=1}^n \beta_i k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}_i), \tag{10}$$

parameterized by the $(n \times 1)$ vector $\boldsymbol{\beta} = (\beta_i)$. It should be noted that the functional form of (10) sets an upper bound on the number of local minima in the new function. That is, $g(\mathbf{z})$ has *at most* n local minima points in the embedded space, which is reasonable given a finite sample scenario. The kernel trick replaces $\mathbf{Z}\mathbf{c}$ by $\mathbf{K}_{\mathbf{z}}\boldsymbol{\beta}$, which yields the kernelized version of (9):

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \frac{1}{2}||\mathbf{K}_{\mathbf{z}}\boldsymbol{\beta} - \mathbf{f}||^2 + \lambda \text{tr}(\boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{x}} \mathbf{L} \mathbf{K}_{\mathbf{x}} \boldsymbol{\alpha}) \\ \text{s.t.} \quad & (\mathbf{U} - \mathbf{V})\mathbf{K}_{\mathbf{z}}\boldsymbol{\beta} \leq 0 \\ & \left|\left| \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_*} \right|\right|^2 \leq \tau,\ ||\mathbf{z}_* - \mathbf{z}'_*||^2 \geq \delta. \end{aligned} \tag{11}$$

As $\mathbf{K}_{\mathbf{z}}$ is nonlinearly related to $\boldsymbol{\alpha}$ through the RBF function, (11) is an instance of non-convex optimization with non-convex inequality constraints. We solve this problem using the constrained optimization toolbox in Matlab with the function `fmincon()`.

Once we have found the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the embedding of a new test point $\mathbf{x}$ can be obtained from:

$$\mathbf{z} = \boldsymbol{\alpha}^\top \mathbf{k}_{\mathbf{x}}(\mathbf{x}), \tag{12}$$

where $\mathbf{k}_{\mathbf{x}}(\mathbf{x}) = [k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_1), \ldots, k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_n)]^\top$ is the $(n \times 1)$ test kernel vector for $\mathbf{x}$.

5. Experiments

This section demonstrates the effectiveness of LME for visualization and local minima discovery. In the first experiment we tested the ability of LME to find an embedding that preserves the local minima of an analytic function. In the second experiment LME found an embeeding of the error function used for template matching over translation, rotation and scale.

5.1. Synthetic data

We performed a synthetic experiment on the three hump camel function, defined as

$$y = \frac{1}{6}x_1^6 - 1.05x_1^4 + 2x_1^2 + x_2^2 - x_1x_2. \qquad (13)$$

It has 3 local minima in the 2D space, where one of them is the global minimum taking a strictly smaller function value than the remaining two local minima. Fig. 3(a) and Fig. 3(b) depict the function and the contour plot, respectively. The training data $\mathcal{D}$ is composed of all local minima points (depicted as squares), 20 randomly sampled neighbor points (crosses) per each local minimum within a ball of radius 0.4, and 30 non-neighbor samples (black dots with sample IDs). We used two RBF kernels for nonlinear mappings, $\mathbf{B}$ and $g(\cdot)$, and set the parameters as $\sigma_{\mathbf{x}} = \sigma_{\mathbf{z}} = 1.0$. Starting from random $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as initial iterates, we optimized (11) using `fmincon()` in Matlab until convergence. We found the 1D embedding and the transformed target function as shown in Fig. 3(c), where the green curve represents the embedded function $g(\mathbf{z})$ obtained by LME (10). LME successfully preserves the same local minima of the original function.

We also compared LME with two baseline approaches that perform embedding of the function (KPCA and KLPP (He & Niyogi, 2003)) followed by a least-square fitting of the objective function in the embedded spaces. As shown in Fig. 3(d) and Fig. 3(e), both independent embedding approaches fail to preserve the local minima structure of the original function.

5.2. Template matching

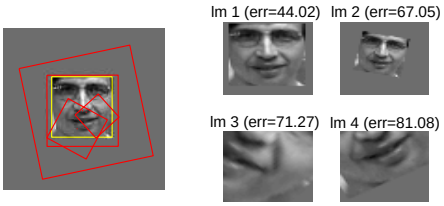
This section tested the LME on the error function for template matching. Given an image frame I and a template model I_0, template matching minimizes:

$$\min_{\mathbf{p}} ||I(\omega(\mathbf{x};\mathbf{p})) - I_0||_2^2, \qquad (14)$$

where $\omega(\mathbf{x};\mathbf{p})$ is a geometric transformation (e.g. affine) that transforms the pixel coordinates $\mathbf{x}$ into the warped ones by the parameters $\mathbf{p}$. We used a Euclidean transformation composed by 4 parameters ($\mathbf{p}$): 2 translation parameters (`x-pos` and `y-pos`), 1 scale parameter (`scale`), and in-plane rotation parameter (`rotation`).

(a) Image frame (b) Template model

Figure 4. (a) Image frame (128×128 pixels). (b) Template patch 48×48 pixels.



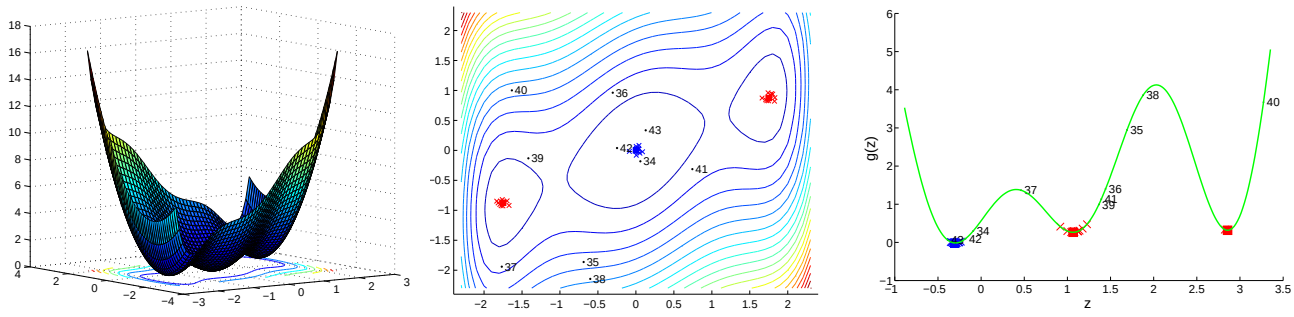
(a) Local minima (b) Images & errors

Figure 5. (a) 4 local minima points depicted as red boxes with different positions, scales, and rotations found by sampling and gradient search. The global minimum is depicted as a yellow box, while the local minima are in red. (b) Images and sum-of-squared errors of the 4 local minima.

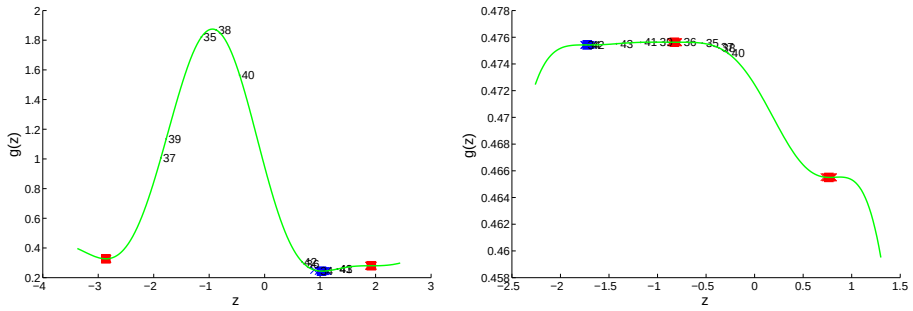
We used Fig. 4(a) as an image frame I of size (128×128) which contains a face template image patch I_0 (Fig. 4(b)) at its center. The template patch is of size (48×48). The image frame has uniform gray background, where the background intensity is set to the average intensity of the template patch.

Note that the error function is defined as the sum-of-squared-distances (SSD) on the 4D input space $\mathbf{x}$. To find some local minima, we did numerical gradient search around some randomly chosen starting points. We found 4 local minima points (lm-1 $\sim$ lm-4) depicted as the red boxes in Fig. 5(a), as well as the global minimum depicted as the yellow box. We also computed the SSD for these 4 local minima w.r.t. the template patch as shown in Fig. 5(b). For each local (and global) minima point, we randomly sampled 8 neighbors taking into account different granularities for different parameters. We also randomly sampled 20 non-neighboring points. LME was applied to reduce the original 4D input to 1D. The resulting embedding and the transformed target error function is shown in Fig. 6. As shown in the figure, the embedding preserves all the local minima points of the given target function observed in the data samples. Interestingly, the embedding reflects a new local minimum point (originally not observable in the available samples). The new point[3] has a lower SSD error than the

[3]We note that this point does not necessarily correspond



(a) Original function (b) Contour plot (c) LME



(d) Kernel PCA (e) Kernel LPP

Figure 3. Three hump camel function. Blue square = global minimum, red squares = (non-global) local minima, crosses = neighbor samples, and black dots (with sample ID numbers) = non-neighbor samples.

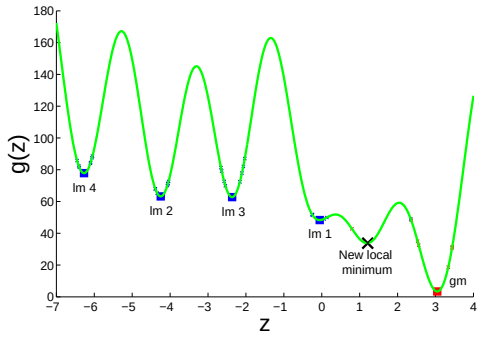


Figure 6. LME for the template matching example. The four local minima points are represented in blue squares, and the global minimum in red. The newly found local minimum point is depicted as a black cross.

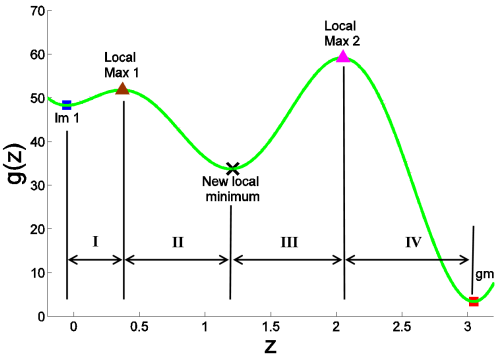


Figure 7. Zoom of Fig. 6 from `lm 1` to `gm`. We split the line search into four regions: `lm 1`∼`local max 1` (I), `local max 1`∼`New local minimum` (II), `New local minimum`∼`local max 2` (III), and `local max 2`∼`gm` (IV). The line search results are shown in Fig. 8.

other local minima points, and the corresponding image can be obtained by solving a pre-image problem. This image is shown in Fig. 8(b), under the title `New LM`. As shown, the image corresponding to the new local minimum is better aligned than the best local minimum (lm-1). This is an example where LME is able to discover the structure of the input w.r.t the error function, potentially yielding new local minima.

Furthermore, to see the effectiveness of LME in finding the dependencies in the original parameters, we did a one dimensional line search in the embedded space around the regions adjoining the global minimum. More specifically, we took the region from the best local minimum in the training samples (i.e., lm-1) to the global minimum (i.e., gm). This region is zoomed in and shown in Fig. 7. We split the region

to a local minimum in the original space. But it turned out to be a local minimum along a particular direction. These types of important points (e.g., saddle points) can be useful for understanding the impact of input variables on the error function, although more research needs to be done to exactly establish the relationship between the minima in the embedded space and those in the original space.

into four segments knotting on the local/global minima/maxima points (I $\sim$ IV). For each of these four segments, we took 5 points (the two ending knots and the three uniformly sampled points between the ending points). We solved the pre-image problem (with the initial points chosen from the previous solutions) to retrieve the corresponding parameters in the original 4D input space. The pre-image provides the visual image for these parameters. The results (parameters and images) retrieved by the line search for each of four segments are shown in Fig. 8.

Observe that most local minima are produced by rotation and scale, which are well known to be worse conditioned than translation parameters for registration problems. For example, for region I, we rotated the lm-1 image counter-clockwise to get the local max 1. Region II, that goes from the local maxima to the new local minimum, corresponds to a rotation clockwise plus a decrease in the scale parameter. This yields a new local minima point, better aligned than lm-1. Continuing these rotation/scale changes, however, led to increasing errors, and we denote as `LMax-2` the new local maximum point shown in Fig. 8(c). In region IV, we again reversed the trend in both scale and rotation parameters, and finally land at the global minimum as shown in Fig. 8(d). This example shows how LME allows to visualize the structure of the local minima and understand the parameters in the original space that create local minima.

6. Conclusion

This paper proposes LME, an embedding technique that preserves the local minima structure of a given error function. We have shown in synthetic and real examples that LME is a useful technique to understand the structure of high-dimensional error functions. Despite a promising technique LME has a few issues that need to be further explored in future work. Firstly, given the embedded point, we need to solve the pre-image problem to find the original parameter in the original space. The pre-image problem is sensible to local minima because there is no one-to-one mapping between the original input space and the embedded space. In future work, we plan to convexify it by formulating a learning problem in the kernel (Hilbert) feature space directly. Secondly, throughout the paper we have assumed that the locations of the local minima are known. We left for future work to address the unsupervised problem where the locations of the local minima are not known.

References

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering, NIPS 2002.

Fukumizu, K., Bach, F., and Jordan, M. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *JMLR* 2004.

Globerson, Amir and Roweis, Sam. Metric learning by collapsing classes, NIPS 2005.

He, Xiaofei and Niyogi, Partha. Locality preserving projections, NIPS 2005.

Kim, Minyoung and Pavlovic, Vladimir. Dimensionality reduction using covariance operator inverse regression, CVPR 2008.

Li, K.-C. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 1991.

Nguyen, M. and de la Torre, F. Local minima free parameterized appearance models, CVPR 2008.

Nilsson, J., Sha, F., and Jordan, M. Regression on manifolds using kernel dimension reduction, ICML 2007.

Roweis, Sam and Saul, Lawrence. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, 2000.

Schölkopf, B., Smola, A. J., and Muller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Shaw, B. and Jebara, T. Structure preserving embedding, ICML 2009.

Tenenbaum, J. B., Silva, V. De, and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

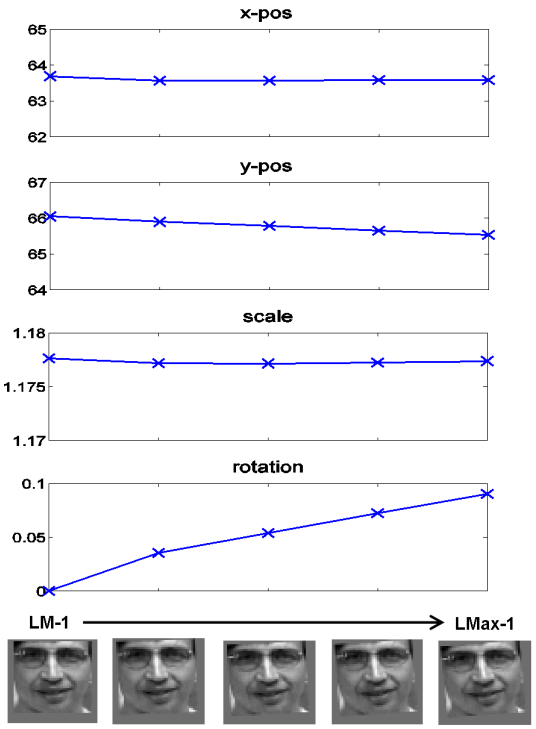
Weinberger, K., Blitzer, J., and Saul, L. Distance metric learning for large margin nearest neighbor classification, NIPS 2005.

Wu, H. M. Kernel sliced inverse regression with applications to classification. *Journal of Computational and Graphical Statistics*, 17(3):590–610, 2008.

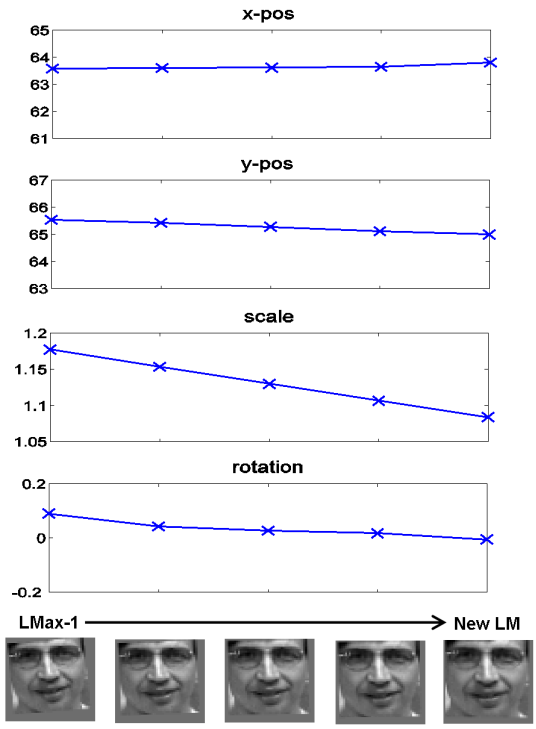
Wu, Hao, Liu, Xiaoming, and Doretto, Gianfranco. Face alignment via boosted ranking model. In CVPR 2008.

Xing, Eric P., Ng, Andrew Y., Jordan, Michael I., and Russell, Stuart. Distance metric learning with application to clustering with side-information, NIPS 2002.

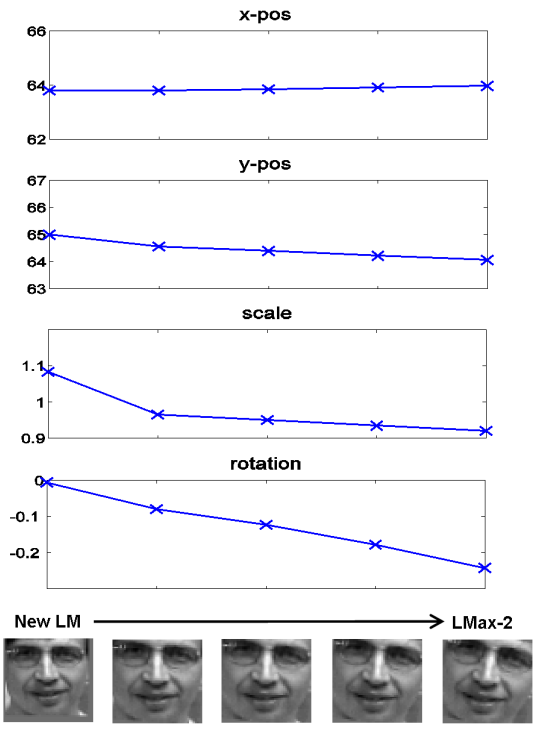
Yan, Shuicheng, Xu, Dong, Zhang, Benyu, Zhang, Hong-Jiang, Yang, Qiang, and Lin, Stephen. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.



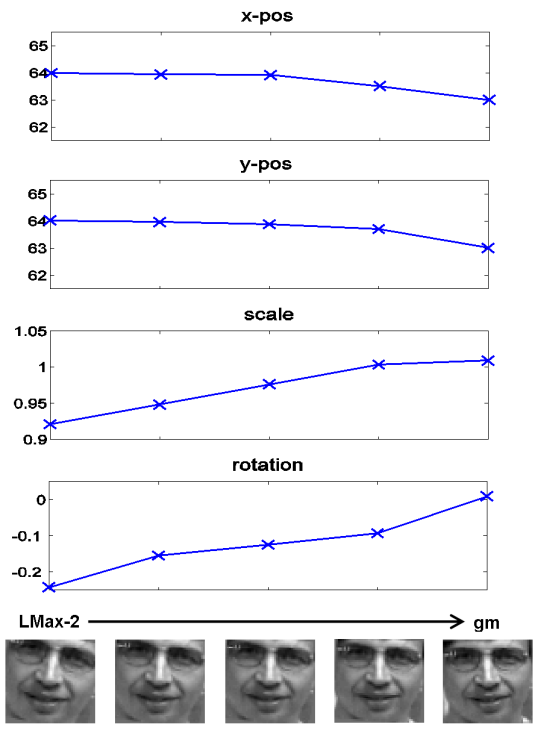
(a) lm 1 to lmax 1 (region I)



(b) lmax 1 to lm new (region II)



(c) lm new to lmax 2 (region III)



(d) lmax 2 to gm (region IV)

Figure 8. Line search along the 4 regions (from I to IV): (a) `lm 1`∼`local max 1` (I), (b) `local max 1`∼`New local minimum` (II), (c) `New local minimum`∼`local max 2` (III), and (d) `local max 2`∼`gm` (IV).