Data Visualization and Dimensionality Reduction using Kernel Maps with a Reference Point

Johan A.K. Suykens

K.U. Leuven, ESAT-SCD/SISTA
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Tel: 32/16/32 18 02 Fax: 32/16/32 19 70
Email: johan.suykens@esat.kuleuven.be

Published in
IEEE Transactions on Neural Networks,
Vol.19, No.9, pp.1501-1517, Sept. 2008

Abstract

In this paper a new kernel based method for data visualization and dimensionality reduction is proposed. A reference point is considered corresponding to additional constraints taken in the problem formulation. In contrast with the class of kernel eigenmap methods, the solution (coordinates in the low dimensional space) is characterized by a linear system instead of an eigenvalue problem. The kernel maps with a reference point are generated from a least squares support vector machine core part that is extended with an additional regularization term for preserving local mutual distances together with reference point constraints. The kernel maps possess primal and dual model representations and provide out-of-sample extensions e.g. for validation based tuning. The method is illustrated on toy problems and real life data sets.

Keywords: Dimensionality reduction, data visualization, kernel methods, least squares support vector machines, constrained optimization, feature map, positive definite kernel, validation.

1 Introduction

Traditionally, techniques such as principal component analysis and self organizing maps have been frequently applied for dimensionality reduction and data visualization [13, 15]. In recent years the search for new approaches and solutions to this problem has become an active area of research and a variety of new techniques have been proposed such as isomap, locally linear embedding (LLE), Hessian locally linear embedding, diffusion maps, Laplacian eigenmaps and others [2, 6, 9, 14, 27, 31]. Currently, many of these techniques have been characterized under the umbrella of kernel eigenmap methods and manifold learning [3, 7]. Nevertheless, the selection of components and tuning parameters is often difficult to assess. For several existing approaches the characterization of out-of-sample extensions is unclear, though approximate versions have been proposed [4]. Furthermore, the issue of model selection of tuning parameters is to a large extent still an open problem [3].

In this paper we want to take a fresh look at the problem of data visualization with dimensionality reduction. A main objective here is to design a technique for which the solution follows from solving a linear system instead of an eigenvalue problem. At the same time the method should be able to perform nonlinear dimensionality reduction with preserving local mutual distances and allow for out-of-sample extensions that can be made in an exact way. The tuning parameters should be set in view of obtaining a good generalization ability of the underlying model on new data.

Kernel based learning methods have proven to be successful in many applications in different areas, especially also on problems with high-dimensional input spaces. Different methodologies and mathematical frameworks have emerged that are making use of kernels, including support vector machine (SVM) methodologies, function estimation in reproducing kernel Hilbert spaces (RKHS) and Bayesian learning viewpoints with Gaussian processes [10, 20, 21, 26, 28, 30]. In this paper we propose a new kernel based method in the framework of least squares support vector machines (LS-SVMs). In this context kernel based models have been studied for regression, classification, principal component analysis, spectral clustering, canonical correlation analysis, recurrent networks, optimal control and others [1, 18, 23, 24, 26, 25]. The formulations are in terms of constrained optimization problems with use of a feature map in the primal problem and a related positive

definite kernel in the dual (which is the problem in the Lagrange multipliers that are related to the constraints). This is similar to standard support vector machine formulations [28] but by making use of an L_2 loss function and equality constraints instead of inequality constraints. Some advantages of this setting are the systematic and straightforward way of deriving optimal model representations and constraints handling.

The kernel map approach that we propose in this paper makes use of a reference point. Such a reference point is expressed in terms of constraints that are added to the formulation. We show how this mechanism converts the eigenvalue problem into a linear system. As stated in [22], complexity theory results for linear systems are easier to obtain than for eigenvalue problems. Also the development of fast cross-validation techniques and on-line learning methods for the design of adaptive systems becomes easier [5] and both direct and iterative solvers can be used [12]. The solution to the linear system delivers the coordinates in the low dimensional space. The support values follow from a second linear system. The optimal model representation and solution are generated in a straightforward way from a least squares support vector machine core part that maps the input data to the coordinates in the low dimensional space. An additional regularization term that preserves local mutual distances is incorporated. This term is a modified version of the one that is considered in locally linear embedding [16]. While the reference point enables to achieve a linear system solution, the point has to be sacrificed and omitted in the final data visualization. Its role is related to choosing a geometric perspective on the object for the data visualization and can be fixed beforehand.

The problem statement as a constrained optimization problem admits primal and dual model representations, respectively in terms of feature maps and the associated positive definite kernel functions. An important consequence is that the models can immediately be extended for the evaluation at new data points with out-of-sample extensions. Out-of-sample extensions can also be obtained by considering regularized least squares approaches with function estimation in a reproducing kernel Hilbert space [3, 29] when the RKHS is defined everywhere in the input space. However, in this paper we optimize not only over an unknown function, but jointly over the unknown coordinates in the low dimensional space, the primal weights of the unknown model and the error variables. The optimization setting with primal and dual problems enables to obtain the optimal model representations

and the final solution at once from the conditions for optimality.

Despite the importance of studies in learning theory with bounds on the generalization error, mainly for classification and regression problems [8, 28], practitioners often still rely on validation based methods, due to the lack of sharp bounds in general. As stated in the conclusions of [3] the issue of model selection in problems of manifold regularization is currently not well understood. In this paper we investigate two model selection criteria with use of a validation set (and randomized validation set) and cross-validation. Both criteria are normalized in the sense that they are invariant with respect to scaling of the underlying model by a constant. A normalized training objective was also proposed for problems of supervised graph inference in [29]. We show results for a number of examples with toy problems and real-life data sets with 2D and 3D visualizations of the data sets.

This paper is organized as follows. In Section 2 we discuss the context and outline of this paper. In Section 3 we consider kernel maps that lead to an eigenvalue problem, generated from a least squares support vector machine core model. In Section 4 we discuss the introduction of a reference point leading to a linear system solution. First this is explained for projection to a two-dimensional space and next for the general case. In Section 5 aspects of model selection are discussed. In Section 6 examples are given on toy problems and real life data sets. Comparisons with other methods are made. In order to enhance the readability of the paper, all proofs are given in Appendix. Supplementary material is provided with links to a Matlab demo file (Swiss roll problem) and with respect to the experimental section.

2 Context and outline

Consider a given input data set $\mathcal{D} = \{x_i\}_{i=1}^N$ of N data points $x_i \in \mathbb{R}^p$. We aim at realizing a dimensionality reduction $x \mapsto z$ to a low dimensional space by $g(\cdot) : \mathbb{R}^p \to \mathbb{R}^d$ with $d = 2$ or $d = 3$ for data visualization of the corresponding points $\{z_i\}_{i=1}^N$ with $z_i \in \mathbb{R}^d$. Let us define the vector $z = [z_1; z_2; ...; z_N] = [z_1^T z_2^T ... z_N^T]^T \in \mathbb{R}^{dN}$ and the following mechanism to

select the i-th data point from the vector z for the different components:

$$\begin{aligned} z_{i,1} &= c_{i,1}^T z \\ z_{i,2} &= c_{i,2}^T z \\ &\vdots \\ z_{i,d} &= c_{i,d}^T z \end{aligned} \tag{1}$$

where $z_{i,l}$ denotes the l-th component ($l = 1, .., d$) of the i-th data point vector z_i ($i = 1, ..., N$) with

$$\begin{aligned} c_{1,1} = [1; 0_{d-1}; 0_d; ...; 0_d], \quad ... \quad , c_{1,d} = [0_{d-1}; 1; 0_d; ...; 0_d] \\ ... \\ c_{N,1} = [0_d; 0_d; ...; 1; 0_{d-1}], \quad ... \quad , c_{N,d} = [0_d; 0_d; ...; 0_{d-1}; 1]. \end{aligned} \tag{2}$$

A meaningful objective for dimensionality reduction is to find the points z_i as the solution to

$$\min_{z_i \in \mathbb{R}^d} \; J = -\frac{\gamma}{2} \sum_{i=1}^{N} \|z_i\|_2^2 + \frac{1}{2} \sum_{i=1}^{N} \|z_i - \sum_{j=1}^{N} s_{ij} z_j\|_2^2 \tag{3}$$

where the first term is a regularization term to avoid the trivial solution with $\gamma > 0$. The second term minimizes the objective $\sum_{i=1}^{N} \|z_i - \tilde{z}_i\|_2^2$ with

$$\tilde{z}_i = \sum_{j=1}^{N} s_{ij} z_j \tag{4}$$

based on similarities s_{ij} defined on pairs of input data x_i, x_j. A typical choice of the matrix S are the ij-th entries $s_{ij} = \exp(-\|x_i - x_j\|_2^2 / \sigma^2)$. Written in terms of the unknown vector z the objective (3) becomes

$$\min_{z \in \mathbb{R}^{dN}} \; J = -\frac{\gamma}{2} z^T z + \frac{1}{2} (z - Pz)^T (z - Pz) \tag{5}$$

where

$$P = \begin{bmatrix} s_{11} I_d & s_{12} I_d & ... & s_{1N} I_d \\ s_{21} I_d & s_{22} I_d & ... & s_{2N} I_d \\ \vdots & & & \vdots \\ s_{N1} I_d & s_{N2} I_d & ... & s_{NN} Id \end{bmatrix}.$$

Setting $\partial J / \partial z = 0$ yields the eigenvalue problem

$$Rz = \gamma z \tag{6}$$

with $R = (I - P)^T(I - P)$ from which one has to select one of the candidate eigenvector solutions.

This problem (3)(5) has the following connections to the methods of locally linear embedding and Laplacian eigenmaps:

- *Locally linear embedding (LLE)* [16]

  The objective $\sum_{i=1}^N \|z_i - \sum_{j=1}^N s_{ij} z_j\|_2^2$ is related to locally linear embedding as follows. In LLE one establishes a connection between the cost function

  $$\mathcal{E}_{\text{in}} = \sum_{i=1}^N \|x_i - \sum_{j=1}^N w_{ij} x_j\|_2^2 \tag{7}$$

  and the cost function

  $$\mathcal{E}_{\text{out}} = \sum_{i=1}^N \|z_i - \sum_{j=1}^N w_{ij} z_j\|_2^2 \tag{8}$$

  through minimization of the weight matrix W with ij-th entries w_{ij} which are in common between both cost functions. One takes $w_{ij} = 0$ if x_j is not among the k-nearest neighbors of x_i and $\sum_j w_{ij} = 1$ for all i. Furthermore the output coordinates are centered and have a unit covariance matrix to prevent degenerate solutions [17]. A difference between LLE and (3)(5) in this paper is that we consider the weights s_{ij} to be functions of the differences $x_i - x_j$ such that $s(\cdot)$ can be evaluated on any pair of data points. In LLE these weights are taken to be fixed and determined at the training level.

- *Laplacian eigenmaps* [2]

  For the method of Laplacian eigenmaps, a connection in an approximate sense with an iterated Laplacian has been derived in [2]. It was noticed that in relation to $\mathcal{E}_{\text{in}}$ of LLE one has

  $$\mathcal{A} f \approx \frac{1}{2} (\Delta_{\mathcal{M}})^2 f \tag{9}$$

  where $\Delta_{\mathcal{M}}$ denotes the Laplace Beltrami operator on a differentiable function f on a manifold $\mathcal{M}$ and $\mathcal{A}$ is an operator related to the matrix $A = (I - W)^T(I - W)$.

In this paper we are interested now in the use of kernel maps in order to project the original input data to the estimated coordinates in the lower dimensional space. This will be approached in two steps:

- Firstly, we will consider the use of an underlying model that gets the given data x_i as input and outputs the estimated coordinates $\hat{z}_i$ for data visualization (instead of z_i). The core part is constituted by a least squares support vector machine regression which takes the criterion (3) as an additional regularization term in a regularized least squares objective. The solution is characterized by an eigenvalue problem and is discussed in Section 3.

- Secondly, the coordinates z_i for one single data point (e.g. for the first data point z_1) are fixed beforehand. This is incorporated by additional constraints in the problem formulation. As a result the solution is characterized by a linear system instead of an eigenvalue problem, as will be explained in Section 4.

The obtained underlying models with primal and dual model representations allow to make out-of-sample extensions. In this way the coordinates for data visualization $\hat{z}_i$ can be obtained for training, validation and test data. Model selection for the tuning parameters will be studied then in Section 5 for making use of validation sets and cross-validation.

3 Kernel maps and eigenvalue problem

We can realize the nonlinear mapping g through a least squares support vector machine regression. This is combined then with the criterion (5) which is taken as an additional regularization term. This results in the following primal problem

$$\begin{aligned} \min_{z,w_j,e_{i,j}} \quad & J_2 = -\frac{\gamma}{2} z^T z + \frac{1}{2}(z - Pz)^T (z - Pz) + \frac{\nu}{2} \sum_{j=1}^{d} w_j^T w_j + \frac{\eta}{2} \sum_{i=1}^{N} \sum_{j=1}^{d} e_{i,j}^2 \\ \text{such that} \quad & c_{i,j}^T z = w_j^T \varphi_j(x_i) + e_{i,j}, \quad \forall i = 1, ..., N; j = 1, ..., d \end{aligned} \tag{10}$$

with $w_j \in \mathbb{R}^{n_{h_j}}$ and error variables $e_{i,j}$. The regularization constants ν, η are assumed to be positive. Different feature maps $\varphi_j(\cdot) : \mathbb{R}^p \to \mathbb{R}^{n_{h_j}}$ are used in order to realize mappings from the given input space to the different components in the lower dimensional space, where n_{h_j} denote the dimensions of the feature maps. A positive definite kernel function $K(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is related to a feature map $\varphi(\cdot) : \mathbb{R}^p \to \mathbb{R}^{n_h}$ through $K(x, z) = \varphi(x)^T \varphi(z)$ (which is often called the *kernel trick*). For the commonly used radial

basis function kernel $K(x,z) = \exp(-\|x-z\|_2^2/\sigma^2)$ the feature map is infinite dimensional ($n_h \to \infty$).

The constrained optimization problem (10) is the primal problem with as unknowns the coordinates z of the data in the lower dimensional space, the error variables $e_{i,j}$ and the vector w_j for the primal representation of the model

$$\hat{z}_{*,j} = w_j^T \varphi_j(x^*) \tag{11}$$

evaluated at any point $x^* \in \mathbb{R}^p$ in the input space (and hence also allows for out-of-sample extensions). It yields the corresponding predicted coordinates $\hat{z}_{*,j}$. Neural networks interpretations with a hidden layer and hidden units can be given both to this primal representation of the model (in terms of the feature map) and to the dual representation of the model (in terms of the kernel function) [26].

The objective function J from (5) is acting in (10) as an additional regularization term to the least squares support vector machine core part. In a different context, an additional Laplacian based regularization term has also been considered in [3] for Laplacian regularized least squares methods with function estimation in a reproducing kernel Hilbert space. Also in problems of supervised graph inference [29] this has been proposed. Combined objectives with use of an LS-SVM core part have also been studied in a different area of finding approximate solutions to optimal control problems in [25].

In order to show the difference between the solutions to (5) and (10) we consider the zero bias term case. To fix the ideas let us consider the case $d = 2$ where the given input data are to be projected and visualized in a two-dimensional space:

(**Problem P1**)

$$\begin{array}{rl} \min\limits_{z,w_1,w_2,e_{i,1},e_{i,2}} & J_3 = -\dfrac{\gamma}{2} z^T z + \dfrac{1}{2}(z-Pz)^T(z-Pz) + \dfrac{\nu}{2}\left(w_1^T w_1 + w_2^T w_2\right) + \dfrac{\eta}{2}\displaystyle\sum_{i=1}^{N}(e_{i,1}^2 + e_{i,2}^2) \\ \text{such that} & c_{i,1}^T z = w_1^T \varphi_1(x_i) + e_{i,1}, \quad \forall i = 1,...,N \\ & c_{i,2}^T z = w_2^T \varphi_2(x_i) + e_{i,2}, \quad \forall i = 1,...,N. \end{array} \tag{12}$$

The solution to this problem can be viewed as a regularized version of (6) in the following sense.

Lemma 1. The solution to the problem (12) is given by the eigenvalue problem

$$\left(R + V_1(\frac{1}{\nu}\Omega_1 + \frac{1}{\eta}I)^{-1}V_1^T + V_2(\frac{1}{\nu}\Omega_2 + \frac{1}{\eta}I)^{-1}V_2^T \right) z = \gamma z \tag{13}$$

with $R = (I-P)^T(I-P)$ and kernel matrices $\Omega_1, \Omega_2 \in \mathbb{R}^{N\times N}$ with ij-th entries $\Omega_{1,ij} = K_1(x_i, x_j) = \varphi_1(x_i)^T\varphi_1(x_j)$, $\Omega_{2,ij} = K_2(x_i, x_j) = \varphi_2(x_i)^T\varphi_2(x_j)$ for $i, j = 1, ..., N$ and positive definite kernel functions $K_1(\cdot,\cdot), K_2(\cdot,\cdot)$. The matrices V_1, V_2 equal

$$\begin{array}{rcl} V_1 & = & [c_{1,1}\, c_{2,1} \ldots c_{N,1}] \\ V_2 & = & [c_{1,2}\, c_{2,2} \ldots c_{N,2}] \end{array} \tag{14}$$

with elements $c_{i,l}$ as defined in (2).

Proof: see Appendix

□

Corollary 1. The dual representations of the model, evaluated at a point $x^* \in \mathbb{R}^p$, are expressed in terms of the Lagrange multipliers $\alpha_{i,1}, \alpha_{i,2}$ (corresponding to the constraints in (12))

$$\begin{array}{rcl} \hat{z}_{*,1} & = & \frac{1}{\nu}\sum_{i=1}^{N} \alpha_{i,1} K_1(x_i, x^*) \\ \hat{z}_{*,2} & = & \frac{1}{\nu}\sum_{i=1}^{N} \alpha_{i,2} K_2(x_i, x^*) \end{array} \tag{15}$$

where $\alpha_{i,1}, \alpha_{i,2}$ are the unique solution to the linear systems

$$(\frac{1}{\nu}\Omega_1 + \frac{1}{\eta}I)\alpha_1 = V_1^T z \;\text{ and }\; (\frac{1}{\nu}\Omega_2 + \frac{1}{\eta}I)\alpha_2 = V_2^T z \tag{16}$$

with z a solution to (13) and $\alpha_1 = [\alpha_{1,1}; ...; \alpha_{N,1}]$, $\alpha_2 = [\alpha_{1,2}; ...; \alpha_{N,2}]$.

The eigenvalue problem (13) has $2N$ eigenvector solutions corresponding to the eigenvalues γ. However, it turns out that the selection of the best solution from this pool of $2N$ candidates is not straightforward. Selection of a best solution interacts in fact with the model selection problem for ν, η and the kernel tuning parameters. The best solution (in terms of good generalization) is often not corresponding with the largest or smallest eigenvalue γ in (13). A numerical example to further illustrate this problem will be given in Section 6.

4 Main result: linear system solution obtained from reference point constraints

The constrained optimization problem formulations in least squares support vector machines enable to incorporate additional constraints in a straightforward manner. We will demonstrate now how adding reference point constraints converts the eigenvalue problem (13) into a new problem where the solution is given by a linear system. We first consider projection of the input data to a two-dimensional space. Next we consider the general case.

4.1 Projection to a two-dimensional space

The main idea is to define one of the data points of the training set as a *reference point* for which the coordinates in the low dimensional space are chosen (approximately) and fixed beforehand to a non-zero vector. We assume here (without loss of generality) that this is the first data point of the given training set. The new problem formulation becomes

(**Problem P2**)

$$\begin{aligned}
\min_{z,w_1,w_2,b_1,b_2,e_{i,1},e_{i,2}} \quad & J_4 = -\frac{\gamma}{2} z^T z + \frac{1}{2}(z - P_D z)^T (z - P_D z) + \frac{\nu}{2}\left(w_1^T w_1 + w_2^T w_2\right) + \frac{\eta}{2}\sum_{i=1}^{N}(e_{i,1}^2 + e_{i,2}^2) \\
\text{such that} \quad & c_{1,1}^T z = q_1 + e_{1,1} \\
& c_{1,2}^T z = q_2 + e_{1,2} \\
& c_{i,1}^T z = w_1^T \varphi_1(x_i) + b_1 + e_{i,1}, \quad \forall i = 2, ..., N \\
& c_{i,2}^T z = w_2^T \varphi_2(x_i) + b_2 + e_{i,2}, \quad \forall i = 2, ..., N.
\end{aligned} \tag{17}$$

In this constrained optimization problem the core part is the LS-SVM mapping of the input data to the coordinates in the two-dimensional space. The first two terms of the objective function take into account neighborhood preservation. More flexibility is incorporated by modifying (5) into $\sum_{i=1}^{N} \|z_i - \sum_{j=1}^{N} s_{ij} D z_j\|_2^2 = (z - P_D z)^T (z - P_D z)$ with diagonal matrix $D \in \mathbb{R}^{d \times d}$ and

$$P_D = \begin{bmatrix} s_{11}D & s_{12}D & ... & s_{1N}D \\ s_{21}D & s_{22}D & ... & s_{2N}D \\ \vdots & & & \vdots \\ s_{N1}D & s_{N2}D & ... & s_{NN}D \end{bmatrix}.$$

For the first training data point x_1 *fictitious* coordinates of the non-zero reference point $q = [q_1; q_2] \in \mathbb{R}_0^2$ are specified by the user. Technically speaking, this point q makes the problem in fact semi-supervised instead of unsupervised (in an artificial way). On the other hand, this reference point needs to be sacrificed in the final visualization of the coordinates z_i. It serves as an *eye* for taking a geometrical perspective on an object. Furthermore, additional bias terms b_1, b_2 are taken in the model. The solution is characterized now as follows.

Lemma 2. Assuming $\gamma \leq 0$, $\nu, \eta > 0$ and $q = [q_1; q_2] \in \mathbb{R}_0^2$, the unique solution to the problem (17) is given by the linear system

$$\left[\begin{array}{c|c|c} U & -V_1 M_1^{-1} 1 & -V_2 M_2^{-1} 1 \\ \hline -1^T M_1^{-1} V_1^T & 1^T M_1^{-1} 1 & 0 \\ \hline -1^T M_2^{-1} V_2^T & 0 & 1^T M_2^{-1} 1 \end{array}\right] \left[\begin{array}{c} z \\ \hline b_1 \\ \hline b_2 \end{array}\right] = \left[\begin{array}{c} \eta(q_1 c_{1,1} + q_2 c_{1,2}) \\ \hline 0 \\ \hline 0 \end{array}\right] \tag{18}$$

with matrices

$$\begin{aligned} U &= (I - P_D)^T (I - P_D) - \gamma I + V_1 M_1^{-1} V_1^T + V_2 M_2^{-1} V_2^T + \eta c_{1,1} c_{1,1}^T + \eta c_{1,2} c_{1,2}^T \\ M_1 &= \frac{1}{\nu}\Omega_1 + \frac{1}{\eta} I \,,\; M_2 = \frac{1}{\nu}\Omega_2 + \frac{1}{\eta} I \\ V_1 &= [c_{2,1} \ldots c_{N,1}] \,,\; V_2 = [c_{2,2} \ldots c_{N,2}] \end{aligned}$$

and kernel matrices $\Omega_1, \Omega_2 \in \mathbb{R}^{(N-1)\times(N-1)}$ with ij-th entries $\Omega_{1,ij} = K_1(x_i, x_j) = \varphi_1(x_i)^T \varphi_1(x_j)$, $\Omega_{2,ij} = K_2(x_i, x_j) = \varphi_2(x_i)^T \varphi_2(x_j)$ for $i, j = 2, ..., N$ and positive definite kernel functions $K_1(\cdot,\cdot), K_2(\cdot,\cdot)$.

Proof: see Appendix

□

Corollary 2. The primal and dual model representations allow making out-of-sample extensions. Evaluated at point $x^* \in \mathbb{R}^p$ the predicted coordinates are

$$\begin{aligned} \hat{z}_{*,1} &= w_1^T \varphi_1(x^*) + b_1 = \frac{1}{\nu} \sum_{i=2}^{N} \alpha_{i,1} K_1(x_i, x^*) + b_1 \\ \hat{z}_{*,2} &= w_2^T \varphi_2(x^*) + b_2 = \frac{1}{\nu} \sum_{i=2}^{N} \alpha_{i,2} K_2(x_i, x^*) + b_2 \end{aligned} \tag{19}$$

where b_1, b_2 are the solution to (18) and $\alpha_1, \alpha_2 \in \mathbb{R}^{N-1}$ are the unique solutions to the linear systems

$$M_1 \alpha_1 = V_1^T z - b_1 1_{N-1} \;\text{ and }\; M_2 \alpha_2 = V_2^T z - b_2 1_{N-1} \tag{20}$$

with z the solution to (18) and $\alpha_1 = [\alpha_{2,1}; ...; \alpha_{N,1}]$, $\alpha_2 = [\alpha_{2,2}; ...; \alpha_{N,2}]$, $1_{N-1} = [1; 1; ..., ; 1]$.

The assumption $\gamma \leq 0$ in Lemma 2 is made for establishing convexity of the problem. Observe that in (3) $\gamma > 0$ has been assumed. However, this assumption was needed for a different motivation of avoiding the trivial solution and pushing the coordinates away from the origin. In problem P2 the first term $-\gamma z^T z$ has been included for the sake of generality. In practice $\gamma = 0$ can be set, which at the same time will simplify the tuning parameter search process.

4.2 General case

In the general case with dimensionality reduction to a d-dimensional space the problem statement is

(**Problem P3**)

$$\begin{aligned} \min_{z,w_j,b_j,e_{i,j}} \quad & J_5 = -\frac{\gamma}{2} z^T z + \frac{1}{2}(z - P_D z)^T (z - P_D z) + \frac{\nu}{2} \sum_{j=1}^{d} w_j^T w_j + \frac{\eta}{2} \sum_{i=1}^{N} \sum_{j=1}^{d} e_{i,j}^2 \\ \text{such that} \quad & c_{1,j}^T z = q_j + e_{1,j}, \quad \forall j = 1, ..., d \\ & c_{i,j}^T z = w_j^T \varphi_j(x_i) + b_j + e_{i,j}, \quad \forall i = 2, ..., N \,;\, j = 1, ..., d. \end{aligned} \tag{21}$$

The solution is characterized in a similar way as in the two-dimensional case.

Lemma 3. Assuming $\gamma \leq 0$, $\nu, \eta > 0$ and $q \in \mathbb{R}_0^d$, the unique solution to the problem (21) is given by the linear system

$$\left[\begin{array}{c|c|c|c} U & -V_1 M_1^{-1} 1 & ... & -V_d M_d^{-1} 1 \\ \hline -1^T M_1^{-1} V_1^T & 1^T M_1^{-1} 1 & 0 & 0 \\ \hline \vdots & 0 & \ddots & 0 \\ \hline -1^T M_d^{-1} V_d^T & 0 & 0 & 1^T M_d^{-1} 1 \end{array}\right] \left[\begin{array}{c} z \\ \hline b_1 \\ \hline \vdots \\ \hline b_d \end{array}\right] = \left[\begin{array}{c} \eta \sum_{j=1}^{d} q_j c_{1,j} \\ \hline 0 \\ \hline \vdots \\ \hline 0 \end{array}\right] \tag{22}$$

with

$$U = (I - P_D)^T (I - P_D) - \gamma I + \sum_{j=1}^{d} V_j M_j^{-1} V_j^T + \eta \sum_{j=1}^{d} c_{1,j} c_{1,j}^T$$

$$M_j = \frac{1}{\nu} \Omega_j + \frac{1}{\eta} I \,,\, V_j = [c_{2,j} \, ... \, c_{N,j}] \,,\, j = 1, ..., d$$

and kernel matrices $\Omega_j \in \mathbb{R}^{(N-1)\times(N-1)}$ with ij-th entries $\Omega_{j,kl} = K_j(x_k, x_l) = \varphi_j(x_k)^T \varphi_j(x_l)$, for $j = 1, ..., d$ and $k, l = 2, ..., N$ and positive definite kernel functions $K_j(\cdot, \cdot)$.

Proof: see Appendix

□

Corollary 3. The primal and dual model representations allow to make out-of-sample extensions. Evaluated at point $x^* \in \mathbb{R}^p$ the predicted coordinates are

$$\hat{z}_{*,j} = w_j^T \varphi_j(x^*) + b_j = \frac{1}{\nu} \sum_{i=2}^{N} \alpha_{i,j} K_j(x_i, x^*) + b_j \tag{23}$$

where the bias terms b_j are the solution to (22) and $\alpha_j \in \mathbb{R}^{N-1}$ are the unique solutions to the linear systems

$$M_j \alpha_j = V_j^T z - b_j 1_{N-1} \tag{24}$$

with z the solution to (22) and $\alpha_j = [\alpha_{2,j}; ...; \alpha_{N,j}]$ for $j = 1, ..., d$.

For moderate size data sets the linear systems can be solved using direct methods [12]. For larger scale problems iterative solvers such as conjugate gradient methods can be used [12]. However, for conjugate gradient methods the matrix of the problem needs to be positive definite. The problems (18) and (22) can be converted then into problems with a positive definite matrix [26], similar as for LS-SVM models in classification and regression with bias terms. The computational complexity is well understood in terms of the condition number of the matrix and the required precision [22].

5 Model selection by validation

5.1 Criteria

In order to achieve a good generalization ability of the kernel map approach, a careful selection of the tuning parameters is needed. We investigate here two criteria with use of a validation set or cross-validation. The kernel map with reference point approach can be tuned then in a similar fashion as kernel based regression and classification methods with linear system solving, though for a problem of unsupervised instead of supervised learning now. While in problems of regression and classification one aims at designing an accurate predictive model, for data visualization applications finding a *good range* of the tuning

parameters will usually be satisfactory. One may typically search over a rougher grid for data visualization than in regression and classification applications.

Consider an input training data set $\mathcal{D}_T = \{x_i\}_{i=1}^N$, validation set $\mathcal{D}_V = \{x_j^{\text{val}}\}_{j=1}^{N_V}$ and test set $\mathcal{D}_E = \{x_l^{\text{test}}\}_{l=1}^{N_E}$ where N, N_V, N_E denote the number of training, validation and test data points, respectively. The following two criteria are considered here with use of a validation set $\mathcal{D}_V$:

- *Normalized metric multidimensional scaling criterion* (normalized metric MDS)

$$\min_{\Theta} \mathcal{Q}_{\text{nmds}} = \sum_{i,j=1}^{N_V} \left(\frac{\hat{z}_i^{\text{val}T} \hat{z}_j^{\text{val}}}{\|\hat{z}_i^{\text{val}}\|_2 \|\hat{z}_j^{\text{val}}\|_2} - \frac{x_i^{\text{val}T} x_j^{\text{val}}}{\|x_i^{\text{val}}\|_2 \|x_j^{\text{val}}\|_2} \right)^2 \tag{25}$$

- *Normalized locally linear embedding criterion* (normalized LLE)

$$\min_{\Theta} \mathcal{Q}_{\text{nlle}} = \frac{\sum_{i=1}^{N_V} \|\hat{z}_i^{\text{val}} - \sum_{j=1}^{N_V} s_{ij}^{\text{val}} D \hat{z}_j^{\text{val}}\|_2^2}{\sum_{j=1}^{N_V} \|\hat{z}_j^{\text{val}}\|_2^2} \tag{26}$$

with $s_{ij}^{\text{val}} = \exp(-\|x_i^{\text{val}} - x_j^{\text{val}}\|_2^2/\sigma^2)$ denoting the ij-th entry of matrix S^{val},

where Θ denotes the set of tuning parameters to be determined. Both criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ are normalized in the sense that they are invariant with respect to scaling of the estimated coordinates $\hat{z}_j^{\text{val}}$ by a constant. Also in [29] such a type of invariance has been considered at the training level in supervised graph inference. Taking only the numerator part of $\mathcal{Q}_{\text{nlle}}$ turns out to be non suitable for model selection purposes.

5.2 Choice of the tuning parameters for kernel maps with a reference point

While the coordinates z, the support values α and bias terms b follow from solving linear systems, the choice of a number of tuning parameters needs to be decided. In setting these tuning parameters, on the one hand there should be enough flexibility to obtain a model that yields good generalization on data sets in different problems. On the other hand having too many tuning parameters severely complicates the model selection process. For this reason the following tuning parameters will be fixed beforehand throughout this paper:

- *Choice of the reference point q:*

  The choice of the point q plays the role of choosing a geometric perspective on an object. For 2D and 3D projections we take as candidate reference points $q \in \mathcal{S}_{\text{ref}}^{\text{2D}}$ and $q \in \mathcal{S}_{\text{ref}}^{\text{3D}}$ respectively, where

$$\begin{aligned} \mathcal{S}_{\text{ref}}^{\text{2D}} &= \{[+1;+1],[+1;-1],[-1;+1],[-1,-1]\} \\ \mathcal{S}_{\text{ref}}^{\text{3D}} &= \{[+1;+1;+1],[+1;-1;+1],[-1;+1;+1],[-1;-1;+1], \\ &\quad [+1;+1;-1],[+1;-1;-1],[-1;+1;-1],[-1,-1;-1]\}. \end{aligned} \tag{27}$$

  The role of this reference point is further illustrated by examples in Section 6. A multiplication of the reference vector q with a positive constant $c \in \mathbb{R}^+$ leads to similar results as taking this constant $c = 1$, corresponding to the choices in (27).

- *Choice of the diagonal matrix D:*

  Experiments indicate that better results are obtained by taking the diagonal matrix D different from the identity matrix. Throughout this paper, for 2D projection we fix $D = \text{diag}\{10, 1\}$ and for 3D projections $D = \text{diag}\{10, 5, 1\}$.

- *Choice of γ:*

  As explained in Section 4, we set $\gamma = 0$.

In this way the unknown set of tuning parameters Θ is reduced to:

- *Kernels tuning parameters in s_{ij}, $K_1, K_2, (K_3)$:*

  When using radial basis functions in

$$s_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma^2),\ K_j(x,z) = \exp(-\|x - z\|_2^2/\sigma_j^2) \tag{28}$$

  for $j = 1, .., d$ the kernel parameters σ, σ_j need to be determined. Especially the determination of a good range for these values turns out to be relevant as well as the values relative with respect to each other.

- *Regularization constants ν, η:*

  The determination of a good range is needed for the value ν. The value $\eta = 1$ is set.

It is also recommended to (linearly) scale the input data properly with elements of the input data in the range $[-1, +1]$ before applying the method or to normalize the data.

5.3 Algorithms: randomized validation set and cross-validation

Data visualization techniques are frequently applied for exploratory data analysis. This process is more interactive than the design of models for regression or classification purposes. For existing methods one often varies then the tuning parameters of the method which results in different data visualizations of a given data set. Currently, due to the lack of general theories for setting the tuning parameters, this model selection aspect is mostly ignored, often resulting into a quite subjective data visualization process.

One aim of this paper is to make this data visualization process less subjective, by estimating an underlying model that one can extend and evaluate on validation data. We describe now two practical algorithms: a cross-validation method with grid search and a method with use of a randomized validation set which is more suitable for interactive data exploration. The following cross-validation algorithm is proposed.

Algorithm 1: cross-validation with grid search for 3D visualization

---

1. **Input:** load data set $\mathcal{D}$, linearly scale elements to interval [-1,+1].

2. Partition $\mathcal{D}$ into test part $\mathcal{D}_E$ and part $\mathcal{D}_{TV}$ for training-validation with cross-validation (e.g. $n_{CV} = 10$ for 10-fold cross-validation).

3. Assign the first data point of $\mathcal{D}_{TV}$ to be the reference point (kept the same over all folds in the cross-validation process) and choose $q \in \mathcal{S}_{\text{ref}}^{\text{3D}}$ for (27).

4. Define tuning parameter grid:
   $v_{\sigma^2} = p \times [10; 20; 30; 40; 50; 60; 70; 100; 120] \in \mathbb{R}^{n_{\sigma^2}}$,
   $v_{\sigma_1^2} = p \times [10; 20; 30; 40; 50; 60; 70; 100; 120] \in \mathbb{R}^{n_{\sigma_1^2}}$, for (28)
   $v_\nu = [0.1; 1; 10] \in \mathbb{R}^{n_\nu}$,
   $\sigma_2^2 = a_2\sigma_1^2$, $\sigma_3^2 = a_3\sigma_1^2$, where $a_2, a_3 \in \{0.5, 1, 2\}$.

5. **For CV-loop** index = 1 to n_{CV}

6. Split $\mathcal{D}_{TV}$ into training part $\mathcal{D}_T$ and validation part $\mathcal{D}_V$ according to the cross-validation procedure.

7. **For tuning-loops** over indices = 1 to $\{n_{\sigma^2}, n_{\sigma_1^2}, n_\nu, 3, 3\}$

8. Construct kernel matrices S, S^{val}, $\Omega_1, \Omega_2, \Omega_3$.
   Construct matrices P_D, V_1, V_2, V_3.

9. Solve linear system (22) which gives z, b_j.
   Solve linear system (24) which gives α_j of the model.

10. Evaluate the model on validation data based on (23) to obtain $\hat{z}_j^{\text{val}}$.

11. Compute the criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ on validation data (25)(26).

12. **End tuning-loops**

13. **End CV-loop**

14. **Output:** Sort results according to $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$. Visualize data for $\mathcal{D}_E$ and $\mathcal{D}_{TV}$.

Note that in $v_{\sigma^2}, v_{\sigma_1^2}$ a scaling with p is taken. For higher dimensional input spaces the σ^2, σ_1^2 values need to scale proportionally. It could also be needed to enlarge or further refine the grid depending on the outcome of the grid search. In principle one could also define vectors $v_{\sigma_2^2}$, $v_{\sigma_3^2}$ similar to $v_{\sigma_1^2}$. However, this would lead to a very time consuming search process over all different tuning parameters. Experiments also indicate that for a good generalization the ranges of σ, σ_1, σ_2, σ_3 should not be too far away from each other. This motivates the search over neighboring ranges for σ_2, σ_3 relative with respect to σ_1.

Cross-validation with grid search is especially popular for kernel based methods in regression and classification problems. However, for exploratory data analysis a more interactive search process might be more appealing to the user. The use of a single validation set is one possibility but the tuning might become too specific with respect to this validation set then. Therefore, tuning parameter search with a randomized validation set is proposed to make the result less depending on a specific set and keep the overall search process sufficiently interactive. No grid is defined in this case. The user interactively explores the tuning parameter search space then trying to discover structure in the data. This process can be guided now by the criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ evaluated on validation data. This gives an indication to the user about whether the observed structure can be trusted

in an objective way (in the sense of good generalization) or is rather a "fake" structure corresponding to a situation of data overfitting. Experiments also indicate that the criterion $\mathcal{Q}_{\text{nlle}}$ has a tendency to overfit for small values of σ, σ_1, σ_2, σ_3. The following algorithm is proposed for use of a randomized validation set.

Algorithm 2: randomized validation set for 3D visualization

---

1. **Input:** load data set $\mathcal{D}$, linearly scale elements to interval [-1,+1].
2. Partition $\mathcal{D}$ into test part $\mathcal{D}_E$ and part $\mathcal{D}_{TV}$ for training-validation.
3. Assign the first data point of $\mathcal{D}_{TV}$ to be the reference point (kept the same over all folds in the cross-validation process) and choose $q \in \mathcal{S}_{\text{ref}}^{\text{3D}}$ in (27).
4. Decide on the numbers N, N_V for splitting $\mathcal{D}_{TV}$ into training and validation parts.
   Decide on the number of randomizations n_{rand} of the validation set.
5. Make a choice of the tuning parameters $\sigma, \sigma_1, \sigma_2, \sigma_3, \nu$.
6. **For randomization-loop** index = 1 to n_{rand}
7. Randomly split $\mathcal{D}_{TV}$ into training part $\mathcal{D}_T$ and validation part $\mathcal{D}_V$
8. Construct kernel matrices S, S^{val}, $\Omega_1, \Omega_2, \Omega_3$.
   Construct matrices P_D, V_1, V_2, V_3.
9. Solve linear system (22) which gives z, b_j.
   Solve linear system (24) which gives α_j of the model.
10. Evaluate the model on validation data based on (23) to obtain $\hat{z}_j^{\text{val}}$.
11. Compute the criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ on validation data (25)(26).
12. **End randomization-loop**
13. Compute the means of $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ over the n_{rand} randomized validation sets.
14. **Interactive output:** Visualize data for $\mathcal{D}_E$ and $\mathcal{D}_{TV}$ for the chosen tuning parameters.

15. **If** validation criteria indicate a good solution, **End search**
    Else go to 5 and continue the search process.

Algorithms 1 and 2 have been described here for the 3D visualization case. The algorithms for 2D visualization are similar but simpler as one has the search then over fewer tuning parameters. Under supplementary material a demo is provided to illustrate the method with use of a single validation set and the criterion $\mathcal{Q}_{\text{nmds}}$.

6 Examples

6.1 Spiral problem

Role of the reference point

In this first toy problem, a spiral data problem is considered with data generated in Matlab (see supplementary material), scaled to ranges $[-1, 1]$ and made zero mean. The given input data are in a 3-dimensional space, shown in Figure 1. The problem P2 was considered for data visualization in 2D. The figure illustrates the role of the reference point with different choices of $q \in \mathcal{S}_{\text{ref}}^{\text{2D}}$ from (27). Different choices of q lead to different perspectives on the object. In this example the tuning parameter selection was done interactively with use of a single validation set resulting into the kernel parameters $\sigma^2 = 100$, $\sigma_1^2 = 80$, $\sigma_2^2 = 0.1\sigma_1^2$, $\nu = 0.5$. All other tuning parameters were taken as described in Section 5. Besides good visualization of the training and validation set parts, an excellent generalization is also obtained on test data. The arrows indicate the positions of the reference points which are located outside the range of all other points, playing the role of an eye for looking at the object.

Tuning by cross-validation

In Figure 2 and 3 the results of tuning parameter selection by 10-fold cross-validation is illustrated with application of Algorithm 1 for 2D visualization. Figure 2 shows the two best results for the considered (rough) grid according to the criteria $\mathcal{Q}_{\text{nmds}}$ and $\mathcal{Q}_{\text{nlle}}$. The

reference point $q = [+1; -1]$ is chosen. The grid search process with evaluated criteria over the 10 different runs is shown in Figure 3. The resulting tuning parameters are $\sigma^2 = 210$, $\sigma_1^2 = 210$, $\sigma_2^2 = 105$, $\nu = 1$ according to $\mathcal{Q}_{\text{nmds}}$ and $\sigma^2 = 300$, $\sigma_1^2 = 210$, $\sigma_2^2 = 105$, $\nu = 0.1$ according to $\mathcal{Q}_{\text{nlle}}$. For smaller values of σ, σ_1, σ_2 (not included in this grid search) it can be observed that the criterion $\mathcal{Q}_{\text{nlle}}$ has a tendency to overfit. The mean value over the different runs becomes small then, but with a large variance over the ten different runs (such that one can detect the problem in this way).

Comparison with other methods

A comparison with the following other methods is shown in Figure 4: MDS, PCA, LLE (Hessian LLE, which is not shown, gives a qualitatively similar result to LLE), Laplacian eigenmaps and diffusion maps using the software mentioned in the supplementary material. In this case no division is made in training, validation and test parts. From the total of 800 data points 600 points were randomly selected for the training with these methods. The tuning parameters are set as follows: (LLE and Hessian LLE) 10 nearest neighbors, (Laplacian eigenmap) 10 nearest neighbors, sigma = 10, (diffusion map) sigma = 10, alpha = 1. The results of LLE and Hessian LLE are closest to the kernel map with reference point method.

Comparison with eigenvalue problem method

Also the problem P1 with the solution characterized by the eigenvalue problem has been tested. All experiments indicated that the component selection was difficult in that case: the best solutions were often not corresponding to the largest or smallest eigenvalue in (13). As a result an exhaustive search through different eigenvector solution candidates may be needed then together with a validation procedure for the optimal component selection (or by any other suitable model selection procedure), which is computationally much harder. Moreover, the conditioning of the numerical algorithms for computing the eigenvector solutions is inferior (even if one specifies to compute only a subset of the eigenvectors such as with the Matlab functions svds and eigs) with respect to the well-conditioned linear systems to be solved in problems P2 and P3. A numerical example is given in Figure 4: the optimal component in this case did not correspond to the largest or smallest eigenvalue.

Such issues also occur in denoising applications with kernel principal component analysis [19, 24] where the ordering of the eigenvalues and the corresponding components does not always correspond with their relevance towards the reconstruction error. Also note the bad numerical conditioning for this problem in Figure 4 (bottom-right) with $\hat{z}_1$ values in the order of magnitude 10^{-15} while $\hat{z}_2$ is in the order 10^{-2}, despite the specification of suitable tolerance parameters.

6.2 Swiss roll data

A well-known example is the swiss roll problem [16]. We generated the input data in Matlab as shown in the supplementary material, scaled to ranges $[-1, 1]$ and made zero mean. The given data in 3D are shown in Figure 5. This Figure also shows the result of a kernel map with reference point with use of a single validation set, where the first 100 points are chosen as the validation set and the next 600 as the training data. The resulting tuning parameters for problem P2 with data visualization in 2D are $\sigma^2 = 200$, $\sigma_1^2 = 200$, $\sigma_2^2 = \sigma_1^2$, $\nu = 0.5$. The reference point $q = [+1; -1]$ is chosen.

In Figure 5 (bottom) tuning parameter search is done using Algorithm 2 with a randomized validation set and $n_{rand} = 3$. The training, validation and test parts are partitioned as $N = 600 \times (2/3)$, $N_V = 600 \times (1/3)$, $N_E = 200$. The best results indicated by $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ are in the ranges $\sigma^2 \in [125, 150]$, $\sigma_1^2 \in [50, 100]$, $\sigma_2^2 = \sigma_1^2$, $\nu = 1$. Figure 5 (bottom) shows an obtained result from this selected range of tuning parameters.

A comparison with the following other methods is shown in Figure 6: MDS, PCA, LLE, Laplacian eigenmaps and diffusion maps. In this case no division is made in training, validation and test parts. All 800 data points are used for the training. The results of the methods LLE and Laplacian eigenmaps give very different results depending on the choice of their tuning parameters. While these methods are often applied in practice with a relatively small number of nearest neighbors (e.g. 10), for achieving results similar to the solution of kernel maps with a reference point, a very large number of nearest neighbors is needed in this case (different from the choices illustrated e.g. in [2]): (LLE) 200 nearest neighbors, (Laplacian eigenmap) 300 nearest neighbors, sigma = 50. For LLE the significant difference in results between using 50 and 200 nearest neighbors is shown: for smaller numbers of nearest neighbors the rolling structure is not obtained. Larger

numbers of nearest neighbors also takes much higher computation times. The diffusion map method is applied with sigma = 10 and alpha = 1. The tuning parameters of these existing methods were set in such a way that they reveal a similar structure as the kernel map with reference point method.

6.3 Alon colon cancer microarray data set

In this example we aim at visualizing the gene distribution for the Alon colon cancer microarray data set (for further information see the supplementary material). In total 1500 genes are taken: the number of training, validation and test parts are partitioned as $N = 1000 \times (1/2)$, $N_V = 1000 \times (1/2)$, $N_E = 500$. For the given microarray data set the dimension of the input space is $p = 62$.

In Figure 7 the results of 2D visualization by the kernel map with reference point are shown. Tuning parameter search is done using Algorithm 2 with a randomized validation set and $n_{rand} = 3$. The resulting tuning parameters for problem P2 with data visualization in 2D are $\sigma^2 = 10000$, $\sigma_1^2 = 1000$, $\sigma_2^2 = 0.1\sigma_1^2$, $\nu = 50$. The reference point $q = [+1; -1]$ is chosen. Figure 7 also shows the result of assigning different parts to be training and test data, which does not affect the discovered structure in the data visualization.

In Figure 8 (top-left) the role of the reference point is illustrated in 3D visualization of the data with a choice of $q \in \mathcal{S}_{\text{ref}}^{\text{3D}}$ from (27). Different choices of q lead to different perspectives on the object. The tuning parameter search is done using Algorithm 2 with a randomized validation set and $n_{rand} = 3$. The resulting tuning parameters for problem P3 with data visualization in 3D are $\sigma^2 = 10000$, $\sigma_1^2 = 1000$, $\sigma_2^2 = 0.8\sigma_1^2$, $\sigma_3^2 = 0.3\sigma_1^2$, $\nu = 50$, indicated by the criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$.

A comparison with the following other methods is shown in Figure 8: MDS, PCA, LLE, Laplacian eigenmaps and diffusion maps. In this case no division is made in training, validation and test parts. From the total of 1500 data points 700 points were randomly selected as the training set for these methods. For LLE, Laplacian eigenmaps and diffusion maps the tuning parameters were selected that lead to similar structure as found by kernel maps with reference points. For Laplacian eigenmaps the result is very much outlying with respect to all other methods on this example. The tuning parameters are as follows: (LLE) 100 nearest neighbors, (Laplacian eigenmap) 10 nearest neighbors, sigma = 2000,

(diffusion map) sigma = 100, alpha = 1.

6.4 Santa Fe chaotic laser data

In this example we visualize the time-series $\{y_t\}_{t=1}^T$ of the Santa Fe chaotic laser data set (www-psych.stanford.edu/∼andreas/Time-Series/SantaFe.html) by defining the vectors $y_{t|t-m} = [y_t; y_{t-1}; y_{t-2}; ...; y_{t-m}]$ with $m = 9$. The data set consists then of the points $\{y_{t|t-m}\}_{t=m+1}^{t=m+N_{tot}}$ with $N_V = 600 \times (1/3)$ validation data (first part) and $N = 600 \times (2/3)$ training data points (middle part) and $N_E = 300$ test data (Figure 9) in a $p = 10$ dimensional input space.

In Figure 9 (bottom) tuning parameter search is done using Algorithm 2 with a randomized validation set and $n_{rand} = 3$. The best results for 2D visualization indicated by $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ are $\sigma^2 = 500$, $\sigma_1^2 = 100$, $\sigma_2^2 = \sigma_1^2$, $\nu = 10$. The reference point $q = [+1; -1]$ is chosen.

In Figure 10 (top-left) a 3D visualization is shown for a reference point choice. The application of Algorithm 2 with a randomized validation set leads to the selection of the tuning parameters as $\sigma^2 = 480$, $\sigma_1^2 = 120$, $\sigma_2^2 = \sigma_1^2$, $\sigma_3^2 = \sigma_1^2$, $\nu = 10$.

A comparison with the following other methods is shown in Figure 10: MDS, PCA, LLE, Laplacian eigenmaps and diffusion maps. In this case no division is made in training, validation and test parts. All 900 data points were used as training set for these methods. The tuning parameters are as follows: (LLE) 100 nearest neighbors, (Laplacian eigenmap) 300 nearest neighbors, sigma = 100, (diffusion map) sigma = 300, alpha = 1. These were set in such a way that they reveal a similar structure as the kernel map with reference point method.

7 Conclusions

In recent years considerable progress has been made in the area of data visualization and dimensionality reduction using kernel eigenmap methods. In this paper we have shown how to characterize solutions by linear systems instead of eigenvalue problems. For this purpose we proposed a new method of kernel maps with a reference point. The framework of least squares support vector machines is adopted which naturally allows to incorpo-

rate reference point constraints in the primal problem within the constrained optimization problem. Kernel model representations are generated in a systematic and straightforward way in the dual. The optimal coordinates for visualization in the low dimensional space are computed from a linear system in a numerically reliable way. The framework immediately allows for out-of-sample extensions. Validation based learning of the tuning parameters has been demonstrated with good results on toy problems and real-life data sets.

Acknowledgements

Research supported by Research Council K.U. Leuven: GOA AMBioRICS, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07; Research communities (ICCoS, ANMMM, MLDM); AWI: BIL/05/43, IWT: PhD Grants; Belgian Federal Science Policy Office: IUAP P5/22, IUAP DYSCO.

Supplementary material

A Matlab demo file that shows the application of the kernel map reference point method to the Swiss roll problem can be downloaded from:

http://www.esat.kuleuven.be/sista/lssvmlab/KMref/demoswissKMref.m.

For comparisons with other methods including MDS, PCA, ISOMAP, LLE, Hessian LLE, Laplacian, diffusion map, LTSA:

http://www.math.umn.edu/∼wittman/mani/

http://www.cs.toronto.edu/∼roweis/lle/code.html.

Spiral data Matlab code:

```
t=0.01:0.01:4;x=cos(7*t);y=sin(7*t);z=t
```

Swiss roll data Matlab code:

```
Ntot=700;tt =(3*pi/2)*(1+2*(1:1:Ntot)/Ntot);height=2*rand(1,Ntot);X=[tt.*cos(tt);height;tt.*sin(tt)]
```

http://www.cs.toronto.edu/∼roweis/lle/code/swissroll.m

Alon colon cancer microarray data set:

information and links at http://homes.esat.kuleuven.be/∼npochet/Bioinformatics/

Appendix

Proof of Lemma 1

The solution (13) is obtained in a straightforward way. First construct the Lagrangian

$$\mathcal{L}(z, w_1, w_2, e_{i,1}, e_{i,2}; \alpha_{i,1}, \alpha_{i,2}) = J_3 + \sum_{i=1}^{N} \alpha_{i,1}(c_{i,1}^T z - w_1^T \varphi_1(x_i) - e_{i,1}) + \sum_{i=1}^{N} \alpha_{i,2}(c_{i,2}^T z - w_2^T \varphi_2(x_i) - e_{i,2}).$$

For characterization of the local minima, take the conditions for optimality $\partial\mathcal{L}/\partial z = 0, \partial\mathcal{L}/\partial w_1 = 0, \partial\mathcal{L}/\partial w_2 = 0, \partial\mathcal{L}/\partial e_{i,1} = 0, \partial\mathcal{L}/\partial e_{i,2} = 0, \partial\mathcal{L}/\partial\alpha_{i,1} = 0, \partial\mathcal{L}/\partial\alpha_{i,2} = 0$. This leads to the following set of equations which should be satisfied simultaneously

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial z} &= -\gamma z + Rz + \sum_{i=1}^{N} \alpha_{i,1} c_{i,1} + \sum_{i=1}^{N} \alpha_{i,2} c_{i,2} = 0 \\ \frac{\partial\mathcal{L}}{\partial w_1} &= \nu w_1 - \sum_{i=1}^{N} \alpha_{i,1} \varphi_1(x_i) = 0 \\ \frac{\partial\mathcal{L}}{\partial w_2} &= \nu w_2 - \sum_{i=1}^{N} \alpha_{i,2} \varphi_2(x_i) = 0 \\ \frac{\partial\mathcal{L}}{\partial e_{i,1}} &= \eta e_{i,1} - \alpha_{i,1} = 0\,,\ i = 1, ..., N \\ \frac{\partial\mathcal{L}}{\partial e_{i,2}} &= \eta e_{i,2} - \alpha_{i,2} = 0\,,\ i = 1, ..., N \\ \frac{\partial\mathcal{L}}{\partial \alpha_{i,1}} &= c_{i,1}^T z - w_1^T \varphi_1(x_i) - e_{i,1}\,,\ i = 1, ..., N \\ \frac{\partial\mathcal{L}}{\partial \alpha_{i,2}} &= c_{i,2}^T z - w_2^T \varphi_2(x_i) - e_{i,2}\,,\ i = 1, ..., N. \end{cases}$$

After elimination of $w_1, w_2, e_{i,1}, e_{i,2}$ and application of the kernel trick the set of equations can be expressed in terms of z, α_1, α_2. One obtains

$$-\gamma z + Rz + V_1\alpha_1 + V_2\alpha_2 = 0$$

and

$$\begin{aligned} V_1^T z - \tfrac{1}{\nu}\Omega_1\alpha_1 - \tfrac{1}{\eta}\alpha_1 = 0 \\ V_2^T z - \tfrac{1}{\nu}\Omega_2\alpha_2 - \tfrac{1}{\eta}\alpha_2 = 0. \end{aligned}$$

The dual model representation follows from the conditions for optimality.

Proof of Lemma 2

The solution (18) is obtained in a straightforward way. Construct the Lagrangian

$$\begin{aligned} \mathcal{L}(z, w_1, w_2, b_1, b_2, e_{i,1}, e_{i,2}; \beta_{1,1}, \beta_{1,2}, \alpha_{i,1}, \alpha_{i,2}) = J_4 + \beta_{1,1}(c_{1,1}^T z - q_1 - e_{1,1}) + \beta_{1,2}(c_{1,2}^T z - q_2 - e_{1,2}) \\ + \sum_{i=2}^{N} \alpha_{i,1}(c_{i,1}^T z - w_1^T \varphi_1(x_i) - b_1 - e_{i,1}) + \sum_{i=2}^{N} \alpha_{i,2}(c_{i,2}^T z - w_2^T \varphi_2(x_i) - b_2 - e_{i,2}). \end{aligned}$$

Taking the conditions for optimality [11] $\partial\mathcal{L}/\partial z = 0, \partial\mathcal{L}/\partial w_1 = 0, \partial\mathcal{L}/\partial w_2 = 0, \partial\mathcal{L}/\partial b_1 = 0, \partial\mathcal{L}/\partial b_2 = 0, \partial\mathcal{L}/\partial e_{i,1} = 0, \partial\mathcal{L}/\partial e_{i,2} = 0, \partial\mathcal{L}/\partial \beta_{1,1} = 0, \partial\mathcal{L}/\partial \beta_{1,2} = 0, \partial\mathcal{L}/\partial \alpha_{i,1} = 0, \partial\mathcal{L}/\partial \alpha_{i,2} = 0$ this gives the following set of equations which should be satisfied simultaneously

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial z} &= -\gamma z + (I - P_D)^T(I - P_D)z + \beta_{1,1}c_{1,1} + \beta_{1,2}c_{1,2} + \sum_{i=2}^N \alpha_{i,1}c_{i,1} + \sum_{i=2}^N \alpha_{i,2}c_{i,2} = 0 \\ \frac{\partial\mathcal{L}}{\partial w_1} &= \nu w_1 - \sum_{i=2}^N \alpha_{i,1}\varphi_1(x_i) = 0 \\ \frac{\partial\mathcal{L}}{\partial w_2} &= \nu w_2 - \sum_{i=2}^N \alpha_{i,2}\varphi_2(x_i) = 0 \\ \frac{\partial\mathcal{L}}{\partial b_1} &= \sum_{i=2}^N \alpha_{i,1} = 1_{N-1}^T\alpha_1 = 0 \\ \frac{\partial\mathcal{L}}{\partial b_2} &= \sum_{i=2}^N \alpha_{i,2} = 1_{N-1}^T\alpha_2 = 0 \\ \frac{\partial\mathcal{L}}{\partial e_{1,1}} &= \eta e_{1,1} - \beta_{1,1} = 0 \\ \frac{\partial\mathcal{L}}{\partial e_{1,2}} &= \eta e_{1,2} - \beta_{1,2} = 0 \\ \frac{\partial\mathcal{L}}{\partial e_{i,1}} &= \eta e_{i,1} - \alpha_{i,1} = 0\,,\ i = 2, ..., N \\ \frac{\partial\mathcal{L}}{\partial e_{i,2}} &= \eta e_{i,2} - \alpha_{i,2} = 0\,,\ i = 2, ..., N \\ \frac{\partial\mathcal{L}}{\partial \beta_{1,1}} &= c_{1,1}^T z - q_1 - e_{1,1} = 0 \\ \frac{\partial\mathcal{L}}{\partial \beta_{1,2}} &= c_{1,2}^T z - q_2 - e_{1,2} = 0 \\ \frac{\partial\mathcal{L}}{\partial \alpha_{i,1}} &= c_{i,1}^T z - w_1^T\varphi_1(x_i) - b_1 - e_{i,1} = 0\,,\ i = 2, ..., N \\ \frac{\partial\mathcal{L}}{\partial \alpha_{i,2}} &= c_{i,2}^T z - w_2^T\varphi_2(x_i) - b_2 - e_{i,2} = 0\,,\ i = 2, ..., N. \end{cases}$$

After elimination of $w_1, w_2, e_{i,1}, e_{i,2}$ and application of the kernel trick the set of equations can be expressed in terms of $z, b_1, b_2, \alpha_1, \alpha_2$. One obtains

$$-\gamma z + (I - P_D)^T(I - P_D)z + V_1\alpha_1 + V_2\alpha_2 + \eta c_{1,1}c_{1,1}^T z + \eta c_{1,2}c_{1,2}^T z = \eta(q_1 c_{1,1} + q_2 c_{1,2})$$

and

$$\begin{aligned} V_1^T z - \tfrac{1}{\nu}\Omega_1\alpha_1 - \tfrac{1}{\eta}\alpha_1 - b_1 1_{N-1} = 0 \\ V_2^T z - \tfrac{1}{\nu}\Omega_2\alpha_2 - \tfrac{1}{\eta}\alpha_2 - b_2 1_{N-1} = 0 \\ \beta_{1,1} = \eta(c_{1,1}^T z - q_1) \\ \beta_{1,2} = \eta(c_{1,2}^T z - q_2). \end{aligned}$$

The dual model representation follows from the conditions for optimality.

Proof of Lemma 3

The proof is similar to the proof of Lemma 2. The Lagrangian is given by

$$\begin{aligned}\mathcal{L}(z, w_j, b_j, e_{i,j}; \beta_{1,j}, \alpha_{i,j}) = J_5 + \sum_{j=1}^{d} \beta_{1,j}(c_{1,j}^T z - q_j - e_{1,j}) \\ + \sum_{j=1}^{d} \sum_{i=2}^{N} \alpha_{i,j}(c_{i,j}^T z - w_j^T \varphi_j(x_i) - b_j - e_{i,j}).\end{aligned}$$

From the conditions for optimality $\partial\mathcal{L}/\partial z = 0, \partial\mathcal{L}/\partial w_j = 0, \partial\mathcal{L}/\partial b_j = 0, \partial\mathcal{L}/\partial e_{i,j} = 0, \partial\mathcal{L}/\partial\beta_{1,j} = 0, \partial\mathcal{L}/\partial\alpha_{i,j} = 0$, one obtains

$$\left\{\begin{array}{rcl} \frac{\partial\mathcal{L}}{\partial z} & = & -\gamma z + (I - P_D)^T(I - P_D)z + \sum_{j=1}^{d} \beta_{1,j} c_{1,j} + \sum_{j=1}^{d} \sum_{i=2}^{N} \alpha_{i,j} c_{i,j} = 0 \\ \frac{\partial\mathcal{L}}{\partial w_j} & = & \nu w_j - \sum_{i=2}^{N} \alpha_{i,j} \varphi_j(x_i) = 0\,,\ j = 1, ..., d \\ \frac{\partial\mathcal{L}}{\partial b_j} & = & \sum_{i=2}^{N} \alpha_{i,j} = 1_{N-1}^T \alpha_j = 0\,,\ j = 1, ..., d \\ \frac{\partial\mathcal{L}}{\partial e_{1,j}} & = & \eta e_{1,j} - \beta_{1,j} = 0\,,\ j = 1, ..., d \\ \frac{\partial\mathcal{L}}{\partial e_{i,j}} & = & \eta e_{i,j} - \alpha_{i,j} = 0\,,\ i = 2, ..., N\,,\ j = 1, ..., d \\ \frac{\partial\mathcal{L}}{\partial\beta_{1,j}} & = & c_{1,j}^T z - q_j - e_{1,j} = 0\,,\ j = 1, ..., d \\ \frac{\partial\mathcal{L}}{\partial\alpha_{i,j}} & = & c_{i,j}^T z - w_j^T \varphi_j(x_i) - b_j - e_{i,j} = 0\,,\ i = 2, ..., N\,,\ j = 1, ..., d. \end{array}\right.$$

After elimination of $w_j, e_{i,j}$ and application of the kernel trick the set of equations can be expressed in terms of z, b_j, α_j. This gives

$$-\gamma z + (I - P_D)^T(I - P_D)z + \sum_{j=1}^{d} V_j \alpha_j + +\eta \sum_{j=1}^{d} c_{1,j} c_{1,j}^T z = \eta \sum_{j=1}^{d} q_j c_{1,j}$$

and

$$\begin{aligned} & V_j^T z - \tfrac{1}{\nu}\Omega_j \alpha_j - \tfrac{1}{\eta}\alpha_j - b_j 1_{N-1} = 0 \\ & \beta_{1,j} = \eta(c_{1,j}^T z - q_j). \end{aligned}$$

The dual model representation follows from the conditions for optimality.

References

[1] C. Alzate, J.A.K. Suykens, "A Weighted Kernel PCA Formulation with Out-of-Sample Extensions for Spectral Clustering Methods," in *Proc. of the 2006 International Joint Conference on Neural Networks* (IJCNN'06), Vancouver, Canada, pp. 138-144, 2006.

[2] M. Belkin, P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, 15(6): 1373-1396, June 2003.

[3] M. Belkin, P. Niyogi, V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *Journal of Machine Learning Research*, 7: 2399–2434, 2006.

[4] 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 *Advances in Neural Information Processing Systems*, 16, 2004.

[5] G.C. Cawley, N.L.C. Talbot, "Fast exact leave-one-out cross-validation of sparse least-squares support vector machines," *Neural Networks*, 17(10): 1467-1475, 2004.

[6] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Natl. Acad. Sci. U.S.A.*, 102(21): 7426-7431, May 2005.

[7] R.R. Coifman, S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, 21(1): 5-30, 2006.

[8] F. Cucker, S. Smale, "On the mathematical foundations of learning theory," *Bulletin of the AMS*, 39, 1–49, 2002.

[9] D.L. Donoho, C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. U.S.A*, 100(10): 5591-5596, May 2003.

[10] T. Evgeniou, M. Pontil, T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics,* 13(1), 1–50, 2000.

[11] R. Fletcher, *Practical Methods of Optimization,* Chichester and New York: John Wiley & Sons, 1987.

[12] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Baltimore MD: Johns Hopkins University Press, 1989.

[13] I.T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer-Verlag, 1986.

[14] G.E. Hinton, R.R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, 313(5786): 504–507, July 2006.

[15] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, 78(9): 1464–1480, 1990.

[16] S. Roweis, L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, 290(5500), 2323–2326, Dec. 2000.

[17] L. Saul, K.Q. Weinberger, F. Sha, J. Ham, D.D. Lee, "Spectral methods for dimensionality reduction," Chapter 16 in O. Chapelle, B. Schölkopf, A. Zien (Eds.), *Semi-supervised learning*, MIT Press, Cambridge Massachusetts, 293–308, 2006.

[18] C. Saunders, A. Gammerman, V. Vovk, "Ridge regression learning algorithm in dual variables," *Proc. of the 15th Int. Conf. on Machine Learning (ICML-98)*, Madison-Wisconsin, 515–521, 1998.

[19] B. Schölkopf, A. Smola, K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation,* 10, 1299–1319, 1998.

[20] B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[21] J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, June 2004.

[22] S. Smale, "Complexity theory and numerical analysis," *Acta Numerica*, 523–551, 1997.

[23] J.A.K. Suykens, J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, 9(3), 293–300, 1999.

[24] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, "A support vector machine formulation to PCA analysis and its kernel version," *IEEE Transactions on Neural Networks*, 14(2): 447–450, March 2003.

[25] J.A.K. Suykens, J. Vandewalle, B. De Moor, "Optimal control by least squares support vector machines," *Neural Networks*, 14(1), 23-35, 2001.

[26] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.

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

[28] V. Vapnik, *Statistical Learning Theory,* John Wiley & Sons, New York, 1998.

[29] J.P. Vert, Y. Yamanishi, "Supervised graph inference," *Advances in Neural Information Processing Systems*, 17: 1433-1440, 2005.

[30] G. Wahba, *Spline Models for Observational Data*, Series in Applied Mathematics, 59, SIAM, Philadelphia, 1990.

[31] K.Q. Weinberger, L.K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR-04), Washington D.C., 2004.

Captions of Figures

Figure 1: (top) Given input data of a spiral data set in a 3-dimensional space (training data (blue *), validation data (magenta o), test data (red +)); (middle-bottom) Kernel maps with a reference point: illustration of the role of the reference point. Each plot shows the result for a different choice of the reference point: $[+1; -1]$ and (bottom) $[+1; +1]$. The positions of the reference points are indicated by the arrows. The tuning parameters are based on the use of a validation set. The method achieves good generalization on test data.

Figure 2: Kernel maps with a reference point: illustration of Algorithm 1 using a 10-fold cross-validation procedure on the spiral data. (Left) two best results selected according to criterion $\mathcal{Q}_{\text{nmds}}$; (Right) two best results according to criterion $\mathcal{Q}_{\text{nlle}}$.

Figure 3: Algorithm 1 with 10-fold cross-validation: (Top) $\mathcal{Q}_{\text{nmds}}$ criterion with respect to a running index for a grid search process; (Bottom) $\mathcal{Q}_{\text{nlle}}$ criterion with respect to the running index. The different curves on both plots correspond to the 10 runs in the 10-fold cross-validation.

Figure 4: Spiral data set: comparison with existing methods. (Left) (top) MDS, (middle) PCA, (bottom) LLE; (Right) (top) Laplacian eigenmap, (middle) diffusion map, (bottom) comparison with solution to eigenvalue problem for problem P1, which is numerically ill-conditioned.

Figure 5: (Top) Swiss roll data set (containing 600 training data points and 100 validation set points); (middle) 2D visualization with dimensionality reduction using the kernel map with reference point; (bottom) application of Algorithm 2 with a randomized validation set. The figure shows a result for which the criteria $\mathcal{Q}_{\text{nmds}}, \mathcal{Q}_{\text{nlle}}$ indicate a good generalization performance.

Figure 6: Swiss roll data: comparison with existing methods. (Left) (top) MDS, (middle) PCA, (bottom) LLE with 50 nearest neighbors; (Right) (top) LLE with 200 nearest

neighbors, (middle) Laplacian eigenmap with 300 nearest neighbors (only a large number reveals the rolling structure), (bottom) diffusion map. Tuning parameters were set to reveal similar structure as the kernel map with reference point method.

Figure 7: Alon colon cancer microarray data set: 2D visualization of the gene distribution obtained by the kernel map reference point method (training data (blue *), validation data (black o), test data (red +)). The top and bottom figures show the result for different assignments of training and test parts of the data, which does not affect the obtained structure.

Figure 8: Alon data set: comparison of different methods. (Top) (left) kernel map with reference point $[-1; +1; +1]$, (right) PCA; (Middle) (left) LLE, (right) Laplacian eigenmap; (Bottom) (left) MDS, (right) diffusion map. Tuning parameters were set to reveal similar structure as the kernel map with reference point method.

Figure 9: (Top) Santa Fe chaotic laser data in time: validation set (black) - training set (blue) - test set (red); (Bottom) 2D visualization obtained by the kernel map reference point method (training data (blue *), validation data (black o), test data (red +)).

Figure 10: Santa Fe data set: comparison with existing methods. (Top) (left) kernel map with reference point $[-1; +1; -1]$, (right) PCA; (Middle) (left) LLE with 100 nearest neighbors, (right) Laplacian eigenmap with 300 nearest neighbors (note the large number needed in this example); (Bottom) (left) MDS, (right) diffusion map. Tuning parameters were set to reveal similar structure as the kernel map with reference point method.

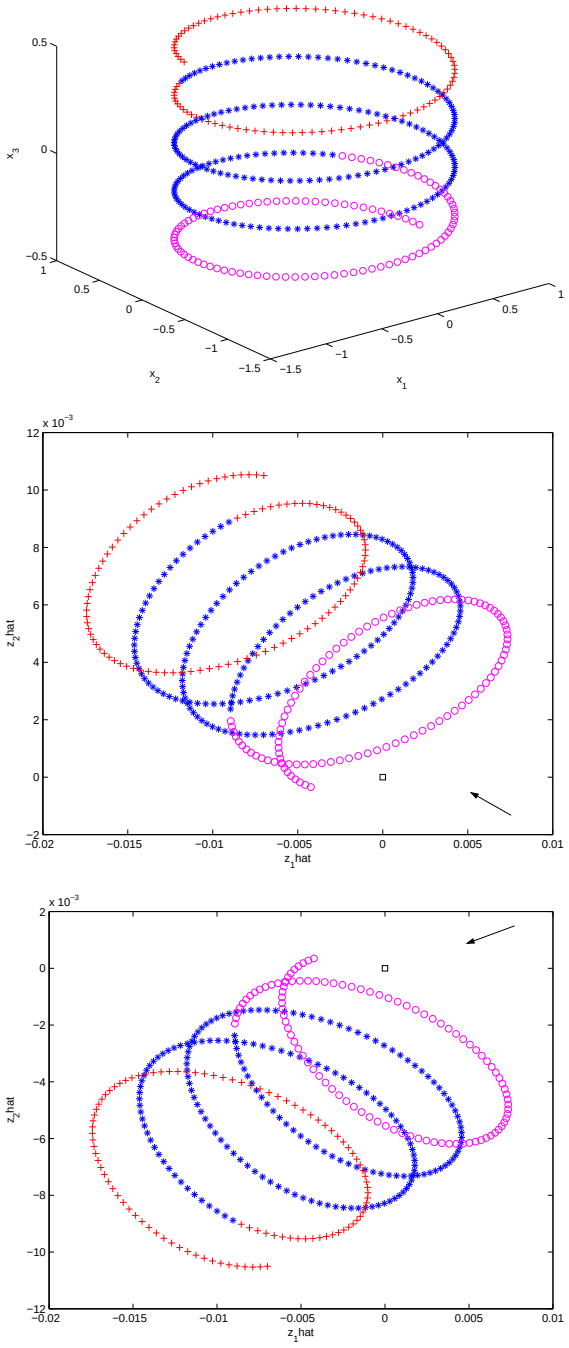

Figure 1

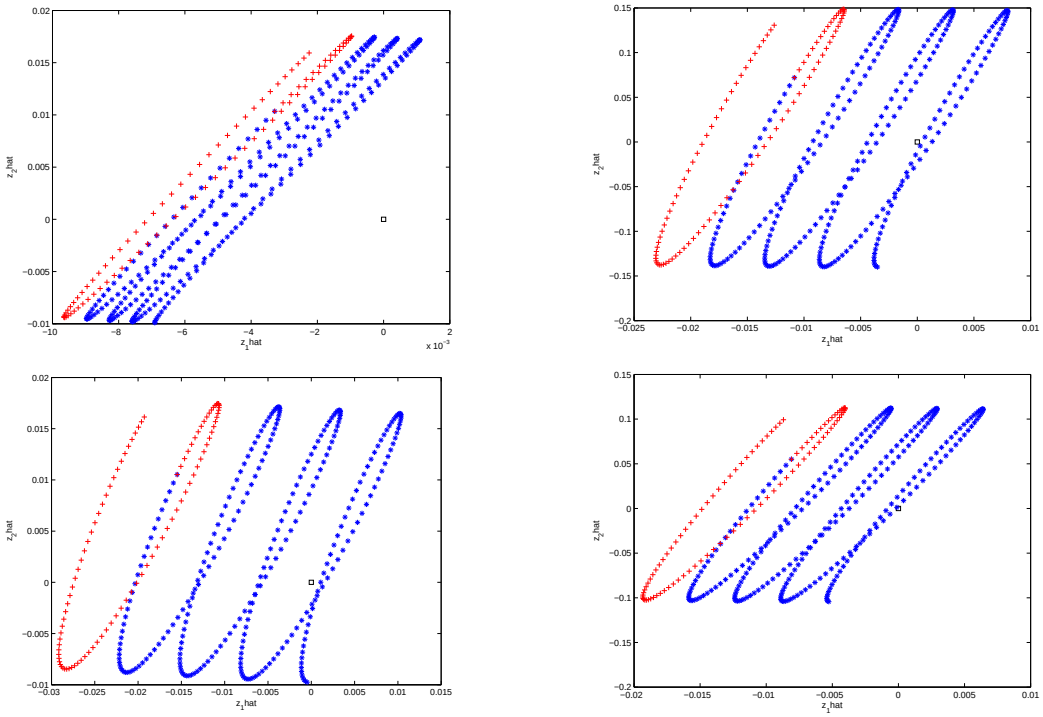

Figure 2

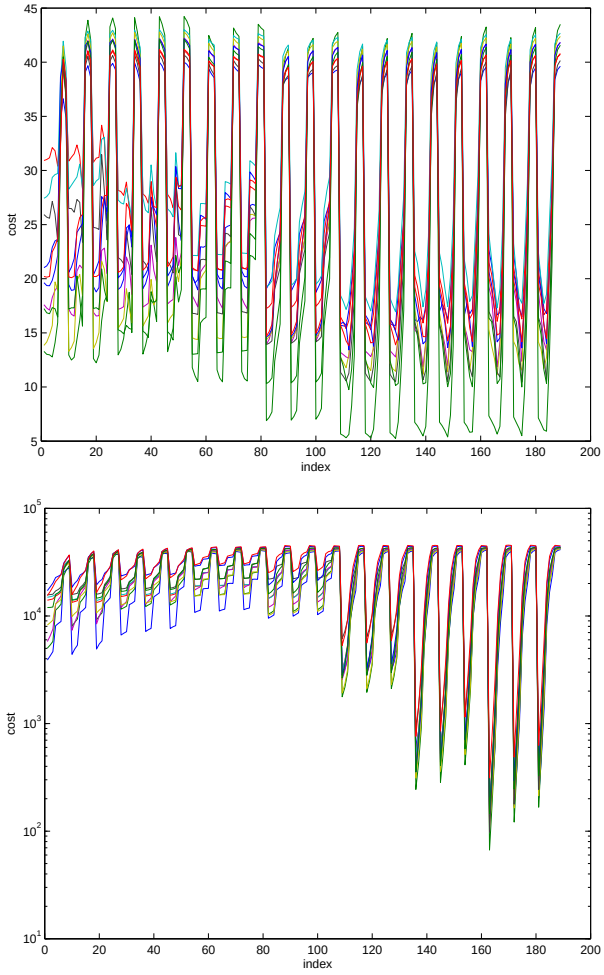

Figure 3

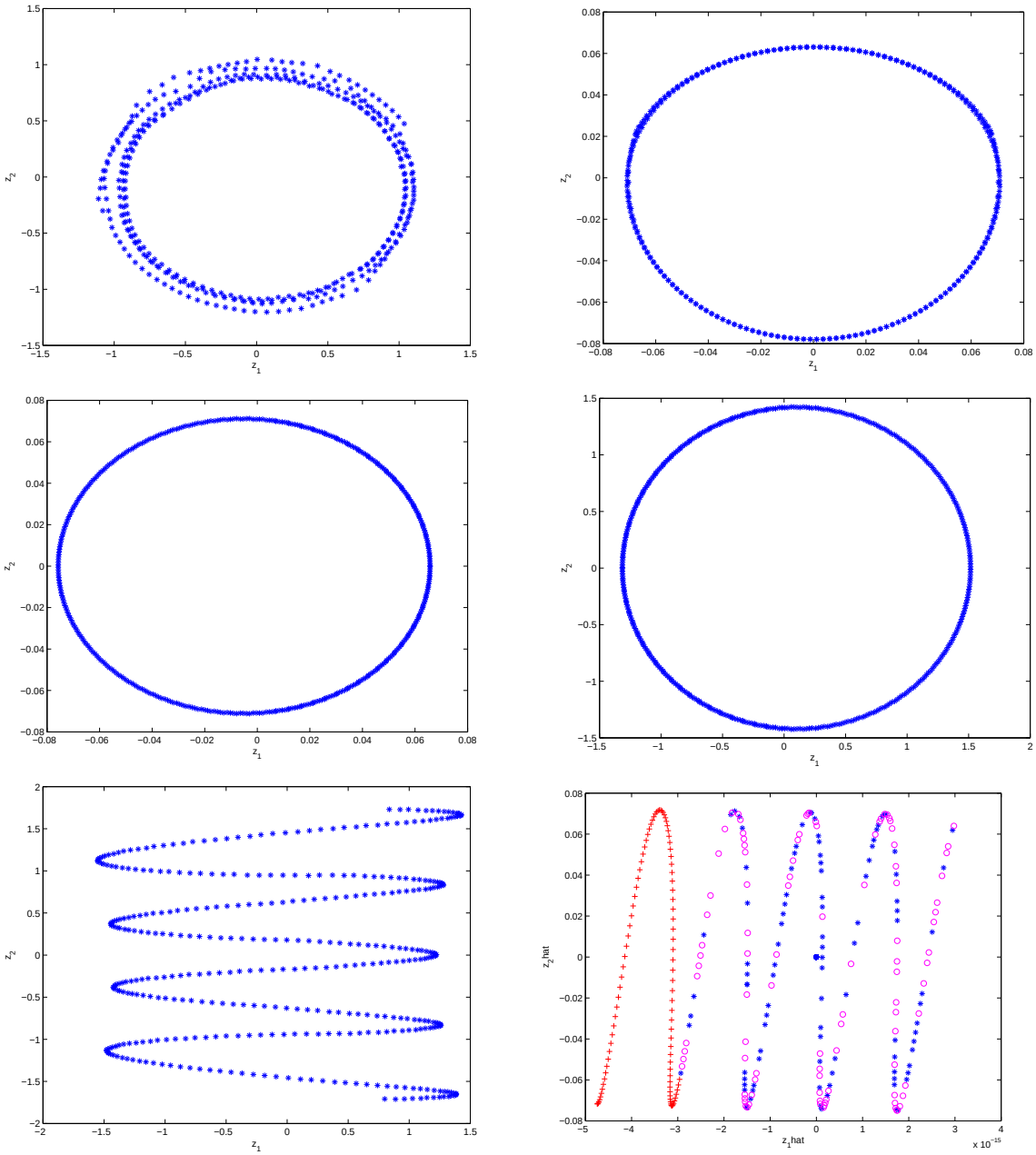

Figure 4

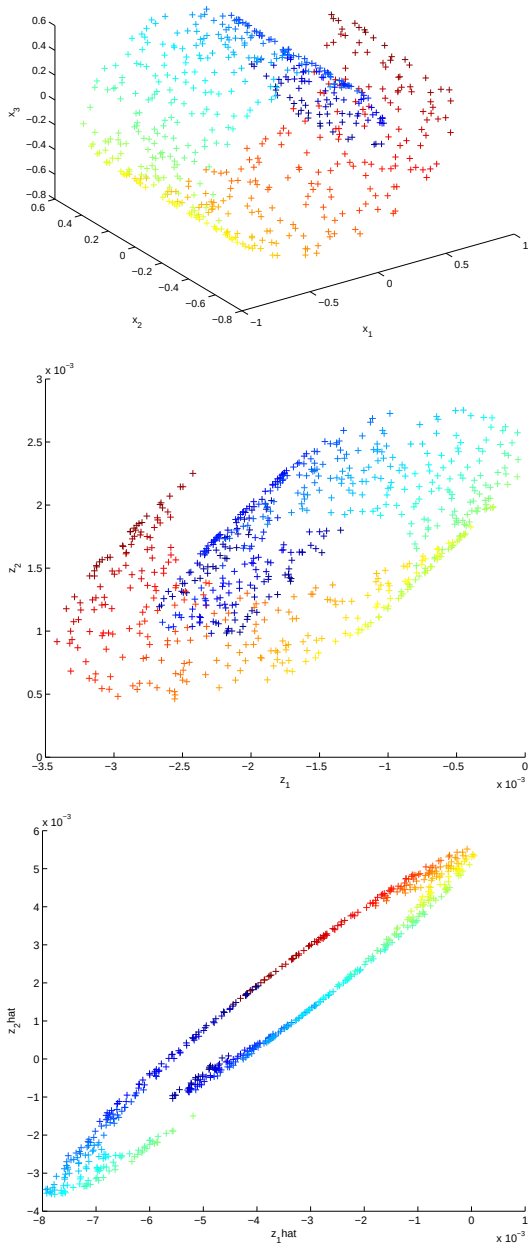

Figure 5

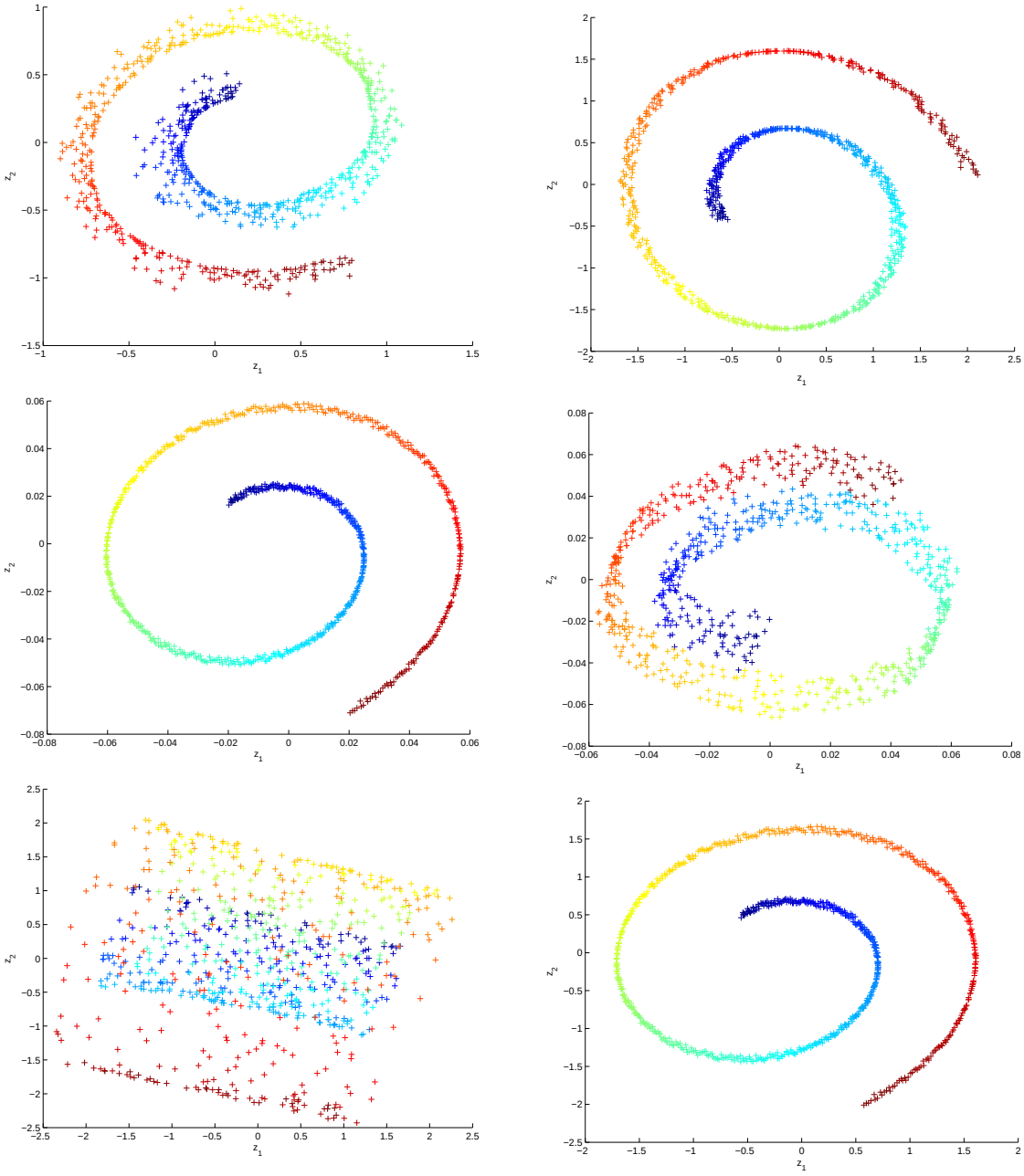

Figure 6

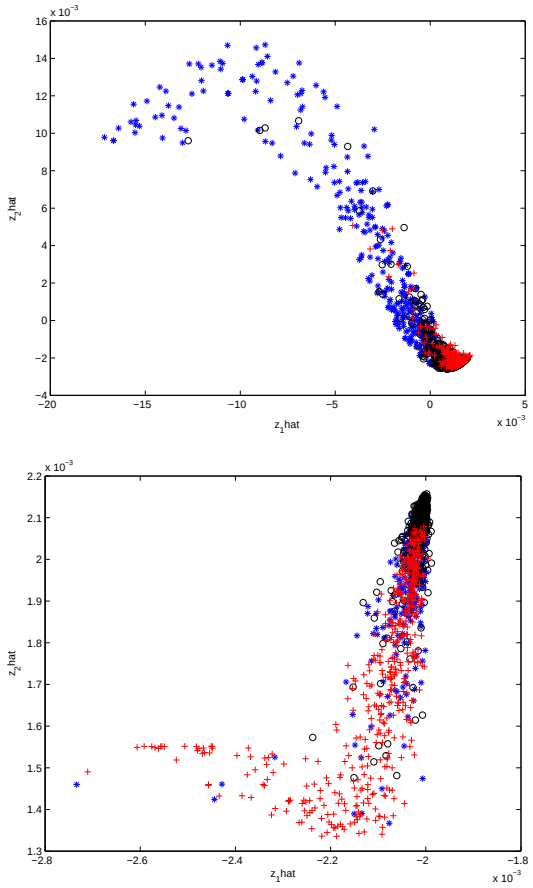
x 10^-3
16
14
12
10
8
6
4
2
0
−2
−4
−20
−15
−10
−5
0
5
x 10^-3
z_2hat
z_1hat
x 10^-3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
−2.8
−2.6
−2.4
−2.2
−2
−1.8
x 10^-3
z_2hat
z_1hat

Figure 7

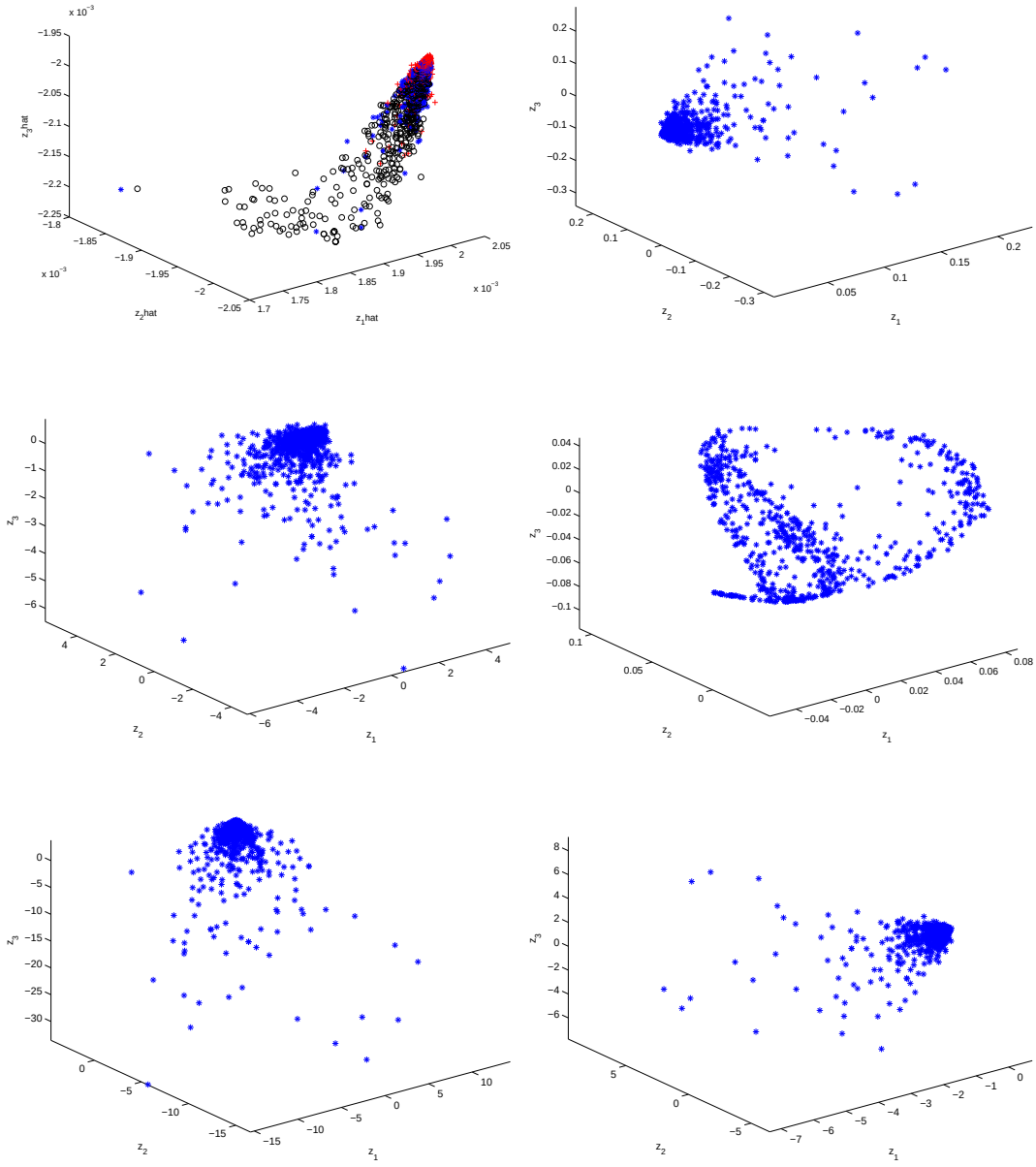

Figure 8

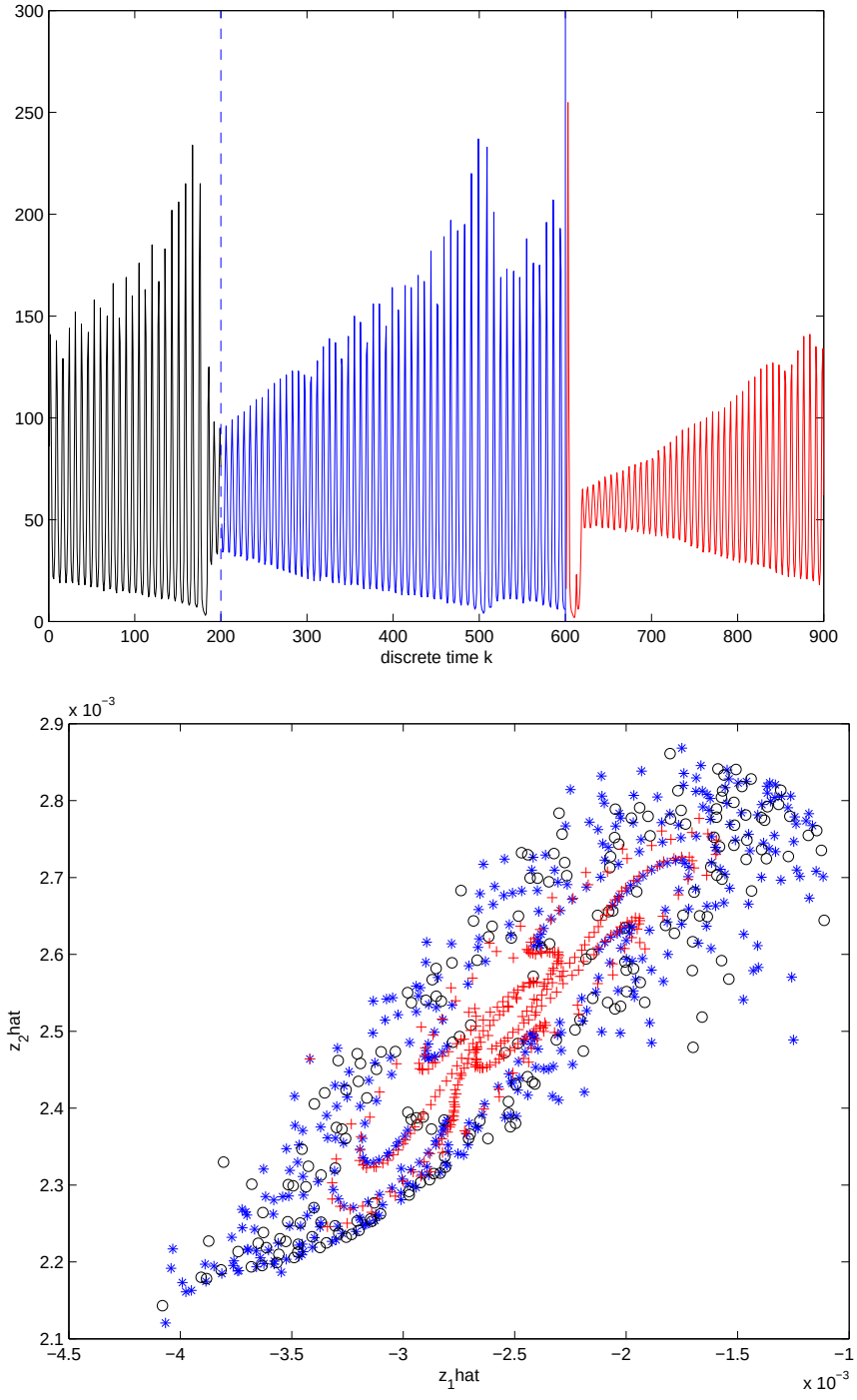
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
discrete time k
x 10^-3
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
z_2hat
−4.5
−4
−3.5
−3
−2.5
−2
−1.5
−1
z_1hat
x 10^-3

Figure 9

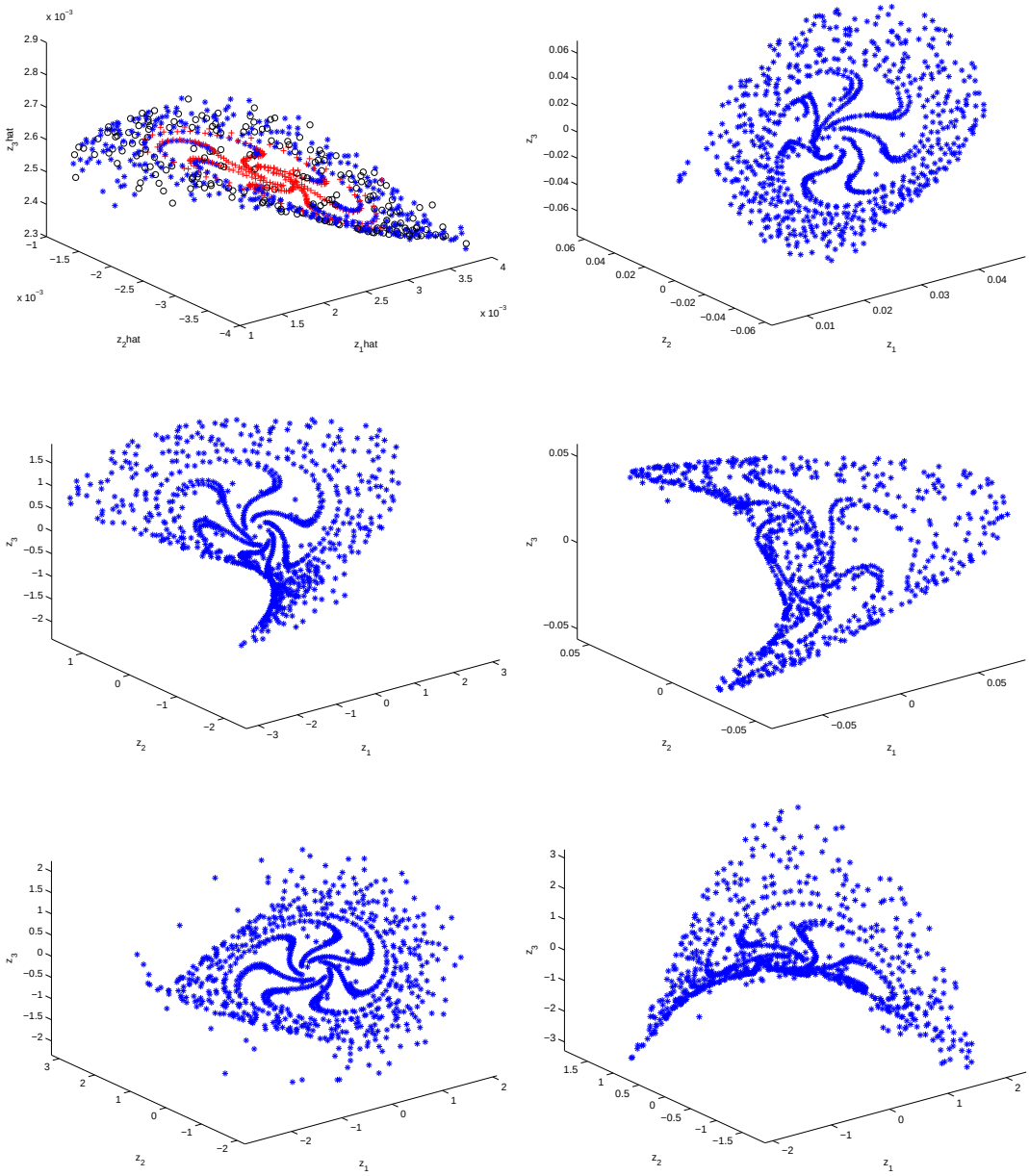

Figure 10