Kernel Sammon Map

Fernando K. Inaba, Evandro O. T. Salles,
Departamento de Engenharia Elétrica, Centro Tecnológico
Universidade Federal do Espírito Santo
29060-970 Vitória, Brazil
Email: fkinaba@ele.ufes.br, evandro@ele.ufes.br

Thomas W. Rauber
Departamento de Informática, Centro Tecnológico
Universidade Federal do Espírito Santo
29060-970 Vitória, Brazil
Email: thomas@inf.ufes.br

***Abstract*—We extend the visualization technique of high-dimensional patterns conceived by Sammon to the case when the patterns have been previously mapped to an implicitly defined Hilbert feature space in which distances can be measured by kernels. The principal benefit of our technique is the possibility to gain insight into the distribution of the patterns, even in this generally non-accessible feature space.**

***Keywords*-Sammon map, Hilbert feature space, Kernels, Visualization, Kernel Principal Component Analysis.**

I. Introduction

The visualization of patterns residing in a high dimensional feature space is one of the main concerns of pattern recognition. Van der Maaten et al. [17] give an overview of dimensionality reduction techniques. A classical method to map patterns from a high-dimensional continuous feature space to two or three dimensions is the Sammon map [11]. A stress function is defined which measures the discrepancy between the mutual distances in the original feature space and the mutual distances in the projected feature space. Usually the projected feature space has two or three dimensions to permit the visualization, but in general the dimension of the projected space is not restricted.

In recent years, the use of feature mapping into Hilbert space, together with kernel based inner product similarity measures[1] has gained considerable attention, because its introduces an additional, in general non-linear, mapping from the original feature space to an intermediate space which should improve regression (cf. for example, [16]) and/or classification quality of the new features. Especially the Support Vector Machine has drawn much attention to kernel-based implicit feature mapping, although many other classification techniques can be enhanced by kernels, for instance the Kernel Nearest-Neighbor classifier or Kernel Principal Component Analysis mentioned later in this paper.

If we want to introduce kernel-based mapping into the Sammon mapping visualization technique, consequently we deal with two different mappings, from the original, eventually non-numeric feature space to the intermediate implicitly defined Hilbert space where inner products can be calculated by the kernel, and finally the Sammon map which transforms the pattern from the intermediate feature space to the final, usually low-dimensional Euclidean feature space, where is possible to apply common visualization techniques.

An approach to use kernels in Sammon mapping was presented by Mingbo Ma et al. [6]. The mapping of training patterns and the interpolation of new patterns are combined in a unique mapping function which is a linear combination of the kernel expanded unknown pattern with the Sammon mapped training patterns. Our work is motivated by the fact that the distances between two patterns that have been mapped to the implicit feature space prior to the Sammon mapping can be included directly into the stress function that measures the quality of the Sammon mapping. Besides, we propose an interpolation method for new patterns based on a linear combination of already mapped patterns. Our main contribution is the direct formulation of the Sammon stress function based on kernel-based distances in the implicit feature space and a novel linear interpolation of new patterns based on the mapped training patterns.

The rest of the paper is organized in the following manner: Section II reviews the nonlinear mapping proposed by Sammon. In section III implicit kernel mapping and the distance measure in the kernel mapped space is discussed. Section IV incorporates the kernel mapping into the subsequent Sammon mapping. Section V analyzes the important case when new, previously unseen patterns have to be Sammon mapped. Experimental results for the visualization of kernel enhanced pattern mappings are presented in section VI and finally the conclusions are drawn in section VII.

II. Sammon plot

In its original form [11], the Sammon map $\mathbf{y}$

$$\begin{aligned} \mathbf{y} : \mathcal{X} &\rightarrow \mathcal{Y} \\ \mathbf{x} &\mapsto \mathbf{y} \end{aligned} \tag{1}$$

is a nonlinear projection of n patterns $\mathbf{x}_i, i = 1, \ldots, n$ in a D-dimensional Euclidean space $\mathcal{X}$ onto n corresponding patterns $\mathbf{y}_i, i = 1, \ldots, n$ in another d-dimensional Euclidean space $\mathcal{Y}$, $d \leq D$, that should preserve the notion of mutual geometric distances among the patterns. For direct visualization purposes the mapped dimension is set to $d = 2$ or $d = 3$. The faithfulness of the mapping is naturally limited by the intrinsic dimension of the data. The end points $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ of the axes of $\mathbb{R}^3$, for example, can be mapped onto the vertices

[1] For the sake of simplicity, we will use the term “kernel mapping” to identify the mapping from the original feature space into Hilbert space where kernels can be used to calculate an inner product to measure similarity.

of an equilateral triangle in $\mathbb{R}^2$ without error, but not onto the x-axis of $\mathbb{R}^1$ anymore, because there it is impossible to position three distinct points with equal mutual distances there.

We will use the following abbreviations, with $||\mathbf{a}-\mathbf{b}|| = \left[(\mathbf{a}-\mathbf{b})^\mathsf{T}(\mathbf{a}-\mathbf{b})\right]^{1/2}$ as the Euclidean distance between two column vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\begin{array}{lll} D_{ij} := & ||\mathbf{x}_i - \mathbf{x}_j|| & \text{Distance in original feature space } \mathcal{X} \\ d_{ij} := & ||\mathbf{y}_i - \mathbf{y}_j|| & \text{Distance in mapped feature space } \mathcal{Y} \end{array} \tag{2}$$

A stress function

$$E(\{\mathbf{y}_1, \dots, \mathbf{y}_n\}) = \frac{1}{\sum_{i<j}^{n} D_{ij}} \sum_{i<j}^{n} \frac{(D_{ij} - d_{ij})^2}{D_{ij}} \tag{3}$$

expresses a discrepancy between all mutual distances in the original $\mathcal{X}$ and mapped space $\mathcal{Y}$. In Sammon's original work the stress function is minimized by an unconstrained optimization method, namely gradient descent. Note that the $n(n-1)/2$ mutual distances D_{ij} in $\mathcal{X}$ have to be calculated only once, whereas the distances in $\mathcal{Y}$ have to be constantly updated during the minimization of the stress function (3), since the positions of the $\mathbf{y}_i$ change (they are the parameters of the gradient descent). The normalizing constant $1/\sum_{i<j}^{n} D_{ij}$ has no influence on the determination of a minimum of (3), since it does not affect the gradient descent with respect to the necessary condition of a minimum $\nabla E = 0$, and could consequently be omitted from (3).

An observation when comparing our method with the method proposed by Mingbo Ma et al. [6] is the nonlinear nature of the Sammon map, i.e. in general there is no affine function, implementable as a matrix multiplication (plus eventually a constant offset) $\mathbf{y}(\mathbf{x}) = \mathsf{M} \cdot \mathbf{x} + \mathbf{y}_0$ that produces the mapping defined in (1).

III. Distances between kernel mapped patterns

Kernel based regression and classification has gained a lot of attention, especially in the context of the Support Vector Machine (SVM) [5], [18]. There are more fundamental pattern recognition methods that were enhanced by the implicit kernel mapping principle prior to the proper technique, such as Fisher Linear Discriminant Analysis [7], Multilayer Perceptron [9], Mahalanobis Distance [10], Mean Squared Error [10], Kernel Principal Component Analysis (kernel PCA) [14] or Nearest Neighbor Classifier [19], without being exhaustive.

A. Implicit kernel map

The basic idea is to use a map $\boldsymbol{\phi}$

$$\begin{aligned} \boldsymbol{\phi} : \mathcal{X} &\to \mathcal{H} \\ \mathbf{x} &\mapsto \boldsymbol{\phi}(\mathbf{x}) \end{aligned} \tag{4}$$

which projects a feature vector $\mathbf{x}$ of dimension D to a new feature vector $\boldsymbol{\phi}(\mathbf{x})$ of dimension $\ell \leq \infty$. This map however may not be defined explicitly in general. The metric of similarities and consequently distances in the mapped $\mathcal{H}$ space [2] must exclusively be expressed as an inner product $\boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_j)$ of the mapped versions of the original feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ when implementable calculus[3] has to be done. This constraint enables the use of kernels k that implement the inner product in the transformed space as

$$k(\mathbf{x}_i, \mathbf{x}_j) := \boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_j). \tag{5}$$

As examples consider the commonly employed Radial Basis Function kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/(2\sigma^2))$ with spread parameter σ^2 or the inhomogeneous polynomial kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$ which maps to all possible monomials up to degree p, see e.g. [12]. The mapped space $\mathcal{H}$ is a Hilbert space where an inner product is defined, whereas the features in the original pattern space $\mathcal{X}$ are not restricted to be continuous values. By virtue of an appropriate kernel, the original features can be of symbolic, discrete or continuous nature, or any mix of these.

B. Kernel based distances of mapped patterns

The idea of kernels does appear early as Potential Functions in Aizerman et al. [2]. The same authors do also present the idea of a kernel version of the Nearest Neighbor Classifier [3]. The kernel version of the Euclidean distance was called Generalized Distance in [1] and redefined by Yu et al. as the Nearest Neighbor [19]. The idea can easily be seen by expanding the (quadratic)[4] Euclidean distance $||\mathbf{x}_i - \mathbf{x}_j||^2$ between a pattern $\mathbf{x}_i$ and a pattern $\mathbf{x}_j$ as

$$D_{ij}^2 = ||\mathbf{x}_i - \mathbf{x}_j||^2 = \mathbf{x}_i \cdot \mathbf{x}_i - 2\mathbf{x}_i \cdot \mathbf{x}_j + \mathbf{x}_j \cdot \mathbf{x}_j \tag{6}$$

and then substituting the pattern $\mathbf{x}$ by its mapped version $\boldsymbol{\phi}(\mathbf{x})$ to get

$$\begin{aligned} D_{ij}^2 &= ||\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)||^2 = \\ &\boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_i) - 2\boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_j) + \boldsymbol{\phi}(\mathbf{x}_j) \cdot \boldsymbol{\phi}(\mathbf{x}_j) = \\ &k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j). \end{aligned} \tag{7}$$

Consequently distances between patterns in Hilbert space can be measured by virtue of the kernel function, although the proper patterns are not accessible.

In the context of the Kernel Nearest Neighbor Classifier, the pattern $\mathbf{x}_i$ assumes the role of an unknown pattern to be classified and $\mathbf{x}_j$ the role of a class-labeled training pattern. Hence we can measure distances in the mapped space between unknown patterns and the training patterns by using the employed kernel thus implementing the Kernel K-Nearest Neighbor classifier, $K \geq 1$. In the context of the Sammon map, we are however only concerned with unsupervised mapping.

IV. Sammon mapped kernel mapped patterns

We combine the mapping $\boldsymbol{\phi}$ of (4) from the original feature space $\mathcal{X}$ to the kernel-based implicitly mapped Hilbert space $\mathcal{H}$ with the Sammon mapping $\mathbf{y}$ of (1), now from the Hilbert

[2] $\mathcal{H}$ stands for the *Hilbert space* but also suggests *hidden*, since the patterns $\boldsymbol{\phi}(\mathbf{x})$ in this space in general are only implicitly available.

[3] Patterns $\boldsymbol{\phi}(\mathbf{x})$ in Hilbert space in general cannot be stored in computer memory explicitly. They can only be handled in analytical calculus.

[4] We keep the distance in the quadratic form D_{ij}^2 to simplify the equations (6) and (7). Later, in (9), we need the distance D_{ij}.

space $\mathcal{H}$ to the final low-dimensional visualization space $\mathcal{Y}$ to define a new Sammon map $\mathbf{y}$

$$\begin{aligned} \mathbf{y} : \mathcal{X} &\to \mathcal{H} \to \mathcal{Y} \\ \mathbf{x} &\mapsto \phi(\mathbf{x}) \mapsto \mathbf{y}(\phi(\mathbf{x})). \end{aligned} \tag{8}$$

The mapping pipeline is shown in fig. 1. From the original feature space $\mathcal{X}$ which is not necessarily Euclidean, the patterns are mapped implicitly to the intermediate feature space $\mathcal{H}$ which is not directly visualizable for two reasons. Firstly the map ϕ is not defined explicitly and secondly the dimension of $\mathcal{H}$ is generally too high. The patterns in $\mathcal{H}$ are finally mapped by the Sammon map $\mathbf{y}$ to the final feature space $\mathcal{Y}$.

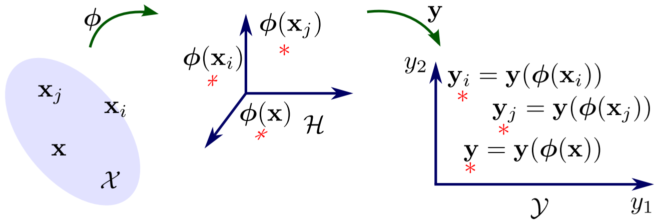


Fig. 1. Mapping pipeline when using an intermediate Hilbert space in the Sammon plot.

The impact on the stress function (3) is minimal. Only the distances D_{ij} in the originally Euclidean space $\mathcal{X}$ of (2) and (6) have to be substituted by the new distances D_{ij} of (7) in the Hilbert space $\mathcal{H}$

$$D_{ij} = \left[k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j)\right]^{1/2}. \tag{9}$$

Again, the distances D_{ij} in Hilbert space have to be calculated only once, such that the complexity of the stress function minimization algorithm is not affected at all. We are now able to visualize patterns $\mathbf{x}$ that have previously been mapped implicitly to Hilbert space by $\phi(\mathbf{x})$, and then measuring distances by virtue of a kernel, permitting to gain insight into the situation of the pattern separability in the implicitly defined space. For instance, in Support Vector Machine regression and classification, the influence of different kernels and the variation of their intrinsic parameters can be made visible.

V. Mapping of New Patterns

We consider the case when a previously unseen new pattern $\mathbf{x}_{n+1}$ should be Sammon mapped by (8) together with the existing n patterns which we will call the training patterns $\mathbf{x}_i, i = 1, \ldots, n$.

A. Error-free new pattern mapping

As mentioned before, the Sammon map is not an affine function. The only way to guarantee that the new pattern(s) $\mathbf{x}_{n+1}$ are projected onto $\mathcal{Y}$ by the same Sammon map as all n training patterns, is to include the new pattern into the same stress function (3) to be minimized as before. The same minimum must be reached, either by including pattern $\mathbf{x}_{n+1}$ beforehand as a training pattern, or after the training has been done. Note that "error-free" does not mean that the map is globally optimized, but only that the same minimum is reached.

Omitting the constant $1/\sum_{i<j}^{n} D_{ij}$ in (3) which does not affect the gradient descent determination of the minimum, we define an equivalent stress function

$$E'_n := E'(\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}) = \sum_{i<j}^{n} \frac{(D_{ij} - d_{ij})^2}{D_{ij}}. \tag{10}$$

If we want to include a new pattern $\mathbf{x}_{n+1}$ into the mapping, the new stress function E'_{n+1} can be decomposed into the the term of the training patterns plus a term of n new discrepancies as

$$E'_{n+1} = E'_n + \sum_{i=1}^{n} \frac{(D_{i,n+1} - d_{i,n+1})^2}{D_{i,n+1}}. \tag{11}$$

This means that we only have to minimize the second term in (11), since E'_n is already optimized, thus avoiding a complete re-training of the Sammon map. This optimization is a relatively cheap operation compared to the global minimization of E'_{n+1}, namely the $(n+1)$-th part of it.

B. New patterns as linear combinations

Since the Sammon plot of a pattern $\mathbf{y}(\mathbf{x})$ in its original form (1) or in the kernel enhanced version $\mathbf{y}(\phi(\mathbf{x}))$ of (8) is not an affine function, it cannot be implemented as a matrix multiplication. We consider only the more general kernel version here, since the linear kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ with the identity $\phi(\mathbf{x}) = \mathbf{x}$ as the implicit map specializes to the original Sammon map. Consequently a mapping

$$\mathbf{y}(\phi(\mathbf{x})) = \mathsf{M} \cdot \phi(\mathbf{x}) \tag{12}$$

can only be an approximation of (8). For the sake of simplicity, we have omitted a constant offset term $\mathbf{y}_0$ in (12),

Consider a $n \times \ell$ matrix $\boldsymbol{\Phi}$ which contains as its i-th row the transposed map $\phi(\mathbf{x}_i)^\mathsf{T}$ of pattern $\mathbf{x}_i$, $i = 1, \ldots, n$

$$\boldsymbol{\Phi} := \begin{bmatrix} \phi(\mathbf{x}_1) & \cdots & \phi(\mathbf{x}_n) \end{bmatrix}^\mathsf{T}. \tag{13}$$

Note that the number of columns ℓ theoretically can be infinite (e.g. using the Radial Basis Function kernel), and that $\boldsymbol{\Phi}$ does only appear in an intermediate, implicit calculus. The n-dimensional product vector

$$\mathbf{k}_\mathbf{x} := \begin{bmatrix} k(\mathbf{x}, \mathbf{x}_1) & \cdots & k(\mathbf{x}, \mathbf{x}_n) \end{bmatrix}^\mathsf{T} = \boldsymbol{\Phi} \cdot \phi(\mathbf{x}) \tag{14}$$

is the empirical kernel map [13], [10] of the new pattern $\mathbf{x}$.

We consider especially the work of [6] since it uses kernels and Sammon mapping. They define a function

$$\mathbf{y}(\mathbf{x}) = \mathsf{W}^\mathsf{T} \mathbf{k}_\mathbf{x}, \tag{15}$$

where W is a weight matrix and $\mathbf{k}_\mathbf{x}$ the empirical kernel map of (14). When we combine the $n \times d$ weight matrix W with the mapped training pattern matrix $\boldsymbol{\Phi}$ in (13) and define the $n \times \ell$ matrix $\mathsf{M} := \mathsf{W}^\mathsf{T} \cdot \boldsymbol{\Phi}$, we can identify the kernel enhanced Sammon map which should to be able to map the existing n training and new patterns, as a linear approximation of the Sammon map in the form of (12). We can state that [6] is a linear approximation of already mapped patterns for new patterns $\mathbf{x}_{n+1}, \ldots$ that did not contribute to the generation of the matrix W. Only the n training patterns $\mathbf{x}_1, \ldots \mathbf{x}_n$ were mapped by a nonlinear Sammon map (the n rows of W). This becomes even more evident when using the linear kernel with the identity mapping $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ with all n patterns (called strict interpolation with the number of prototypes $H = n$ in [6]), since $\mathsf{W}^\mathsf{T} \cdot \mathsf{X}$ can be identified as the matrix M in (12), where X has as its i-th row the transposed training pattern $\mathbf{x}_i$ and W as its i-th row the transposed Sammon mapped training pattern $\mathbf{y}(\mathbf{x}_i)$. We consequently rise the hypothesis, since the technique is linear, that for the learning of the weight matrix, once the n training patterns have been mapped by the Sammon mapping in the conventional way, no iterative gradient descent is needed for the mapping of further test patterns. The matrix M can quite probably be obtained by the pseudoinverse based approach presented by ourselves in section V-C. Besides, it seems that in the work of Mingbo Ma et al., first a geodesic function δ_{ij} between two patterns in the original Euclidean space is defined which is then passed as an argument to a kernel function, i.e. the kernel is not used to measure a distance in Hilbert space.

C. Assuming isometry for linear map

In the following we present our own version of a linear approximation of the kernel enhanced Sammon map. An obvious advantage of our method is its linear nature, hence no unconstrained optimization is needed to learn the mapping, only linear algebra. Linear interpolation for a subset of nonlinearly mapped patterns was employed by Paulovich et al. [8], restricted however to the conventional Euclidean spaces problem of (1), whereas we want to visualize patterns residing in the usually inaccessible Hilbert space.

We express a new pattern $\boldsymbol{\phi}(\mathbf{x})$ in the kernel mapped space $\mathcal{H}$ as a linear combination

$$\boldsymbol{\phi}(\mathbf{x}) = \sum_{i=1}^{n} \beta_i \boldsymbol{\phi}(\mathbf{x}_i) \qquad (16)$$

of all n kernel mapped training patterns $\boldsymbol{\phi}(\mathbf{x}_i)$ and assume that the Sammon mapped (8) new pattern $\mathbf{y}(\boldsymbol{\phi}(\mathbf{x}))$ reflects the same linear combination in the final space $\mathcal{Y}$ using the same n coefficients β_i as

$$\mathbf{y}(\boldsymbol{\phi}(\mathbf{x})) = \sum_{i=1}^{n} \beta_i \mathbf{y}(\boldsymbol{\phi}(\mathbf{x}_i)). \qquad (17)$$

Multiplying both sides of (16) by $\boldsymbol{\phi}(\mathbf{x}_j)$, we obtain

$$\boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}_j) = \sum_{i} \beta_i k(\mathbf{x}_i, \mathbf{x}_j). \qquad (18)$$

The left hand side of (18) can be identified as the j-th component $k(\mathbf{x}, \mathbf{x}_j)$ of the empirical kernel map (14). The real symmetric matrix

$$\mathsf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)], i, j = 1, \ldots, n \qquad (19)$$

of the $d-$dimensional patterns $\mathbf{x}_i, i = 1, \ldots, n$ is defined as the Gram matrix (Kernel matrix) [15]. In our work we use only positive definite kernels which guarantee a diagonizable Kernel matrix with the Singular Value eigendecomposition [10]

$$\mathsf{K} = \mathsf{B}\Omega\mathsf{B}^\mathsf{T}, \qquad (20)$$

where the columns of the orthogonal matrix B are the eigenvectors of K and Ω is a diagonal matrix with p non-zero positive real eigenvalues λ_k on its diagonal and $(n - p)$ zero eigenvalues. The rank of the Kernel matrix K is equal to the number p of non-zero eigenvalues.

The kernel matrix K has no inverse, since its rank p by construction is at most $d \approx p << n$, or even smaller when the features are correlated, where n is the number of samples and d is the dimension of the original feature vector. It has however a generalized inverse (aka pseudoinverse) $\mathsf{K}^\dagger$ which can be obtained, from the result of the SVD of (20) as

$$\mathsf{K}^\dagger = \mathsf{B}\Omega^{-1}\mathsf{B}^\mathsf{T}, \qquad (21)$$

where the matrix Ω^{-1} is composed of the inverted eigenvalues $1/\lambda_k$ of K at the corresponding diagonal positions and zero where the eigenvalues were (numerically) zero, i.e. $\Omega^{-1} = \mathrm{diag}(1/\lambda_1, \ldots, 1/\lambda_p, 0, \ldots, 0)$.

Considering (19) and (18), with $\boldsymbol{\beta}$ being the n-dimensional vector of the coefficients β_i, we can write

$$\mathbf{k}_x = \mathsf{K}\boldsymbol{\beta}, \qquad (22)$$

such that the coefficients in a least square approximation, using the pseudoinverse (21) become

$$\boldsymbol{\beta} = \mathsf{K}^\dagger \mathbf{k}_x. \qquad (23)$$

Having obtained the coefficients $\boldsymbol{\beta}$ permits the linear interpolation of new patterns $\mathbf{x}$, previously implicitly mapped as $\boldsymbol{\phi}(\mathbf{x})$, using (17) as $\mathbf{y}(\boldsymbol{\phi}(\mathbf{x}))$.

VI. EXPERIMENTS

In the following we illustrate the usefulness of the kernel enhanced version of the Sammon map by visualizing pattern sets in the usually inaccessible mapped pattern space $\mathcal{H}$ and comparing our approach of linear interpolation of new patterns to the error-free orthodox Sammon map.[5]

We used the following data sets:

1) Circular Data: A synthetic data set composed of 100 patterns in 3-D space, evenly distributed along a circle of radius 2.5.
2) Gaussian distributed data at vertices of a simplex: This data set is similar to the one used in Sammon's original paper [11]. It consists of 20 points sampled from a normal distribution at each of the five vertices of a 4-dimensional simplex considered as a mean vector, totalizing 100 points.
3) Iris Data: Classical data set used by Fisher, composed of three times 50 patterns of dimension four (length and width of sepal and petal of three iris flower species (setosa, virginica, versicolor).

[5] The Kernel Sammon Map has been incorporated into the 'tooldiag' pattern recognition toolbox, written in C, and can be obtained at http://sites.google.com/site/tooldiag.

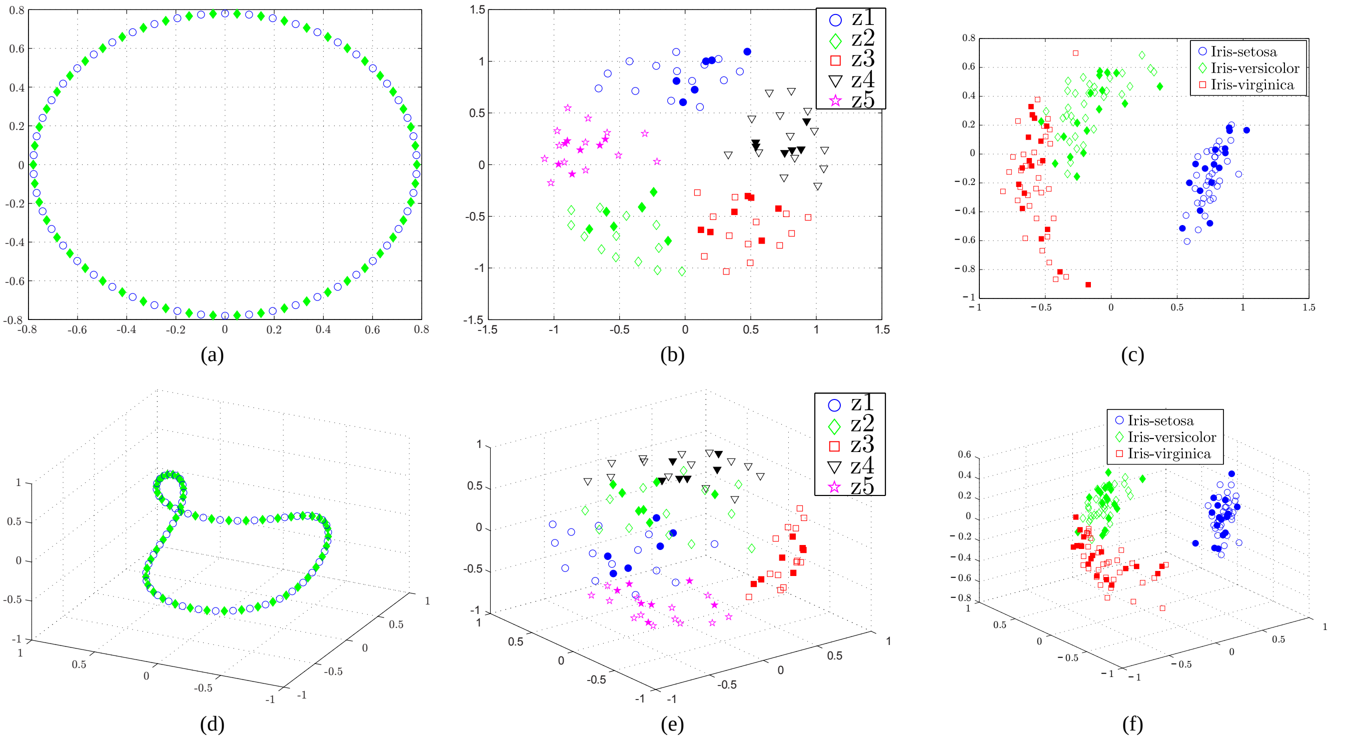


Fig. 2. Kernel Sammon mapping and linear interpolation test. First row is for 2-D projection and second row for 3-D projections. (a) and (d) for Circular data, (b) and (e) for simplex set and (c) and (f) for Iris. The solid marks represent the interpolated test patterns and the hollow marks the training set.

A. *Stress function evaluation*

The quality criterion of the Sammon map is the stress function (3). In order to measure the approximation error of a new test pattern $\mathbf{x}$ with respect to the n training patterns $\mathbf{x}_i$, we define its stress

$$s(\mathbf{x}) = \frac{1}{\sum_{i=1}^{n} D_{i,\mathbf{x}}} \sum_{i=1}^{n} \frac{(D_{i,\mathbf{x}} - d_{i,\mathbf{x}})^2}{D_{i,\mathbf{x}}}, \tag{24}$$

where the distance $D_{i,\mathbf{x}}$ in kernel mapped space respectively in the Sammon mapped pattern space $d_{i,\mathbf{x}}$ between the new pattern $\mathbf{x}$ and the i-th training pattern $\mathbf{x}_i$ are

$$\begin{aligned} D_{i,\mathbf{x}} &= \left[k(\mathbf{x}_i,\mathbf{x}_i) - 2k(\mathbf{x}_i,\mathbf{x}) + k(\mathbf{x},\mathbf{x})\right]^{1/2} \\ d_{i,\mathbf{x}} &= ||\mathbf{y}(\boldsymbol{\phi}(\mathbf{x}_i)) - \mathbf{y}(\boldsymbol{\phi}(\mathbf{x}))||. \end{aligned} \tag{25}$$

The stress for a set $\mathcal{T}$ of m test patterns $\mathbf{x}_p$, $p = 1, \ldots, m$, assuming a uniform probability of observing a test pattern, can be defined as the expected value $\mathbb{E}[s(\mathbf{x})]$ of the stress (24), estimated as the mean over these m test patterns as

$$s(\mathcal{T}) = \frac{1}{m} \sum_{p=1}^{m} s(\mathbf{x}_p). \tag{26}$$

First we compare the original Sammon map and our linear approximation (17), considering the mean stress (26) of m test patterns. We expect the Sammon map to deliver the lowest stress and the linear approximations to perform worse.

For the circular data we chose a set of 50 patterns as the training set and the 50 immediate neighbors of each training pattern as the test set. For the simplex data we do a random 66/33 training/test split and for the iris data the rate is 100/50. We used a RBF kernel with $\sigma = 1$ for the simplex set and $\sigma = \sqrt{5}$ for the circular and Iris data. The results are presented in fig. 2. The solid marks are our linear interpolation method (17) for the test set, while the hollow marks are the mapped training set using the kernel Sammon map (8). In the Iris set the blue circles represent Iris-setosa, red squares Iris-virginica and green diamonds mark Iris-versicolor.

TABLE I

STRESS FUNCTION EVALUATION FOR THE MAPPING OF TRAINING PATTERNS, TEST PATTERNS USING THE SAMMON MAPPING AND TEST PATTERNS USING THE LINEAR INTERPOLATION OF (17). THE KERNEL IS RBF WITH $\sigma = 1$ FOR THE SIMPLEX SET AND $\sigma = \sqrt{5}$ FOR THE CIRCULAR AND IRIS DATA.

Mapping	Training Error	Error-free	Our Method
IRIS 2-D	1.67×10^{-2}	2.08×10^{-2}	2.18×10^{-2}
IRIS 3-D	4.13×10^{-3}	6.01×10^{-3}	7.81×10^{-3}
GAUSSIAN 2-D	1.21×10^{-1}	1.30×10^{-1}	1.32×10^{-1}
GAUSSIAN 3-D	5.97×10^{-2}	6.78×10^{-2}	7.07×10^{-2}
CIRCULAR 2-D	2.06×10^{-2}	2.06×10^{-2}	2.06×10^{-2}
CIRCULAR 3-D	5.78×10^{-3}	5.78×10^{-3}	5.78×10^{-3}

The numerical results for the stress function evaluation are shown in Table I. We calculated the stress function (26) for the test sets for our linear interpolation method and for error-free mapping of new test patterns by the kernel Sammon mapping. In our tests, as expected, the error-free Sammon mapping had the lowest stress for the Iris and Gaussian data. However, for the Circular data the linear approximation delivered an identical stress value compared to the orthodox Sammon mapping. This suggests that the Circular data probably has a linear nature in the RBF kernel mapped space, permitting to express it as a linear transformation, but the theoretical proof is outside the scope of this paper.

B. Sammon map vs. PCA

In his original paper [11], Sammon compared his new technique to the visualization of high-dimensional patterns in Euclidean space from which the first two or three Principal Components were extracted [3]. In analogy to that juxtaposition, we compare the Kernel Sammon map (KSM) of this work to Kernel PCA (KPCA) [14]. Let

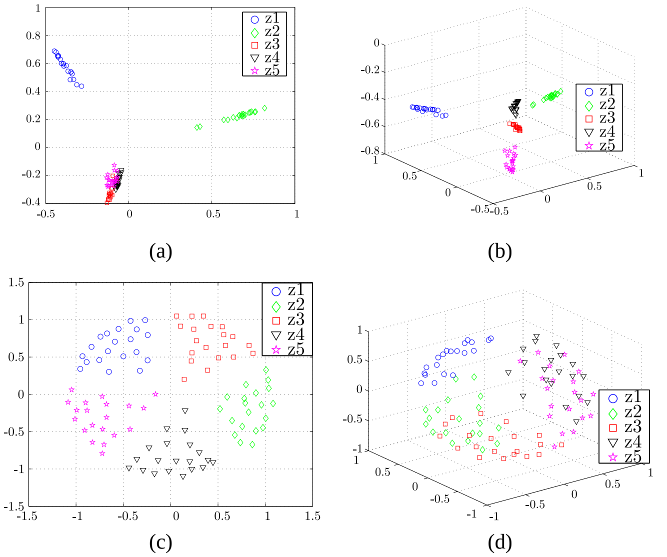
$$\mathsf{K_c} = \mathsf{K} - \mathsf{1K} - \mathsf{K1} + \mathsf{1K1} = \mathsf{B_c}\Omega_\mathsf{c}\mathsf{B_c}^\mathsf{T} \tag{27}$$

be the spectral decomposition of the centered version $\mathsf{K_c}$ of the Kernel matrix K of (19), where $\mathsf{1}$ is a matrix with the value $1/n$ at each position and $\mathsf{B_c}$ and Ω_c have the same meaning as in (20). The Kernel PCA is defined [10] as the function

$$\mathbf{y}(\boldsymbol{\phi}(\mathbf{x})) = \Omega_\mathsf{c}{}^{-1/2}\mathsf{B_c}{}^\mathsf{T}(\mathbf{k_x} - \mathbf{k}_{\boldsymbol{\mu}}). \tag{28}$$

The matrix $\Omega_\mathsf{c}{}^{-1/2}$ is composed of the element-wise inverted square roots of the p nonzero eigenvalues of Ω_c, ignoring the zero-valued eigenvalues, i.e. $\Omega_\mathsf{c}{}^{-1/2} = \mathrm{diag}(1/\sqrt{\lambda_1}, \ldots, 1/\sqrt{\lambda_p}, 0, \ldots, 0)$, $\mathbf{k_x}$ is the empirical kernel map of a pattern $\mathbf{x}$ defined in (14) and $\mathbf{k}_{\boldsymbol{\mu}}$ is the empirical kernel map of the mean vector $\boldsymbol{\mu} = n^{-1}\sum_{i=1}^{n}\mathbf{x}_i$ of all patterns, defined as the n-dimensional vector

$$\mathbf{k}_{\boldsymbol{\mu}} = [\ k(\boldsymbol{\mu}, \mathbf{x}_1) \quad \cdots \quad k(\boldsymbol{\mu}, \mathbf{x}_n)\ ]^\mathsf{T}. \tag{29}$$

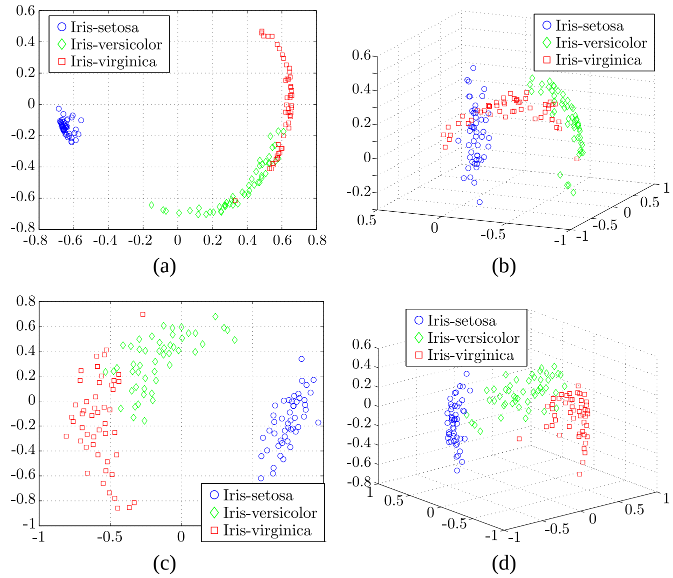


(a) (b)

(c) (d)

Fig. 3. KPCA *vs.* Kernel Sammon Mapping (KSM) for the simplex data. First row is Kernel Principal Component Analysis (KPCA) for (a) 2-D projection and (b) 3-D projection. Second row is KSM for (c) 2-D projection and (d) 3-D projection. RBF kernel with $\sigma = \sqrt{5.0}$ is used.

We used the simplex set and Iris data to compare KSM to KPCA. The results are shown in fig. 3 and fig. 4. For both tests, we used a RBF kernel, with $\sigma = 1$ for the simplex data and $\sigma = \sqrt{5}$ for the Iris set. A similar result as described in [11], where Sammon compares his technique to PCA, is observed for the simplex data. When projected to 2-D, five clusters can be observed (fig. 3 (c)). However, when we compare KSM to KPCA using the two largest eigenvectors, we observe only three clusters (fig. 3 (a)), suggesting at least for this example that KSM provides a more homogeneous and consistent agglomeration. For the 3-D projection, KPCA gives a good clustering behaviour, the KSM however does reflect the circular nature of the Radial Basis kernel much better, since the mapped patterns are basically mapped onto a sphere surface, suggesting that the patterns in the hidden space $\mathcal{H}$ lie on the surface of a hypersphere.



(a) (b)

(c) (d)

Fig. 4. KPCA *vs.* KSM for iris data. First row is KPCA for (a) 2-D projection and (b) 3-D projection. Second row is KSM for (c) 2-D projection and (d) 3-D projection.

C. Visualization of intrinsic kernel parameter variation

The aim of the next experiment is the analysis of the variation of the intrinsic parameter of a kernel, for instance the spread σ of the Radial Basis Kernel. This assists for instance with the evaluation of the behavior of class separability in a classification task. One could imagine a Support Vector Machine as the classifier and the need to study the influence of a kernel parameter on the generation of the separating hyperplane. With the help of KSM this is possible in the usually hidden feature space.

Since we would like to visualize the effect of the kernel parameters, we used an artificial data set (fig. 5), similar to the circular data set used in the previous experiments. It is composed of 150 points distributed in circles of radii 6, 5, 4, 3, 2 and 1 units plus one pattern at the center of all circles. Each circle has 25 points, evenly spaced, in a 3-dimensional space. Different markers are used to provide a better distinguishability of the mapped patterns. The red squares markers represent the circle with the biggest radius, the green diamonds the circle with the smallest radius and center, and the intermediate circles are represented by the blue circles marker.

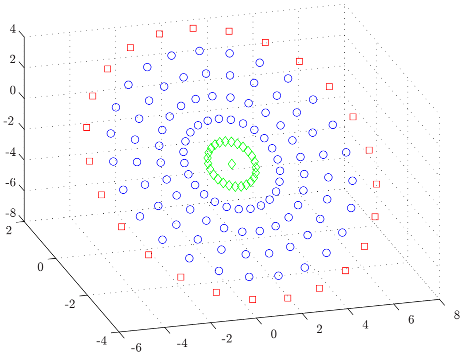


Fig. 5. Data set used for kernel parameter variation tests.

The results for different spreads σ using a RBF kernel are shown in fig. 6. We also used the polynomial kernel, $k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + c)^p$ for which the results are shown in fig. 7 for the polynomial degree $p = 2$ and in fig. 8 for the polynomial degree $p = 3$.

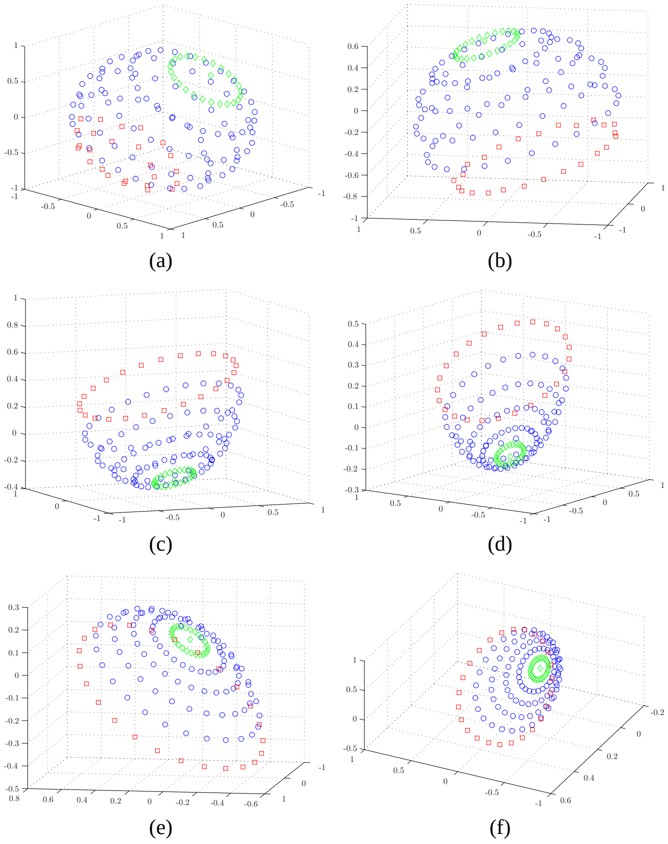


Fig. 6. Parameter variation for the RBF kernel. (a) $\sigma = \sqrt{2.5}$, (b) $\sigma = \sqrt{10}$, (c) $\sigma = 5$, (d) $\sigma = \sqrt{50}$, (e) $\sigma = \sqrt{75}$, (f) $\sigma = \sqrt{87.5}$.

We conclude our experiments with the visualization of three standard UCI Machine Learning data sets [4], “wine”, “WDBC” and “pendigits”, comparing conventional and kernel enhanced Sammon mapping, see fig. 9, fig. 10 and fig. 11.

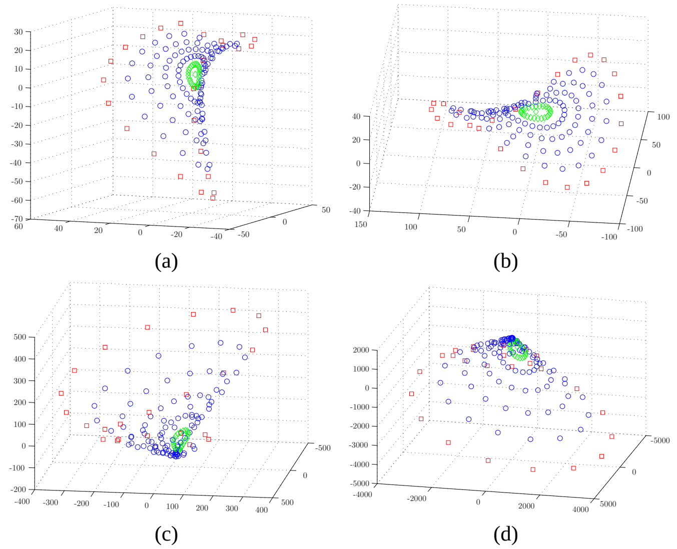


Fig. 7. Parameter variation for the polynomial kernel of degree $p = 2$. (a) $\gamma = 1$, $c = 1$ and $p = 2$, (b) $\gamma = 1$, $c = 10$ and $p = 2$, (c) $\gamma = 10$, $c = 10$ and $p = 2$, (d) $\gamma = 100$, $c = 10$ and $p = 2$.

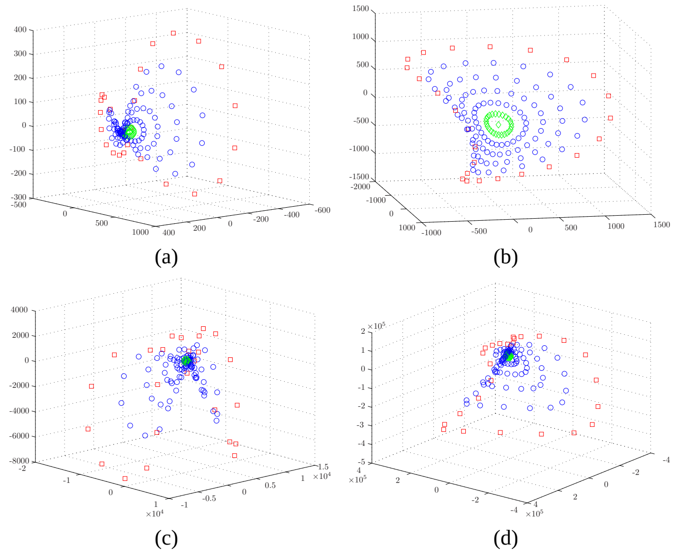
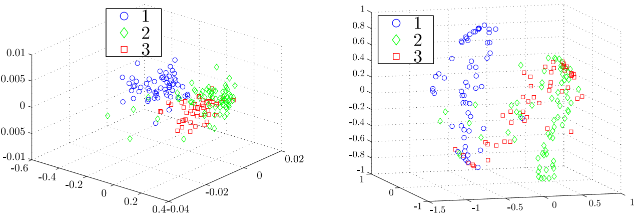


Fig. 8. Parameter variation for the polynomial kernel of degree $p = 3$. (a) $\gamma = 1$, $c = 1$ and $p = 3$, (b) $\gamma = 1$, $c = 10$ and $p = 3$, (c) $\gamma = 10$, $c = 10$ and $p = 3$, (d) $\gamma = 100$, $c = 10$ and $p = 3$.

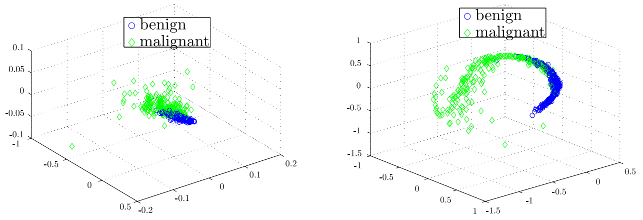
VII. Conclusion and Future Work

We have extended the classical technique of nonlinear high dimensional pattern mapping for visualization proposed by Sammon in the sense that an intermediate nonlinear mapping into Hilbert space is introduced where distances can be measured by virtue of a kernel. We incorporate the distances among the patterns in the intermediate space directly into the stress function proposed by Sammon which provides a straightforward extension of the original concept with a minimal impact onto the gradient descent based learning of the Sammon map. We furthermore propose a linear interpolation of new patterns based on the combination of already mapped



(a) (b)

Fig. 9. Wine data. Three classes ($nc = 3$) with $n = 178$ samples and dimension $d = 30$. (a) Original Sammon map, (b) KSM using RBF kernel with $\sigma = \sqrt{0.0025}$



(a) (b)

Fig. 10. WDBC, $nc = 2, n = 569, d = 30$. (a) Original Sammon map, (b) KSM using RBF kernel with $\sigma = \sqrt{0.025}$

training patterns. The benefits of our technique are obvious. It is now possible to visualize more faithfully the kernel related mapping, compared to Kernel PCA, for instance. Future work will study different kernels and potential applications.

Acknowledgments

The authors would like to thank the CNPq (Brazil) for the financial support given to Mr. Fernando Inaba.

References

[1] M. Aizerman, E. Braverman, and L. Rozonoer. Extrapolative problems in automatic control and the method of potential functions. *Am. Math. Soc. Transl.*, 87:281–303, 1970.

[2] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*,, 25:821–837, 1964.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, New York, 2nd edition, 2001.

[4] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[5] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36:1171–1220, 2008.

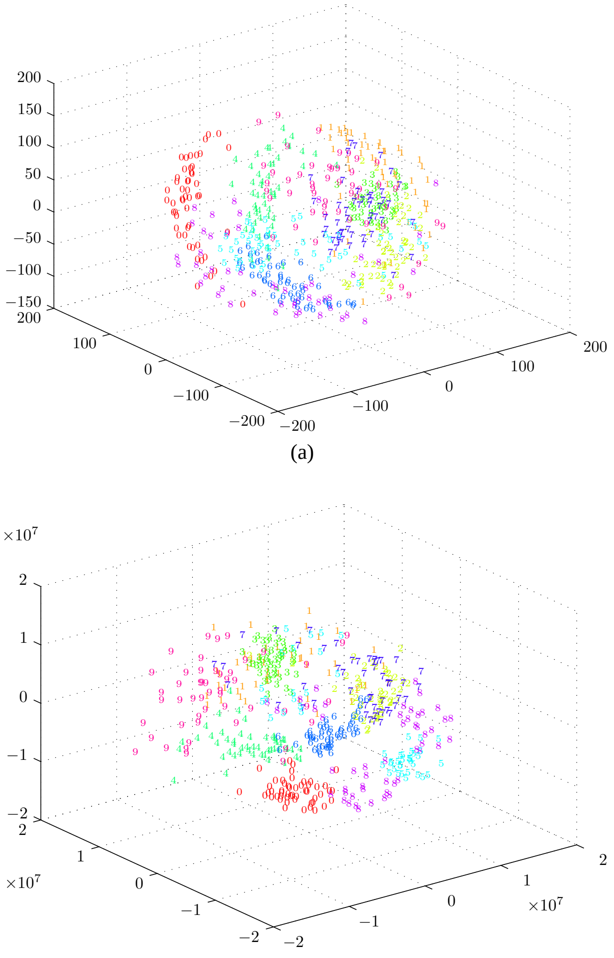
[6] Mingbo Ma, R. Gonet, RuiZhi Yu, and G.C. Anagnostopoulos. Metric representations of data via the kernel-based sammon mapping. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–7, July 2010.

[7] S Mika, G Rätsch, J Weston, B Schölkopf, and K.-R Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[8] Fernando V. Paulovich, Claudio T. Silva, and Luis G. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, 16:1281–1290, November 2010.

[9] Thomas W. Rauber and Karsten Berns. Kernel multilayer perceptron. In Thomas Lewiner and Ricardo Torres, editors, *XXIV Sibgrapi Conference on Graphics, Patterns and Images*, pages 1–7. IEEE Computer Society's Conference Publishing Services (CPS), 2011.

[10] A Ruiz and P.E López-de Teruel. Nonlinear kernel-based statistical pattern analysis. *IEEE Trans. on Neural Networks*, 12(1):16–32, 2001.



(a)

(b)

Fig. 11. Handwritten digits, $nc = 10, n = 5620, d = 64$. (a) Original Sammon map, (b) KSM using polynomial kernel with $\gamma = 1$, $c = 1$ and $p = 3$, only 50 samples shown of each class to avoid overloading of the graph. Compare, for instance, the classes "0" and "3", where less dispersion can be observed, if the polynomial kernel is used.

[11] J. W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[12] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.

[13] B. Schölkopf, S. Mika, C. J.C Burges, P. Knirsch, K.–R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in Kernel-Based methods. *IEEE Trans. on Neural Networks*, 10(5):1000–1017, 1999.

[14] B. Schölkopf, A. Smola, and K. -R Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. Technical Report No. 44, 1996, Max Planck Institut fr biologische Kybernetik, Tübingen.

[15] B Schölkopf and A. J Smola. *Learning with Kernels*. MIT Press, 2002.

[16] A. J Smola and B Schölkopf. A tutorial on support vector regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998. To appear in Statistics and Computing, 2001.

[17] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality Reduction: A Comparative Review. Unpublished, published online, 2007.

[18] V Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[19] Kai Yu, Liang Ji, and Xuegong Zhang. Kernel nearest-neighbor algorithm. *Neural Processing Letters*, 15:147–156, 2002. 10.1023/A:1015244902967.