Metric Representations of Data via the Kernel-based Sammon Mapping

Mingbo Ma, Ryan Gonet, RuiZhi Yu,
and Georgios C. Anagnostopoulos, *Member, IEEE*

***Abstract*—In this paper we present a novel generalization of Sammon's Mapping (SM), which is a popular, metric multi-dimensional scaling technique used in data analysis and visualization. The new approach, namely the Kernel-based Sammon Mapping (KSM), yields the classic SM and other much related techniques as special cases. Apart from being able to approximate distance-preserving projections, it can also learn to metrically represent arbitrarily-defined dissimilarities or similarities between samples. Moreover, it can handle equally well numeric, categorical or mixed-type data. It is able to accomplish all this by modeling its projections as linear combinations of appropriate kernel functions. We report experimental results, which showcase KSM's capabilities in visually representing several meaningful relationships between samples of selected datasets.**

I. Introduction

Exploratory data analysis often relies on visualizing high-dimensional samples to gain understanding about their generation process, their nature and the inter-sample relationships. In a 1938 seminal paper, Young and Householder [1] developed *Multidimensional Scaling (MDS)* as a distance-preserving visualization method for high-dimensional datasets. Since then, a large variety of MDS-based techniques has appeared in the literature. In specific, the family of MDS approaches can be grouped into two major sub-categories: *metric* and *non-metric*. Metric MDS attempts to preserve distances or similarities, while non-metric only preserves their rank order. This paper focuses on a particular version of metric MDS referred to as *Sammon's Mapping (SM)* [2].

SM's popularity stems mainly from its simplicity, elegance and, of course, the very intuitive nature of its outcome, i.e. a visual depiction, where the distances between projected points reflect magnitudes of dissimilarity (usually, distances) between the original samples. In this sense, SM acts as a non-linear isometry between the original high-dimensional space to the low-dimensional (typically, 2 or 3 dimensions) embedding space. While it is capable of handling data sampled from non-linear manifolds, the occasional use of alternative types of dissimilarities may further aid SM in unfolding highly-curved manifolds. As an example, we mention the work of Lee and Verleysen [3], which uses SM with graph distances as an approximation to geodesic (on manifold) distances between samples.

Nevertheless, an important drawback of SM is that it lacks extrapolative and interpolative abilities and is, therefore, incapable of extending its non-linear projection to samples outside the set that was used to design the mapping. In order to overcome this major limitation, several solutions have been proposed, especially ones that attempt to learn and approximate the mapping using the design set and its image via the SM as examples for fitting a projection model. Three contributions that are worth mentioning at this point are SAMMAN [4] and the work of deRidder and Duin [5], both of which utilize a Multi-layer Perceptron to learn the embedding map, as well as the work of Webb [6], which employs a Radial Basis Function (RBF) Neural Network in strict interpolation mode for the same purpose.

In this work, we introduce a generalization of SM, in which a bank of linear kernel machines assumes the role of the projection model responsible of learning the associated SM. We will be referring to this generalization as *Kernel-based Sammon Mapping (KSM)*. First of all, we show that the new approach subsumes the classic SM, as well as the RBF-based approach as special cases. In order to learn the desired non-linear projection, the model allows for the adaptation of the linear weights and, if deemed necessary, of any kernel function parameters as well. As a matter of fact, we discuss a possible option for fitting KSM models efficiently. Additionally, the nature and quality of the approximation can be directly influenced and, possibly, controlled by the choice of the Mercer kernel utilized. This last feature can be highly desirable in the context of exploratory data analysis. Yet another important advantage is the fact that it can handle any type of data, for which a suitable inner-product kernel has been defined. This opens up the possibility of KSM to be used for the visualization of high-dimensional data with categorical or mixed-type attributes, sequenced/ordered data (like time series and micro-array data) and so forth. Finally, let us mention that the concept of utilizing linear kernel machines as a basis for learning the distance-preserving project, which we present here, can be readily applied to and extend the applicability of methods that are related to SM, such as Curvilinear Component Analysis (CCA) [7].

This paper is organized in the following manner. Section II provides some brief, rudimentary background related to

Mingbo Ma is with the Electrical & Computer Engineering Department, Florida Institute of Technology, Melbourne, Florida , US (email: mma2008@my.fit.edu).

Ryan Gonet is with the Computer Science Department, University of South Florida,, Tampa, Florida , US (email: rgonet@mail.usf.edu).

RuiZhi Yu is with the Electrical Engineering Department, Princeton University, Princeton, New Jersey, US (email: ruizhiyu@princeton.edu).

Georgios C. Anagnostopoulos is with the Electrical & Computer Engineering Department, Florida Institute of Technology, Melbourne, Florida , US (phone: +1 321 6747125; email: georgio@fit.edu).

the Sammon Mapping. In section III, we describe the details of the proposed projection model and we discuss methods to train it. Among these methods, we describe an efficient algorithm for adapting the linear weights, namely Iterative Majorization. Section IV contains all our experimental results, whose purpose is to showcase the usefulness and applicability of KSM. Finally, in section V we present our concluding remarks.

II. The Sammon Mapping

Let us assume we have a training set $\{\mathbf{x}_i\}_{i=1,\ldots,N}$ consisting of N samples lying on a manifold that is embedded in some original feature space $\mathbb{F}$. Note that the feature space does not have to be necessarily of Euclidean nature, i.e. the patterns' attributes can also be categorical or of mixed type. We also assume that there is a suitably defined, but otherwise arbitrary, dissimilarity measure $\delta : \mathbf{F} \times \mathbf{F} \to \mathbb{R}^+$. In practice, we use distance metrics to quantify pair-wise dissimilarities. In particular, using geodesic (with respect to the data's manifold) distances are of particular usefulness in removing the manifold's idiosyncrasies (manifold unfolding) from the depiction of the data points' relationships. The Sammon Mapping (SM) seeks to identify a *configuration* of points $\{\mathbf{y}_i\}_{i=1,\ldots,N}$ in $\mathbb{R}^P$, where P is 2 or 3 to allow for visualization, such that the *stress function* below

$$E = \frac{1}{2} \sum_{1 \le i < j \le N} u_{i,j} \left(d_{i,j} - \delta_{i,j}\right)^2 \tag{1}$$

is minimized by adapting the $\mathbf{y}$'s [8]. The stress function E depends on the configuration through the Euclidean distances $d_{i,j} \hat{=} \|\mathbf{y}_i - \mathbf{y}_j\|_2$, where $\mathbf{y}_i$ is the image of $\mathbf{x}_i$ via the SM.

The non-negative weights $u_{i,j}$ can play several roles in the minimization, such as rendering the stress criterion invariant to dissimilarity scaling and so forth. Another interesting role is to determine, which dissimilarities should be ignored, when learning the map, by setting the appropriate weight equal to zero. This has important applications, when using SM to produce an approximately topology-preserving, non-linear projection by setting $u_{i,j} = 0$, if the i^{th} and j^{th} training patterns are not immediate topological neighbors, or by setting it to a non-zero value, if they are not. Furthermore, in exploratory visual data analysis, it allows for removing dissimilarities between user-selected pairs from the stress criterion to eliminate twists and curlings during interactive manifold unfolding.

A minimization algorithm of E adjusts the locations of the training patterns' non-linear projections onto the lower dimensional space $\mathbb{R}^P$, where the pair-wise dissimilarities in the original feature space are portrayed by Euclidean distances with the least amount of distortion, as the embedding dimension P may not be high enough to represent them exactly. The stress function value is precisely measuring this degree of distortion. If the original dissimilarities are Euclidean distances too, the SM can be thought of as an (approximate) isometry, i.e. a distance-preserving mapping. On the other hand, if the dissimilarities are determined solely by neighborhood relationships, SM can e viewed as an approximately topology-preserving mapping.

The $P \times N$ parameters of the mapping can be adjusted by classic unconstrained minimization approaches, such as Conjugate Gradient or quasi-Newton methods, which are economical in storage demands and, simultaneously, offer reasonable convergent speeds. Nevertheless, a fast algorithm for SM learning, that has gained significant popularity, is *iterative majorization (IM)* in the guise of the *SMACOF* algorithm [9]. IM generates a convergent sequence of simpler to minimize subproblems leading to a first-order, globally convergent algorithm, whose speed, implementation simplicity and robustness make it much more attractive than the other alternatives. Some implementation details of IM are discussed in the next section.

Due to its nature, SM can be a very useful visualization tool for explorative data analysis. However, it does not provide a direct mechanism that will allow for sample interpolation: if a new sample becomes available, its image via SM cannot be directly computed. SM learns correspondences between the original samples and their non-linear projections and not the actual, underlying projection map. This very point motivated the works of [4], [5], and [6] to utilize *Multi-Layer Perceptrons* (MLP) and *Radial Basis Function* (RBF) *Neural Networks* for learning the projections.

Two apparent approaches for this issue are (i) first, derive the SM for the given dataset by directly optimizing the output configuration points and then use a regression model to learn, off-line, the mapping of the inputs to the specific outputs obtained from the first step, and (ii) combine these two steps into a single one, where the generative model is trained to minimize the stress criterion; as a byproduct of this optimization, the output configuration can be easily produced.

We would like to point out the advantages of the latter approach over the former one. Overall, the first approach entails two separate optimizations, potentially, with too many parameters (output configuration points as well as model parameters are adapted), when compared with the second one (only model parameters are adjusted). Additionally, the first step of the former approach may produce configuration points, that the subsequent regression model may not be able to satisfactorily match. While adding more degrees of freedom to the regression model may alleviate somewhat this issue, the ensuing over-parameterization of this model may raise other concerns, such as whether the model should be trusted, when it is used for projecting newly available samples. Indeed, the latter (one-step-learning) approach may not guarantee the optimal fidelity in representing dissimilarities (lowest stress function values), but it is definitely capable of yielding trustworthy projection results, as shown in the literature and via our experimental results of Section IV. Thus, we consider it as the only elegant and, at the same time, practical approach to overcoming the limitations of the original SM. As a matter of fact, our proposed Kernel-based Sammon Mapping also falls under this case.

III. Kernel-based Sammon Mapping

Here we introduce the *Kernel-based Sammon Mapping* (*KSM*), which uses linear kernel machines, one for each projection dimension, to model the input/output relationships of traditional Sammon Mappings.

A. The KSM Model

For KSM, the SM projection is modeled as

$$\mathbf{y} = \mathbf{W}^T \mathbf{k}(\mathbf{x}; \boldsymbol{\theta}) \tag{2}$$

where $\mathbf{k}^T(\mathbf{x}; \boldsymbol{\theta}) \hat{=} [k(\mathbf{x}, \mathbf{c}_1; \psi) \dots k(\mathbf{x}, \mathbf{c}_H; \psi)]^T$ is a vector of Mercer (inner-product) kernels evaluated at the input pattern $\mathbf{x} \in \mathbb{F}$. Each kernel is parameterized by their second arguments via the vectors $\mathbf{c}_h \in \mathbb{F}$, which we will be referring to as *prototype vectors*, as well as by a kernel parameter ψ, which, without loss of generality, we'll assume it is scalar and that it has the same value for all kernels in the model. In our notation of the $\mathbf{k}$ vector, we consolidate for clarity all these parameters in a single vector parameter $\boldsymbol{\theta}$. Additionally, $\mathbf{W} \in \mathbb{R}^{H \times P}$ is a weight matrix that projects (not necessarily in an orthogonal manner) the vector of kernel values onto the low-dimensional projection space $\mathbb{R}^P$. In this particular form, the KSM model features $H \times (P + \dim \mathbb{F}) + 1$ free parameters. A typical usage mode of KSM, which we will be referring to as *strict interpolation*, is the case, where all N training patterns $\mathbf{x}_n$ are used as prototype vectors and, therefore, we have $H = N$. In this mode, each input pattern is compared for similarity against every training set sample via the kernel evaluations.

KSM subsumes the classic SM (i.e. the latter is a special case of the former), when employing the *hit-or-miss* kernel shown below

$$k(\mathbf{x}, \mathbf{c}_h; \psi) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{c}_h \\ 0 & \text{if otherwise} \end{cases} \tag{3}$$

and using strict interpolation. In this case, the rows of $\mathbf{W}$ are exactly the $\mathbf{y}$'s, that is, the n^{th} row of $\mathbf{W}$ constitutes the projection of the n^{th} training pattern $\mathbf{x}_n$. It is also obvious, that the particular kernel is less useful with respect to its interpolating capacity. Clearly, KSM also subsumes the RBF approach proposed in [6], as the latter is obtained, when using Gaussian kernels (shown below) in strict interpolation mode

$$k(\mathbf{x}, \mathbf{c}_h; \psi) = e^{-\frac{\|\mathbf{x} - \mathbf{c}_h\|_2^2}{s}} \tag{4}$$

Finally, let us note that using KSM with the hyperbolic tangent kernel shown in Equation (5) would amount to using an MLP with one hidden layer featuring hyperbolic tangent activation functions and an output layer of linear units to learn a projection that represents similarities in the form of inner products in the original feature space with appropriate Euclidean distances in the visualization space.

$$k(\mathbf{x}, \mathbf{c}_h; \psi) = \tanh\left(\mathbf{x}^T \mathbf{c}_h + \psi\right) \tag{5}$$

In its most general form, KSM involves kernels other than the ones already mentioned. Via the use of Mercer kernels, the data in the original feature space are first mapped to another space $\mathbb{G}$ via a mapping $\boldsymbol{\phi} : \mathbb{F} \to \mathbb{G}$ implied by the specific kernel, in the sense, that $\langle \boldsymbol{\phi}(\mathbf{x}; \psi), \boldsymbol{\phi}(\mathbf{x}'; \psi) \rangle_{\mathbb{G}} = k(\mathbf{x}, \mathbf{x}'; \psi)$, where the angle brackets stand for the inner product in $\mathbb{G}$. Then, the similarity of two input patterns is measured by the inner product of their images in $\mathbb{G}$. Therefore, use of different kernels basically amounts to using different similarity measures, which may give rise to different Sammon Mappings. Dissimilarities can also be handled by expressing them as similarities; the Gaussian kernel in Equation (4) is a obvious example of this.

Additionally, the introduction of KSM opens the possibility of handling categorical or mixed-type data, a task that is not possible with the previous MLP and RBF-based methods. If there is a suitably-defined kernel for a particular non-purely-numeric dataset, KSM can be used in a straightforward fashion to represent it in a lower dimensional space.

Finally, let us mention that KSM does not have to be necessarily used in strict interpolation mode (use all training patterns as kernel prototypes). By using a number H of prototypes, that is less than the number N of training patterns, as well as using adjustable prototype vectors, more economical KSM models may be obtained (especially, if $H < N/2$) that are simultaneously quite robust in terms of the quality of test pattern projections.

B. Training the KSM Model

All KSM parameters mentioned in the previous section can be adapted using classic optimization methods, such as Gradient Descent, Conjugate Gradient, quasi-Newton methods, etc. For example, if adjustable prototype vectors are used, the gradient of E (defined in Equation (1)) with respect to the h^{th} prototype vector is given as

$$\frac{\partial E}{\partial \mathbf{c}_h} = \sum_{1 \le i < j \le N} u_{i,j} \left(1 - \frac{\delta_{i,j}}{d_{i,j}}\right) \frac{\partial \Delta \mathbf{k}_{i,j}^T}{\partial \mathbf{c}_h} \mathbf{W} \mathbf{W}^T \Delta \mathbf{k}_{i,j} \tag{6}$$

where we define $\Delta \mathbf{k}_{i,j} \hat{=} \mathbf{k}(\mathbf{x}_i; \boldsymbol{\theta}) - \mathbf{k}(\mathbf{x}_j; \boldsymbol{\theta})$. As a matter of fact, in our experiments with adjustable prototypes, we used a BFGS quasi-Newton method ([10], [11], [12] and [13]) equipped with a line search method described in [14] (Chapter 3, Section 4), which is capable of producing step lengths obeying the Strong Wolfe Conditions.

While it is, indeed, possible to adjust the weights utilizing similar methods, we chose to adapt them using a fixed-point algorithm, that is based on an iterative majorization scheme traditionally used in fitting SMs. The iterative algorithm, whose adaptation to the training of KSM weights is provided below, is reported in the literature as fast and globally convergent.

$$\mathbf{W}_t = \mathbf{A}^{-1} \mathbf{B}(\mathbf{W}_{t-1}) \mathbf{W}_{t-1} \tag{7}$$

where we define the auxiliary matrices $\mathbf{A}$ and $\mathbf{B}$ below. Also, for clarity we explicitly show the dependence of $d_{i,j}$ and, consequently, of $\mathbf{B}$ on the weight matrix $\mathbf{W}$.

$$\mathbf{A} \triangleq \sum_{1\le i<j\le N} u_{i,j}\Delta\mathbf{k}_{i,j}\Delta\mathbf{k}_{i,j}^T \tag{8}$$

$$\mathbf{B}(\mathbf{W}) \triangleq \sum_{1\le i<j\le N, d_{i,j}\neq 0} u_{i,j}\frac{\delta_{i,j}}{d_{i,j}(\mathbf{W})}\Delta\mathbf{k}_{i,j}\Delta\mathbf{k}_{i,j}^T$$

We found that by allowing the weights in $\mathbf{W}$ to get adjusted via Equation (7) more often that the other parameters (e.g. prototypes), the overall algorithm exhibits good stability and speed.

IV. Experimenal Results

In this particular section, we showcase and discuss experimental results obtained by utilizing KSM in tandem with suitable kernels on 4 illustrative datasets, 1 artificial and 3 real, namely the *Teapot*, *MSTAR*, *Mushroom* and *Congressional Voting Records* datasets. For the first two, we show the KSM's potential to depict the structure of the underlying manifold, from which the data are sampled. The last two datasets consist of purely categorical data and are used here to show KSM's capability of successfully handling such type of data.

A. Teapot Dataset

The *Teapot* dataset [15], [16] consists of 100 artificial, color images of the same teapot undergoing a 360° rotation. Each image of the teapot is a 560×420 pixels and represents a sample at 3.6°-increments in angular rotation. Figure 1 depicts three sample frames of the *Teapot* dataset. Due to the small angle increment, consecutive frames differ only slightly.

Fig. 1. Three frames of the *Teapot* dataset.

Since the frames depict the same object, which is just rotated, the obtained frames, once converted to grayscale, should outline a non-intersecting closed curve in the 235200-dimensional space of grayscale images. The manifold related to the teapot's transformation (here, rotation) should be isomorphic to a circle. Only one degree of freedom underlies this periodic (modulo 360°) phenomenon. In the case of the *Teapot* dataset, our goal was to use KSM to visually confirm this fact.

For our experiments, we first converted all images into grayscale and selected every other frame to form a training set. For interpolation (test set), we use the remaining frames. Due to the very nature of the problem, we used geodesic distances (arc-length distances, to be exact) between points, which were derived as follows: for each point the two nearest neighbors were identified using Euclidean distances; then, a fixed value was assigned to the geodesic distances between immediate neighbors. The distance from the i^{th} to the j^{th} training pattern was calculated as the constant amount of geodesic distance between immediate neighbors times the number of training patterns in between those two patterns plus one. We used $H = N/2$ kernels, whose prototypes were chosen randomly among the training set. The kernels were of exponential type with exponents equal to the squared geodesic distances between patterns.

In order to interpolate test patterns, geodesic distances between training and test patterns had to be correctly estimated. Towards this goal, the geodesic distances of each test pattern to its two closest neighbors among the training patterns would be assigned as an appropriate fraction of the geodesic distance between these two neighbors. The specific value of this fraction depended on the test pattern's Euclidean distances to these two closest training patterns.

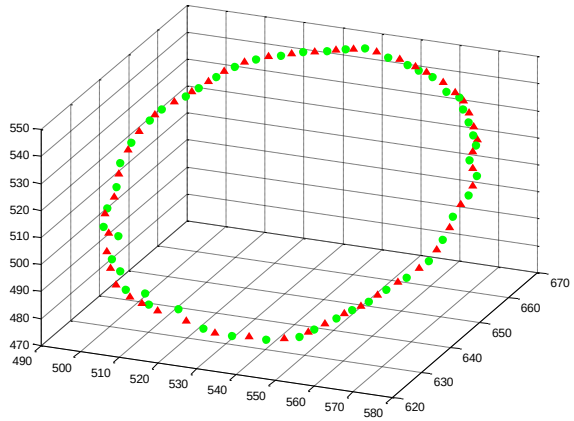

Fig. 2. Projection of the *Teapot* images onto 3 dimensions using KSM. Green solid circles represent training data, while red solid triangles mark interpolated test patterns. As witnessed, KSM was able to place the test patterns on or very close to a very smooth 1-dimensional manifold, the same one passing through the training patterns. On balance, all projected points lie on an almost-flat, closed, non-intersecting curve as expected.

The result we obtained for the *Teapot* dataset is presented in Figure 2. KSM places all points, training and test samples, on a smooth, non-intersecting, closed curve that is almost flat, as expected. In specific, the test images are projected almost in the middle between the projections of similar training patterns.

B. MSTAR Database

In this application of KSM, we consider *Synthetic Aperture RADAR (SAR)* intensity imagery from the well-known *Moving and Stationary Target Acquisition and Recognition (MSTAR)* dataset [17]. The MSTAR dataset consists of a variety of such images for 36 target types. Each image we considered was grayscale and of size 158×158 pixels with

approximate resolution of 1 square foot per pixel. There are 72 poses (different azimuths), one for every 5°, for each target and for a variety of depression angles. Examples of different targets and poses, but of the same depression angle, are shown in Figure 3. Notice that, in general, these images may differ significantly in their intensity distribution. Also, note that the target chip is always in the center and surrounded by ground clutter RADAR returns.

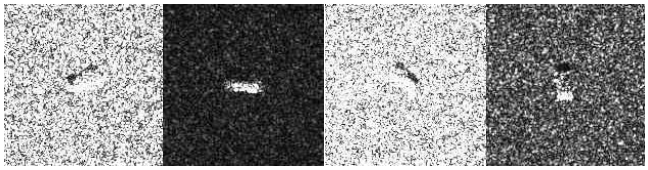

Fig. 3. Four sample SAR images of the *MSTAR* dataset.

For this particular dataset, we showcase the use of KSM to visualize the relationship between different images of the same target at different poses and varying radar illumination. Ideally, as these images depict the same object at different illumination azimuth, they should fall on a simple, non-intersecting closed curve in the 24964-dimensional space of 158×158 grayscale images. As was the case with the *Teapot* dataset, only one degree of freedom underlies this phenomenon. However, these images are not mere rotations of each other; depending on the azimuth, the target may produce a different return, as well as a different shadow region. Furthermore, the resolution is quite low and, judging from the relative intensity of the clutter, the returns are quite noisy. Thus, discovering or confirming the single closed curve hypothesis may seem as a utopia. Nevertheless, we shall see shortly that KSM is able to reflect it.

In specific, we picked SAR images of the *2S1 Gvozdika* (Howitzer class, self-propelled Soviet tank) and removed the ground clutter via a target chip segmentation method described in [18]. Of the 72 available poses for a depression angle of 45° degrees, we used $N = 18$ for training the KSM and another 18 for interpolating. After experimenting with KSM and observing where SAR images, that are topological neighbors in the grayscale space, were projected, we utilized the following $u_{i,j}$ weights to untwist and flatten the manifold

$$u_{i,j} = \begin{cases} 1 & \text{if } |j-i| \mod \frac{N}{3} = 0 \\ 0 & \text{if otherwise} \end{cases} \quad (9)$$

The results showcased in Figure 4 were produced by using $H = N = 18$ kernels in strict interpolation mode using Gaussian kernels with adjustable prototypes. The figure shows that KSM with the aforementioned settings was able to yield a result that captures the nature of the manifold in question, despite the dataset's practical limitations (noisy images, essential dependence of the images on the azimuth). Also, it becomes apparent that the 3^{rd} dimension is superfluous for the mapping. Finally, one can observe that test images fall on or very close to the same manifold as should be expected in a close-to-ideal case.

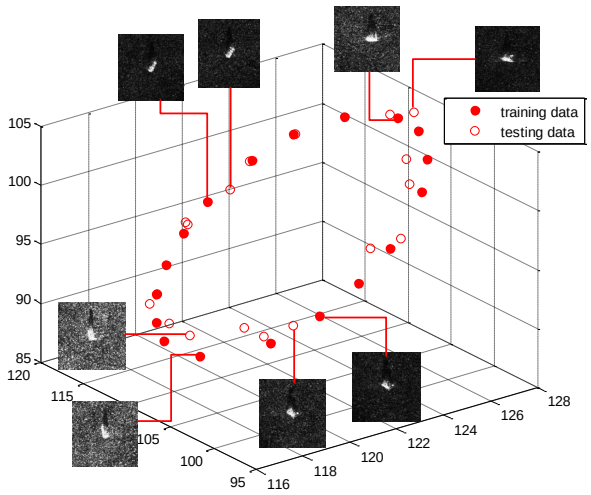

Fig. 4. KSM results for 2S1 tank SAR images need to be represented/visualized as Euclidean distances in a 3-dimensional space. Solid red circles represent the training data, while hollow red circles represent interpolated test patterns. Despite the noisy nature of the images, as well as the dependence of RADAR returns on the particular azimuth with respect to the target's orientation, KSM is able to correctly interpolate previously unseen test images among training images by positioning them very near the same closed curve.

C. Mushroom Database

The *Mushroom* data set [19] consists of 8124 samples, which can be classified as either toxic (poisonous or potentially poisonous) or non-toxic (edible) mushrooms. Each sample in the dataset consists of 22 categorical attributes. A simple measure of dissimilarity between such samples is the number of common attribute values they share, which can be quantified via their pair-wise *Hamming distance*. Here we'll apply KSM to visualize in 2 dimensions some these samples and their dissimilarities, as measured by their Hamming distances. Towards this goal, we employ the *Hamming kernel*, which is defined in [20] and which has been suitably changed to directly use the aforementioned Hamming distances. We picked 40 arbitrary samples, 20 from each class, to fit the KSM model and another 40 (again, 20 from each class) to interpolate. Strict interpolation mode and an *all-ones* adjacency matrix was used, while the Hamming kernel's λ parameter was suitably adjusted. The results are shown in Figure 5 and demonstrate that, in this case, KSM was able to produce an informative non-linear projection of the categorical-natured patterns. Although KSM is not utilized in any discriminatory capacity per se, it illustrates the fact that the difference in value of the provided features seems to be, indeed, powerful enough to discriminate between the two classes of mushrooms. Unlabeled mushrooms, which were not used in the design of the KSM model, eventually project near samples of the same class.

D. US Congressional Voting Records Database

Finally, we apply KSM in an attempt to visually compare the voting behavior of US Democrats and Republicans based

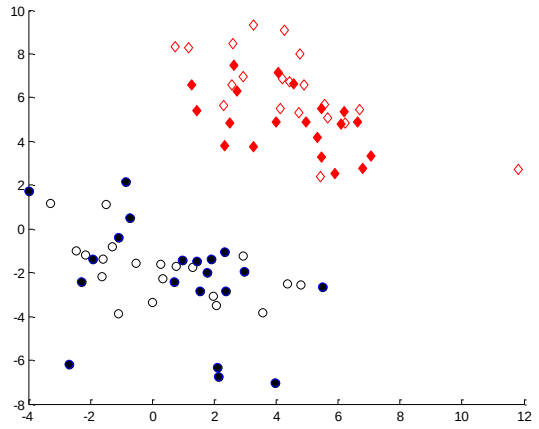

Fig. 5. Results of using KSM with an appropriately modified Hamming Kernel to represented differences in 22 attribute values via Hamming distances between mushroom measurement samples. Diamonds (in red) represent KSM projections of non-toxic mushroom samples, while circles (in blue) of toxic ones. 40 solid color markers represent training data (20 from each kind) and, obviously, 40 hollow markers represent previously unseen (test) patterns (again, 20 from each kind). The figure implies that the features in use may indeed be good discriminators of mushrooms.

on categorical data maintained in the *1984 United States Congressional Voting Records Database* at the UCI Machine Learning Repository [19]. This particular dataset represents the voting records of 435 House of Representatives Members on 16 key issues as identified by a *Certified Quality Auditor (CQA)*. The various issues range from immigration to education and to handicapped infants among other things. Votes were classified as one of three types, "yes" (y) (which include such key words as "voted for," "announced for" and "paired for"), "no" (n) (which include such key words as "voted against," "announced against" and "paired against"), and "unknown" ($?$) (which include "voted present," "voted present to avoid conflict of interest" and "did not vote or otherwise make a position known"). Individual voting records were compared to each other using a variant of the *Tanimoto metric* [21], which, due to the nature of the attribute values, amounted to an increasing function of the pair-wise Hamming distance. Kernels were formed by exponentiating a smaller-than-one scalar to this Tanimoto distance variant.

Figure 6 depicts the results obtained by using $H = N/2$ kernels, whose prototypes were picked from the training set using a greedy combinatorial optimization approach, so to minimize the stress criterion. The weights $\mathbf{W}$, though, were optimized through IM as usual. 10 voting records of each kind (Democratic and Republican) constituted the training set. After convergence, 5 samples of each kind were interpolated. Not surprisingly, the figure illustrates that voters, more or less, casted votes along party lines, since the voting records seemed to be clustered depending on party affiliation. There is, however, a Democrat voter (#7 in the dataset) that is projected closer to Republican voting profiles. More careful post-inspection of the data revealed that the voter was indeed voting very similarly to 2 other Republicans. Instances like this reflect the real world phenomenon that voting behavior does not always conform with party affiliation. Finally, when the test patterns were projected, KSM positioned them along party lines as expected.

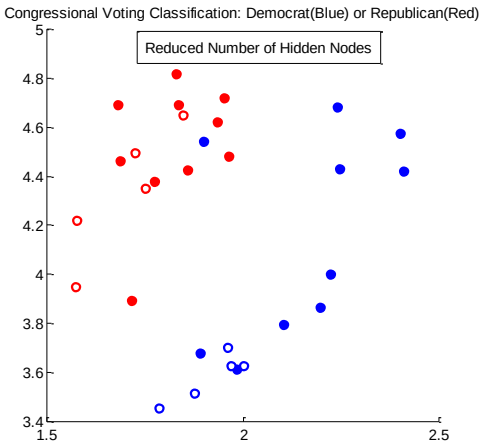

Fig. 6. KSM mapping for a training set of 20 voting records (of 10 Democrats and 10 Republicans), which depicted here via solid circles; red for Republican and blue for Democrat. An additional 5 of each kind are interpolated and depicted with hollow circles. The figure illustrates that members of the House of Representatives predominantly vote along party lines. The only exception is a (conservative?) Democrat outlier.

V. Conclusions

In this paper we presented a novel, kernel-based variant of the classic Sammon Mapping (SM), which we call KSM. This family of models allows for the visual representation of potentially high-dimensional data as projections into a 2- or 3-dimensional space. The projection map strives to preserve as much fidelity in representing inter-sample dissimilarities (typically, distances) or even similarities in the original feature space as Euclidean distances in the visualization space. KSM subsumes the classic SM and other related models as special cases. It also extends SM's original idea and is able to handle data, whose attributes are not necessarily of purely numeric nature. We have also shown a selected set of experimental results to showcase KSM's capabilities, in which KSM emerges as, potentially, a very useful tool for exploratory data analysis. Finally, the generalization of KSM to encompass other SM-related techniques, such as Curvilinear Component Analysis (CCA) [7], is an obvious direction to pursue for future research.

Acknowledgment

The authors acknowledge partial support from the following NSF grants: No. 0647018, No. 0647120, No. 0717680 and No. 0717674. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. The authors would also like to thank Cong Li from the School of Electrical Engineering &

Computer Science at the University of Central Florida for aiding with the *Mushroom* dataset experiments.

REFERENCES

[1] G. Young and A. S. Householder, "Discussion of a set of points in terms of their mutual distances," *Psychometrika*, vol. 3, pp. 19–22, 1938.

[2] J. Sammon, "A nonlinear mapping algorithm for data structure analysis," *IEEE Transactions on Computers*, vol. 18(5), p. 401409, 1969.

[3] J. Lee and M. Verleysen, "Nonlinear dimensionality reduction of data manifolds with essential loops," *Neurocomputing*, vol. 67, p. 2953, 2005.

[4] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, p. 296317, 1995.

[5] D. de Ridder and R. P. W. Duin, "Sammon's mapping using neural networks: A comparison," *Pattern Recognition Letters*, vol. 18, no. 1113, p. 13071316, 1997.

[6] A. R. Webb, "Multidimensional scaling by iterative majorization using radial basis functions," *Pattern Recognition*, vol. 28, no. 5, 1995.

[7] P. Demartines and J. Herault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, January 1997.

[8] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. Boca Raton: Chapman and Hall/CRC, 2001.

[9] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. R. B. et al., Ed. Amsterdam, Netherlands: North-Holland, 1977, pp. 133–145.

[10] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms," *IMA Journal of the Institute of Mathematics and Its Applications*, vol. 6, pp. 76–90, 1970.

[11] R. Fletcher, "A new approach to variable metric algorithms," *Computer Journal*, vol. 13, pp. 317–322, 1970.

[12] D. Goldfarb, "A family of variable metric updates derived by variational means," *Mathematics of Computation*, vol. 24, pp. 23–26, 1970.

[13] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of Computation*, vol. 24, pp. 647–656, 1970.

[14] J. Nocedal and S. J. Wright, *Numerical Optimization*, P. Glynn and S. M. Robinson, Eds. New York, NY: Springer-Verlag, 1999.

[15] J. Tenenbaum, "Mapping a manifold of perceptual observations," *Advances in Neural Information Processing Systems (NIPS 1997)*, vol. 10, pp. 682–688, 1998.

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

[17] "Moving and stationary target acquisition and recognition (mstar) public dataset," Accessed in February 2010. [Online]. Available: https://www.sdms.afrl.af.mil/datasets/mstar/

[18] G. C. Anagnostopoulos, "SVM-based target recognition from synthetic aperture radar images using target region outline descriptors," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e2934 – e2939, 2009.

[19] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[20] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, February 2002.

[21] T. Tanimoto, "An elementary mathematical theory of classification and prediction," Internation Business Machines, Tech. Rep., November 1958.