Accelerated Learning of Generalized Sammon Mappings

Yinjie Huang, Michael Georgiopoulos and Georgios C. Anagnostopoulos

***Abstract*— The Sammon Mapping (SM) has established itself as a valuable tool in dimensionality reduction, manifold learning, exploratory data analysis and, particularly, in data visualization. The SM is capable of projecting high-dimensional data into a low-dimensional space, so that they can be visualized and interpreted. This is accomplished by representing inter-sample dissimilarities in the original space by Euclidean inter-sample distances in the projection space. Recently, Kernel Sammon Mapping (KSM) has been shown to subsume the SM and a few other related extensions to SM. Both of the aforementioned models feature a set of linear weights that are estimated via Iterative Majorization (IM). While IM is significantly faster than other standard gradient-based methods, tackling data sets of larger than moderate sizes becomes a challenging learning task, as IM's convergence significantly slows down with increasing data set cardinality. In this paper we derive two improved training algorithms based on Successive Over-Relaxation (SOR) and Parallel Tangents (PARTAN) acceleration, that, while still being first-order methods, exhibit faster convergence than IM. Both algorithms are relatively easy to understand, straightforward to implement and, performance-wise, are as robust as IM. We also present comparative results that illustrate their computational advantages on a set of benchmark problems.**

I. Introduction

THE Sammon Mapping (SM) is a multi-dimensional scaling technique that was introduced in [1]. Using the available data, SM learns an implicit non-linear projection from the data's original high-dimensional space to, typically, a 2- or 3-dimensional projection space. The location of the SM images of the data are determined, so that the inter-point dissimilarities in the high-dimensional are represented as Euclidean distances in the projection space as faithfully as possible. In the case, where the aforementioned dissimilarities are also Euclidean distances, then the SM learns an approximate isometry from one space to the other. However, it is the fact that it can use almost arbitrarily defined dissimilarities that made SM a very useful and broadly-applicable method in dimensionality reduction, manifold learning, exploratory data analysis and, in particular, data visualization.

The SM has found many applications. For example, it was used in the context of chromosome classification in [2]. It was also applied for visualization of reconstructed phase space trajectories of chaotic systems in [3]. Moreover, it was used for visualizing multi-listener room response equalization in [4], [5]. It was also used in electricity customer classification [6]. Besides, SM was also adopted in city models mapping [7], visualization of web usage patterns [8], classification of protein profiles [9] and visualization of transitions of hepatitis [10]. On balance, as long as the inter-point distances in the projected space lead to a meaningful interpretation, the SM can be an invaluable exploratory/visualization tool.

The original SM lacks the ability to interpolate and extrapolate data, that have not been used for its design. This is because its adjustable parameters are directly the data's projections. Methods were developed to overcome this drawback by assuming that the projections are generated by specific parameterized models. Notable efforts along this path are SAMMAN [11] and the work of deRidder and Duin [12], both of which utilize a Multi-layer Perceptron (MLP) to learn the embedding map. Additionally, generating the projections via a Radial Basis Function (RBF) Neural Network was explored in [13]. More recently, [14] introduced the Kernel Sammon Mapping (KSM), which employs a linear combination of kernel bases to implement the embedding and subsumes the SM and the previously introduced MLP- and RBF-based approaches.

A common element for most of these aforementioned methods to learn the embedding function is a set of linear weights that finally produce the data's images in the projection space. An efficient algorithm based on Iterative Majorization (IM) for estimating these weights has been devised by [15] for the SM and has been extended for what we will refer to in Section II as the Generalized Sammon Mapping (GSM) by [13] and [14]. While IM is significantly faster than gradient descent-based methods, IM is still a linearly convergent method that slows down as the number of weights increases. Note that the number of weights is typically proportional to the number of samples to be projected. This characteristic limits the application of SM and related methods to quite small sample sizes.

The aim of this paper is to explore acceleration methods to the general IM scheme, which will render the SM projection of larger data sets more practical. This can be accomplished by regarding IM as an iterative, differentiable map and applying acceleration procedures specifically designed for fixed-point maps. In this work we explore acceleration of IM based on Successive Over-Relaxation (SOR) and the Parallel Tangents (PARTAN). In specific, we adapt these methods to the original IM scheme and derive two algorithms, namely SOR Accelerated IM (SOR-IM) and PARTAN Accelerated IM (PARTAN-IM), that are relatively easy to understand and to implement. We tested these acceleration schemes on benchmark problems and provide experimental results that

Yinjie Huang is with the Department of EE & CS, University of Central Florida, Orlando, Florida , US (email: darrenhuang22@knights.ucf.edu).

Michael Georgiopoulos is with the Department of EE & CS, University of Central Florida, Orlando, Florida , US (phone: +1 407 8235338; email: michaelg@mail.ucf.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).

show, indeed, that they can achieve significant speedup over the original IM algorithm and that, performance-wise, are equally robust.

The rest of the paper is organized as follows. In Section II we provide some important background regarding the SM, KSM and GSM and describe the IM fixed-point map. In Section III we describe the SOR and PARTAN acceleration methods and show how they can be applied to the IM procedure. Then, in Section IV we provide comparative results for the original IM algorithm and its two accelerated variants on a collection of benchmark problems. Finally, in Section V we end this paper with a summary of our conclusions.

II. The Generalized Sammon Mapping

Let us assume the availability of N samples $\{\boldsymbol{x}_n \in \mathbb{F}\}_{n=1}^N$, where $\mathbb{F}$ is an arbitrary feature space. Let δ_{ij} denote the dissimilarity between the i^{th} and j^{th} samples. It is assumed that these dissimilarities are symmetric in their indices and that $\delta_{ii} = 0 \quad i = 1, \ldots, N$. Note, that any metric on $\mathbb{F}$ can be used as a dissimilarity measure.

Given the aforementioned data set, both the original SM and its extension, the KSM [14], produce a configuration of N points $\left\{\boldsymbol{y}_n \in \mathbb{R}^P\right\}_{n=1}^N$, where $P << \dim \mathbb{F}$ (typically $P = 2, 3$). Each original point $\boldsymbol{x}_n$ is thought to be corresponding to its image $\boldsymbol{y}_n$. (K)SM's goal is to position these projections in such a manner, so that $d_{ij} \hat{=} \|\mathbf{y}_i - \mathbf{y}_j\|_2$ reflects the dissimilarity δ_{ij} as faithfully as possible. The images of the original points are generated as follows

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{k}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{H \times P}$ is an oblique projection matrix that maps the vector $\boldsymbol{k}$ onto the low-dimensional space $\mathbb{R}^P$. Note that, typically, $H \leq N$. Also, $\boldsymbol{k} \in \mathbb{R}^H$ is, in general, a non-linear mapping that may or may not be parameterized. The weights $\boldsymbol{W}$ are estimated so that the stress function, shown below, is minimized.

$$\sigma(\boldsymbol{W}) = \sum_{1 \leq i < j \leq N} u_{ij} \left(d_{ij} - \delta_{ij}\right)^2 \tag{2}$$

where $\boldsymbol{U} \in \mathbb{R}^{N \times N}$ is a hollow symmetric, real-valued matrix with non-negative entries (*i.e.* $u_{ij} = u_{ji} \geq 0$, $u_{ii} = 0$ $i, j = 1, \ldots, N$), that determines the importance of individual discrepancies between dissimilarities and distances in the projection space. Typically, $\boldsymbol{U}$ is an all-ones matrix, save its diagonal, which is assumed to contain entries equal to zero.

We will refer to models that perform the projection according to (1) in order to minimize the stress function in (2) as *Generalized Sammon Mappings* (GSMs). The SM utilizes a mapping $\boldsymbol{k}$ with $H = N$ that is defined as

$$k_h(\boldsymbol{x}) = [\boldsymbol{x} = \boldsymbol{x}_h] \quad h = 1, \ldots, H \tag{3}$$

where $[\cdot]$ denotes the Iversonian bracket; it equals 1, if its enclosed predicate is true, otherwise it equals 0. Because of (3), the SM is unable to produce projections for samples that do not belong to its training set and, thus, is incapable of interpolation and/or extrapolation. On the other hand, the KSM uses a mapping defined as

$$k_h(\boldsymbol{x}) = k\left(\boldsymbol{x}, \boldsymbol{c}_h | \psi\right) \quad h = 1, \ldots, H \tag{4}$$

where $k : \mathbb{F} \times \mathbb{F} \to \mathbb{R}$ is a Mercer (inner-product) kernel (*e.g.* see [16]). In this capacity, the KSM employs an implicit, bounded L_2-norm mapping $\phi_{\boldsymbol{x}} : \mathbb{F} \to \mathbb{H}$ to a (possibly, infinite-dimensional) Hilbert space $\mathbb{H}$, such that $k(\boldsymbol{x}, \boldsymbol{c}) = \langle \phi_{\boldsymbol{x}}, \phi_{\boldsymbol{c}} \rangle_{\mathbb{H}}$, where $\langle \cdot, \cdot \rangle_{\mathbb{H}} : \mathbb{H} \times \mathbb{H} \to \mathbb{R}$ is the inner product that equips $\mathbb{H}$. We assume that the kernels are parameterized by their second arguments via the vectors $\boldsymbol{c}_h \in \mathbb{F}$, which we will be referring to as *prototype vectors*. These vectors can be treated as model parameters or can be appropriately chosen from the training set. Optionally, all kernel functions may have a common scalar parameter $\psi \in \mathbb{R}^+$. The role of these kernels is to measure the similarity between a test sample and the prototype vectors.

For the KSM, the use of appropriate kernels accommodates a variety of data, including data that have categorical or mixed-type attributes, and allow for interpolation/extrapolation in a natural manner. Furthermore, many Mercer kernels are also RBFs. This opens up the possibility of using kernels that solely depend on the dissimilarities between test and prototype patterns, provided that they can be somehow computed.

Nevertheless, in the sequel we'll assume that $\boldsymbol{k}$ is a fixed mapping that only features $\boldsymbol{x}$ as its free argument. As a matter of fact, in our experimental setup in Section IV, when we use the KSM we pick prototype vectors by sub-sampling the training set and fix ψ to a convenient value. Instead, we'll focus on the GSM, whose only model parameter is the weight matrix $\boldsymbol{W}$, regardless of how $\boldsymbol{k}$ is being produced. This way the results we report in this paper apply to all possible SM-based methods and allows us to isolate the process and effects of efficiently estimating $\boldsymbol{W}$, as it will be shown shortly.

For the GSM, estimation of $\boldsymbol{W}$ can be accomplished via the following fixed-point iteration scheme:

$$\boldsymbol{W}_{t+1} = \boldsymbol{M}(\boldsymbol{W}_t) = \boldsymbol{A}^{\dagger} \boldsymbol{B}\left(\boldsymbol{W}_t\right) \boldsymbol{W}_t \tag{5}$$

$\boldsymbol{A}^{\dagger}$ denotes the Moore-Penrose (pseudo)inverse of $\boldsymbol{A}$. The auxiliary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as follows:

$$\boldsymbol{A} \hat{=} \sum_{1 \leq i < j \leq N} u_{ij} \Delta \boldsymbol{k}_{ij} \Delta \boldsymbol{k}_{ij}^T \tag{6}$$

$$\boldsymbol{B}(\boldsymbol{W}) \hat{=} \sum_{1 \leq i < j \leq N} u_{ij} \delta_{ij} d_{ij}^{\dagger}(\boldsymbol{W}) \Delta \boldsymbol{k}_{ij} \Delta \boldsymbol{k}_{ij}^T \tag{7}$$

In (6) and (7), we define $\Delta \boldsymbol{k}_{ij} \hat{=} \boldsymbol{k}(\boldsymbol{x}_i) - \boldsymbol{k}(\boldsymbol{x}_j)$. Also, for a real-valued scalar d we define $d^{\dagger}$ as $\frac{1}{d}$, if $d \neq 0$, and as 0, if $d = 0$. We'll refer to (5) as the *IM iteration*. It is derived in [13] via majorization of the stress function depicted in (2). Application of (5) for the SM produces the SMACOFF

Algorithm 1 Iterative Majorization (IM)

Input: $\boldsymbol{W}_0 \neq \boldsymbol{O},\ t_{max} \geq 1,\ \tau > 0$
Output: $\boldsymbol{W}_{final}$

for $t = 0$ **to** t_{max} **do**
  // Compute gradient matrix
  $\boldsymbol{G}_t \leftarrow [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W}_t)]\,\boldsymbol{W}_t$
  // Check for convergence
  if $\|\text{vec}(\boldsymbol{G}_t))\|_\infty \leq \tau$ **then**
    $\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_t$
    break
  end if
  // Compute IM's search matrix
  $\boldsymbol{D}_{IM} \leftarrow -\boldsymbol{A}^\dagger \boldsymbol{G}_t$
  // Update weights
  $\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t + \boldsymbol{D}_{IM}$
end for
$\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_{t_{max}}$
return $\boldsymbol{W}_{final}$

algorithm presented in [15]. Furthermore, the KSM uses the same rule for updating its weight matrix.

A practical implementation of the estimation process involving IM is provided in Algorithm 1. vec($\boldsymbol{G}$) denotes the vector obtained by orderly concatenating the columns of $\boldsymbol{G}$ into a single-column vector. Note that $\boldsymbol{A}^\dagger$ needs to be computed only once before commencing the iterations. Very often, especially in the case when KSM is used as the model to generate the projections, $\boldsymbol{A}$ is full-rank, and, therefore, the Moore-Penrose inverse simplifies to the ordinary matrix inverse. Moreover, IM always produces descent directions. This is because the gradient matrix $\boldsymbol{G} \hat{=} \frac{\partial \sigma}{\partial \boldsymbol{W}}$ of the stress function σ of (2) is given as

$$\boldsymbol{G} = [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W})]\,\boldsymbol{W} \tag{8}$$

and it can be shown that the IM search direction matrix $\boldsymbol{D}_{IM}$ can be expressed as

$$\boldsymbol{D}_{IM} \hat{=} \boldsymbol{M}(\boldsymbol{W}) - \boldsymbol{W} = -\boldsymbol{A}^\dagger \boldsymbol{G} \tag{9}$$

Since $\boldsymbol{A}$ can be shown to be positive semi-definite, (9) illustrates that $\boldsymbol{D}_{IM}$ corresponds to a descent direction in the weight-space. Another conclusion that can be drawn from (9) is the fact that a fixed point of $\boldsymbol{M}$ (*i.e.* a $\boldsymbol{W}^*$, such that $\boldsymbol{M}(\boldsymbol{W}^*) = \boldsymbol{W}^*$) is also a stationary point of σ.

Overall, IM's popularity stems from the fact that it encompasses several desirable characteristics. Due to its very nature (by design), IM is guaranteed to monotonically converge to a local minimum of the stress function, unless started at a stationary point of σ; curiously enough, such a point is $\boldsymbol{W}^* = \boldsymbol{O}$. Furthermore, there is no specific need to control its step length via a line search method. Hence, it is very straightforward to implement, as one can witness by inspecting Algorithm 1. Finally, it has proven to converge much faster than other algorithms that utilize the gradient of the stress function, such as gradient descent, (non-linear) conjugate gradient and other related methods.

III. Acceleration Methods

Despite its speed advantages, IM can slow down significantly, when the number of weights (*i.e.* the product HP) increases. This typically occurs, when the training set size N increases and, at the same time, high representation fidelity (*i.e.* low σ values) are desired, in which case H needs to be a significant fraction of N. This fact motivates us to seek avenues for devising iterative schemes, that are based on IM, but whose converge speed scales more favorably, as the number of projection weights increases. Since the IM is a fixed-point map, a natural choice is to investigate acceleration methods that are specifically designed for such maps.

An obvious approach is to move further, thus, extrapolating along the IM search direction in the hope that lower values of the stress function are encountered and, longterm, the total number of iterations necessary to reach the vicinity of a local minimum are reduced. At the same time, the computational complexity of the new scheme needs to be controlled, so iterations still remain relatively inexpensive. Such a scheme is offered by a non-linear version of the SOR method, which amounts to employing the following weight update rule

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \alpha \boldsymbol{D}_{IM} \tag{10}$$

with $\alpha \geq 1$ as large as possible (notice that $\alpha = 1$ reduces (10) to an IM update). In other words, the SOR update is given as $\boldsymbol{D}_{SOR} = \alpha \boldsymbol{D}_{IM}$. The pseudo-code for this acceleration technique is provided in Algorithm 2. The **linesearch** procedure, as its name implies, performs a line search along the given direction and will be discussed later in this section.

The second acceleration technique we investigated is the PARTAN method (*e.g.* see [17]). It was originally devised for accelerating gradient descent methods, in specific, to ameliorate the hallmark "zig-zag" sequence pattern of gradient-based updates, thus speeding up convergence. In our context, instead of applying it to gradients of σ, we use PARTAN directly on IM search directions given by (9). We illustrate the inner working of the PARTAN-accelerated IM update via Figure 1.

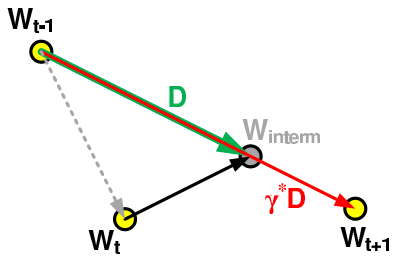


Fig. 1
PARTAN-IM Update.

Algorithm 2 Successive Over-Relaxation acceleration of IM (SOR-IM)

Input: $\boldsymbol{W}_0 \neq \boldsymbol{O},\ t_{max} \geq 1,\ \tau > 0,\ s_{max} \leq 1,$ $\alpha_{init} > 0,\ \eta_{AT} > 1,\ \eta_{BT} \in (0,1),\ c \in (0,1)$
Output: $\boldsymbol{W}_{final}$

for $t = 0$ **to** t_{max} **do**
  // Compute gradient matrix
  $\boldsymbol{G}_t \leftarrow [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W}_t)]\,\boldsymbol{W}_t$
  // Check for convergence
  if $\|\text{vec}(\boldsymbol{G}_t))\|_\infty \leq \tau$ **then**
    $\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_t$
    break
  end if
  // Compute IM's search matrix
  $\boldsymbol{D}_{IM} \leftarrow -\boldsymbol{A}^\dagger \boldsymbol{G}_t$
  // Perform line search and update weights
  $\boldsymbol{W}_{t+1} \leftarrow$ **linesearch**$(\boldsymbol{W}_t, \boldsymbol{D}_{IM}, \boldsymbol{G}_t, \alpha_{init}, s_{max}, \eta_{AT}, \eta_{BT}, c)$
end for
$\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_{t_{max}}$
return $\boldsymbol{W}_{final}$

Assume that $\boldsymbol{W}_{t-1}$ and $\boldsymbol{W}_t$ are generated by PARTAN-IM at iterations $t-1$ and t respectively. Next, our version of PARTAN seeks to find an intermediate point $\boldsymbol{W}_{interm}$ along the IM search direction originating from $\boldsymbol{W}_t$. We choose this update to coincide with an SOR update and, therefore, the step length will, in general, be greater than the one produced by IM. The PARTAN search direction for iteration $t+1$ is given as $\boldsymbol{W}_{interm} - \boldsymbol{W}_{t-1}$, which is equivalent to

$$\boldsymbol{D}_P(t+1) = \boldsymbol{D}_P(t) + \boldsymbol{D}_{SOR} \tag{11}$$

In other words, PARTAN-IM modifies its previous search direction with an intermediate SOR step. Finally, the new iterate $\boldsymbol{W}_{t+1}$ is found by the same line search procedure along PARTAN's search direction. As long as two consecutive iterates of PARTAN-IM monotonically minimize the stress function (*i.e.* $\sigma(\boldsymbol{W}_{t-1}) < \sigma(\boldsymbol{W}_t)$), then it can be seen that the new iterate $\boldsymbol{W}_{t+1}$ is guaranteed to produce an even lower stress function value. In the case, where PARTAN's search direction is an ascent direction, one could revert to an SOR-IM update. However, we noticed in our experiments that this is rarely the case. Pseudo-code for PARTAN-IM is depicted in Algorithm 3.

In both SOR-IM and PARTAN-IM algorithms we used a specialized line search method, which consists of two parts. It starts with an initial step length of α_{init}. If the update along search direction $\boldsymbol{D}$ with the initial step length results in reduction of the stress function, then *ahead-tracking* is engaged to maximize this step-length. For every successful stress function value decrease, the step length is increased by a factor η_{AT} until it is no more possible. On the other hand, if the update $\alpha_{init}\boldsymbol{D}$ leads to an increase in stress function value, then *back-tracking* is employed in order to find a step length that satisfies Armijo's condition [18] of

Algorithm 3 PARTAN acceleration of IM (PARTAN-IM)

Input: $\boldsymbol{W}_0 \neq \boldsymbol{O},\ t_{max} \geq 1,\ \tau > 0,\ s_{max} \leq 1,$ $\alpha_{init} > 0,\ \eta_{AT} > 1,\ \eta_{BT} \in (0,1),\ c \in (0,1)$
Output: $\boldsymbol{W}_{final}$

// Compute initial gradient matrix
$\boldsymbol{G}_0 \leftarrow [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W}_0)]\,\boldsymbol{W}_0$
// Compute first iterate via IM
$\boldsymbol{W}_1 \leftarrow \boldsymbol{A}^\dagger \boldsymbol{B}(\boldsymbol{W}_0)\boldsymbol{W}_0$
for $t = 1$ **to** t_{max} **do**
  // Compute gradient matrix, if it is unavailable
  $\boldsymbol{G}_t \leftarrow [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W}_t)]\,\boldsymbol{W}_t$
  // Check for convergence
  if $\|\text{vec}(\boldsymbol{G}_t))\|_\infty \leq \tau$ **then**
    $\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_t$
    break
  end if
  // Compute IM's search matrix
  $\boldsymbol{D}_{IM} \leftarrow -\boldsymbol{A}^\dagger \boldsymbol{G}_t$
  // Perform line search and compute intermediate matrix
  $\boldsymbol{W}_{interm} \leftarrow$ **linesearch**$(\boldsymbol{W}_t, \boldsymbol{D}_{IM}, \boldsymbol{G}_t, \alpha_{init}, s_{max}, \eta_{AT}, \eta_{BT}, c)$
  // Compute PARTAN-IM's search matrix
  $\boldsymbol{D}_P \leftarrow \boldsymbol{W}_{interm} - \boldsymbol{W}_{t-1}$
  if $\text{trace}\{\boldsymbol{D}_P^T \boldsymbol{G}_{t-1}\} < 0$ **then**
    // The search matrix corresponds to a descent direction; perform a PARTAN-IM step
    $\boldsymbol{W}_{t+1} \leftarrow$ **linesearch**$(\boldsymbol{W}_{t-1}, \boldsymbol{D}_P, \boldsymbol{G}_{t-1}, \alpha_{init}, s_{max}, \eta_{AT}, \eta_{BT}, c)$
  else
    // The search matrix corresponds to an ascent direction; perform an SOR-IM step
    $\boldsymbol{G}_{interm} \leftarrow [\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{W}_{interm})]\,\boldsymbol{W}_{interm}$
    $\boldsymbol{D}_{IM} \leftarrow -\boldsymbol{A}^\dagger \boldsymbol{G}_{interm}$
    $\boldsymbol{W}_{t+1} \leftarrow$ **linesearch**$(\boldsymbol{W}_{interm}, \boldsymbol{D}_{IM}, \boldsymbol{G}_{interm}, \alpha_{init}, s_{max}, \eta_{AT}, \eta_{BT}, c)$
    $\boldsymbol{G}_{t+1} \leftarrow \boldsymbol{G}_{interm}$
  end if
end for
$\boldsymbol{W}_{final} \leftarrow \boldsymbol{W}_{t_{max}}$
return $\boldsymbol{W}_{final}$

sufficient decrease. For each unsuccessful step taken, the step length is decreased by a factor of η_{BT}. According to our experience, back-tracking is rarely employed in practice, but, still, it is necessary, so that the two acceleration techniques remain robust. Pseudo-code for our line search procedure is given in Algorithm 4.

Finally, we should note that $a_{init} = 1$, $\eta_{AT} = 1.95$, $\eta_{BT} = 0.9$ and $c = 0.99$ are good, more or less, empirical values for the line search. We adopted these values to perform all of our experiments, that are presented in the next section, except for the Swiss Roll dataset, for which we obtained better results by using $\eta_{AT} = 2.0$.

Algorithm 4 linesearch() procedure employed by Algorithm 2 and Algorithm 3

Input: $\boldsymbol{W}_0 \neq \boldsymbol{O},\ \boldsymbol{D} \neq \boldsymbol{O},\ \boldsymbol{G}_0 \neq \boldsymbol{O},\ \alpha_{init} > 0,$ $s_{max} \geq 1,\ \eta_{AT} > 1,\ \eta_{BT} \in (0,1),\ c \in (0,1),$ $\boldsymbol{D}$ must correspond to a descent direction

Output: $\boldsymbol{W}_{next}$

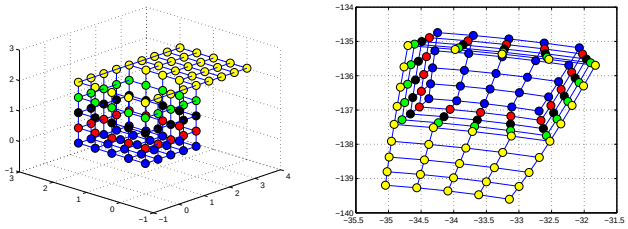
$\alpha \leftarrow \alpha_{init}$
$\sigma_0 \leftarrow \sigma(\boldsymbol{W}_0)$
$\boldsymbol{W}_1 \leftarrow \boldsymbol{W}_0 + \alpha \boldsymbol{D}$
$\sigma_1 \leftarrow \sigma(\boldsymbol{W}_1)$
if $\sigma_1 < \sigma_0$ **then**
  // Perform Ahead-Tracking
  $\alpha \leftarrow \eta_{AT}\alpha$
  for $s = 1$ **to** s_{max} **do**
    $\boldsymbol{W}_{s+1} \leftarrow \boldsymbol{W}_s + \alpha \boldsymbol{D}$
    $\sigma_{s+1} \leftarrow \sigma(\boldsymbol{W}_s)$
    if $\sigma_{s+1} < \sigma_s$ **then**
      $\alpha \leftarrow \eta_{AT}\alpha$
    else
      break
    end if
  end for
else
  // Perform Back-Tracking
  $\Delta\sigma \leftarrow c\ \text{vec}^T(\boldsymbol{G}_0)\,\text{vec}(\boldsymbol{D})$
  $\alpha \leftarrow \eta_{BT}\alpha$
  for $s = 1$ **to** s_{max} **do**
    $\boldsymbol{W}_{s+1} \leftarrow \boldsymbol{W}_s + \alpha \boldsymbol{D}$
    $\sigma_{s+1} \leftarrow \sigma(\boldsymbol{W}_s)$
    // Check sufficient decrease condition
    if $\sigma_{s+1} > \sigma_s + \alpha\Delta\sigma$ **then**
      $\alpha \leftarrow \eta_{AT}\alpha$
    else
      break
    end if
  end for
end if
$\boldsymbol{W}_{next} \leftarrow \boldsymbol{W}_s$
return $\boldsymbol{W}_{next}$

IV. Experimental Results

A. Datasets

In order to evaluate the two acceleration schemes, we compare their convergence speed to the one of the original IM algorithm on five different data sets: 1) Open Box, 2) Teapots, 3) Swiss Roll, 4) Federalist Papers, 5) ORL Faces. For all experiments we used KSM with Gaussian kernels and $H = \frac{N}{2}$ prototype vectors, that were randomly chosen from each data set. Inter-point Euclidean distances in the original space were used in place of dissimilarities. Furthermore, all data sets were projected onto the 2-dimensional plane, *i.e.* $P = 2$. The description of each data set follows, while their characteristics are summarized in Table I.

- *Open Box dataset*. It is an artificially-created data set that consists of points delineating an open box in 3 dimensions as shown in Figure 2a. KSM is used to project its samples onto the plane as demonstrated in Figure 2b. In order to test the scalability of the original IM algorithm and its two accelerated methods, we generated six versions of the data set with 10, 31, 64, 109, 166 and 235 training samples respectively.



(a) (b)

Fig. 2
THE OPEN BOX DATASET AND ITS PROJECTION IN 2D

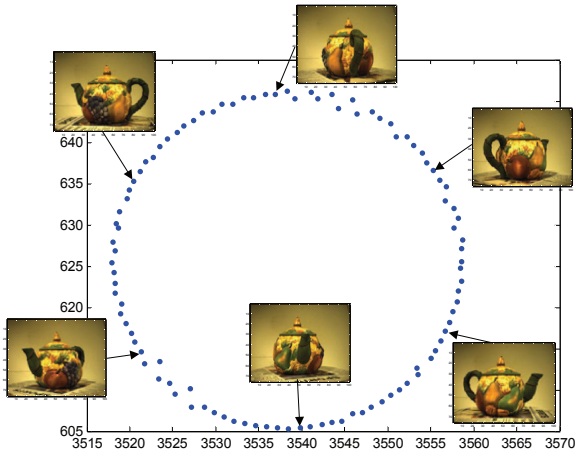
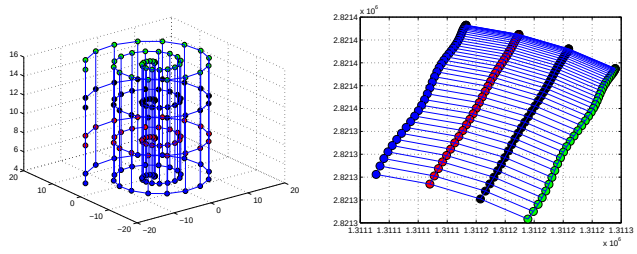


Fig. 3
RESULT OF PROJECTION OF THE TEAPOTS DATASET.

- *Teapots dataset*. This dataset contains 100 different color images of the same artificially-rendered teapot under rotation every $3.6°$ [19], [20]. Each image consists of 560×420 pixels. After conversion to 8-bit grayscale, each image is represented as a $235200-$dimensional vector. Since each image represents a $3.6°$ increment and, thus, only one degree of freedom is involved in this phenomenon, we attempted to project the images onto the plane as shown in Figure 3.
- *Swiss Roll dataset*. This artificially-generated data set contains samples of a rolled sheet in $3D$ space, as depicted in Figure 4a. The sheet's surface is described by the θ-parameterized equations $x_1 = \theta\cos(\theta)$, $x_2 = \theta\sin(\theta)$ and z is arbitrary. Since the intrinsic dimensionality of the manifold at hand is 2, we used KSM to project it onto the plane as shown in Figure 4b.
- *Federalist Papers dataset*. The Federalist Papers were written in 1787 and 1788 and published in many New



(a) (b)

Fig. 4

THE SWISSROLL DATASET AND ITS PROJECTION IN 2D.

York newspapers to persuade the voters to ratify the US Constitution. Of all the collected papers, Alexander Hamilton wrote 56, James Madison wrote 50 and John Jay wrote 5 papers. There are 12 unidentified papers and most of people believe they are from Madison [21]. From this dataset we chose 14 training patterns from Hamilton, 14 from Madison and 12 unidentified papers. Each sample consists of a variety of features that try to capture information that could potentially identify a paper's author, such as the writing style, vocabulary, etc. KSM was used to project these patterns onto to the plane to depict (dis)similarities between papers from different authors. The result of this projection is given in Figure 5.

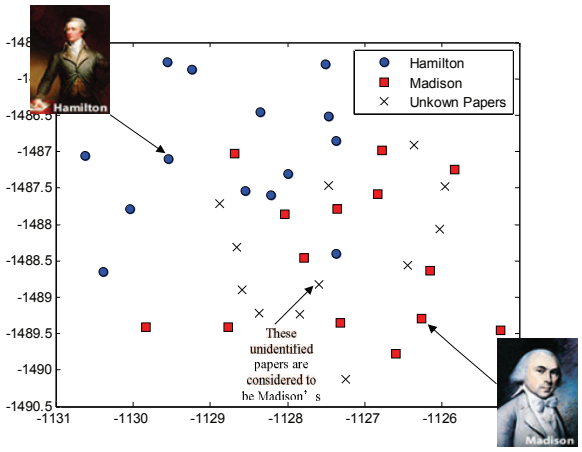


Fig. 5

RESULT OF PROJECTION OF THE FEDERALIST PAPERS DATASET.

- *ORL Faces dataset*. This data set contains images of human faces taken between April 1992 and April 1994 at the Cambridge University Computer Laboratory [22]. There are 40 distinct subjects with each containing 10 different images at various poses, varying lighting conditions, facial expressions and facial details. Each image consists of 119×92 pixels. Again, we used KSM to visualize the relationships between the images of 3 different individuals by representing them as points in the plane. The projection result is illustrated in Figure 6.

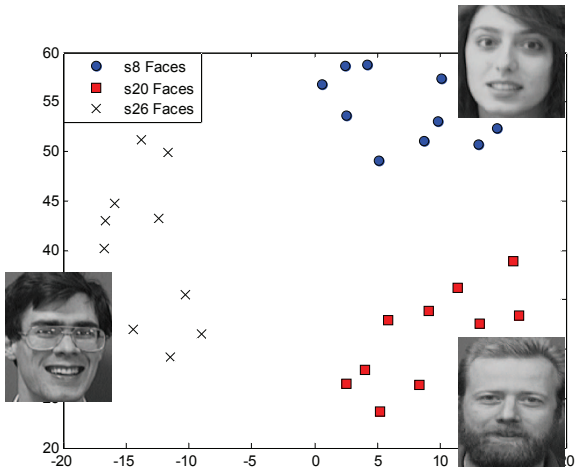


Fig. 6

RESULT OF PROJECTION OF THE ORL FACES DATASET.

TABLE I

DATASET CHARACTERISTICS.

Dataset	Number of training patterns	Dimensionality
Open Box I	10	3
Open Box II	31	3
Open Box III	64	3
Open Box IV	109	3
Open Box V	166	3
Open Box VI	235	3
Teapots	100	235200
Swiss Roll	120	3
Federalist Papers	40	18
ORL Faces	20	10304

In order to compare the three algorithms in terms of convergence speed, we ran each algorithm 100 times for each dataset using each time a different random initial weight matrix. Furthermore, for each run we recorded the number of iterations needed and the time (in seconds) it took for each algorithm to converge to a local minimum. We declared convergence, once the L_∞ norm of the gradient reached or became lower than $\tau = 10^{-4}$. Finally, the values of the spread parameter ψ employed for the Gaussian kernels were 1.2 for Open Box, $50,000$ for Teapots, $12,000$ for Swiss Roll, $8,000$ for Federalist Papers and 800 for ORL Faces.

B. Discussion

Our experimental results for the various versions of Open Box and the remaining datasets are summarized in Table II and Table III respectively. For Open Box I, in terms of median speed, SOR-IM is 57.52% faster than IM, while PARTAN-IM is 74.3% faster than IM. Next, for Open Box II, SOR-IM needs 38.07% less than IM, while PARTAN-IM is 50.01% faster than IM. For Open Box III, the relative difference in medians with respect to IM are 5.35% and 23.64% for SOR-IM and PARTAN-IM respectively. For Open Box IV, the differences are 9.06% between IM and SOR-IM and 34.14% between IM and PARTAN-IM. SOR-IM is 1.72% and PARTAN-IM is 13.49% faster than IM for Open Box

TABLE II

EXPERIMENTAL RESULTS FOR THE OPEN BOX DATASETS.

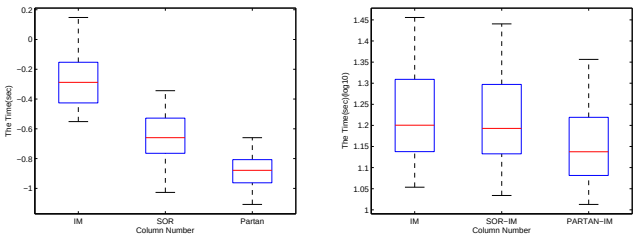
		IM		SOR-IM		PARTAN-IM	
		time(sec)	Iterations	time(sec)	Iterations	time(sec)	Iterations
Open Box I	Max	1.406	168	0.453	60	**0.219**	32
	Up 25	0.703	97.5	0.2965	40	**0.156**	22
	Median	0.5155	71.5	0.219	31	**0.1325**	19
	Low 25	0.375	53	0.172	23	**0.109**	16.5
	Min	0.281	40	0.094	15	**0.078**	14
Open Box II	Max	3.063	573	1.719	149	**0.859**	52
	Up 25	0.945	153.5	0.594	56	**0.461**	33
	Median	0.7815	131.5	0.484	47	**0.39**	29
	Low 25	0.688	122	0.437	41	**0.344**	26.5
	Min	0.438	86	0.312	30	**0.282**	23
Open Box III	Max	9.781	850	8.843	260	**4.203**	82
	Up 25	2.812	244.5	2.64	79.5	**2.1955**	45
	Median	2.4765	217.5	2.344	72	**1.891**	39
	Low 25	1.9295	168	1.969	60.5	**1.7185**	36
	Min	1.438	128	1.453	46	**1.344**	29
Open Box IV	Max	20.6870	695	20.4210	232	**13.2040**	97
	Up 25	15.2180	518.5	14.1250	158.5	**9.6090**	71.5
	Median	12.5160	420	11.3825	130.5	**8.2425**	63.5
	Low 25	10.6565	370.5	9.6175	110.5	**6.7895**	52.5
	Min	7.4370	264	6.8130	82	**5.0000**	40
Open Box V	Max	28.5620	472	27.5780	142	**22.7190**	77
	Up 25	20.3750	335.5	19.8200	103.5	**16.5630**	57.5
	Median	15.8670	258.5	15.5940	84	**13.7265**	48
	Low 25	13.7340	227	13.5705	72.5	**12.0630**	43
	Min	11.3130	186	10.8120	58	**10.2970**	37
Open Box VI	Max	91.9380	720	74.4840	187	**57.3440**	96
	Up 25	56.0240	438	54.7500	137	**41.7500**	71
	Median	50.1645	391.5	48.9840	123	**37.0395**	63
	Low 25	37.9525	298	38.3435	97	**31.5155**	55
	Min	24.0780	189	27.5630	71	**22.6410**	40

TABLE III

EXPERIMENTAL RESULTS ON THE REMAINING DATASETS.

		IM		SOR-IM		PARTAN-IM	
		time(sec)	Iterations	time(sec)	Iterations	time(sec)	Iterations
Teapots	Max	101.609	4270	39.015	475	**31.656**	278
	Up 25	44.43	1844.5	21.0075	249	**16.8125**	148
	Median	30.25	1265.5	16.3515	195	**13.4685**	118.5
	Low 25	20.1015	1001.5	12.4375	149.5	**10.8205**	97.5
	Min	11.828	502	7.469	92	**6.656**	59
Swiss Roll	Max	52.5530	1602	27.0010	187	**19.3100**	104
	Up 25	16.6300	402	15.5495	125.5	**12.2615**	71
	Median	12.8095	321	12.3500	103.5	**9.8580**	62.5
	Low 25	10.1860	249.5	9.8535	84	**8.3155**	55
	Min	4.3780	137	6.3830	59	**5.6740**	35
Federalist Papers	Max	9.406	1482	3.484	226	**2.172**	100
	Up 25	2.3905	439.5	1.906	125	**1.407**	66.5
	Median	1.906	361.5	1.6015	105	**1.1955**	56
	Low 25	1.492	272.5	1.266	82	**1**	48
	Min	0.844	176	0.813	55	**0.703**	36
ORL Faces	Max	3.359	1222	0.75	97	**0.515**	72
	Up 25	1.4295	231	0.391	54	**0.25**	35.5
	Median	0.6485	153	0.297	44	**0.203**	28.5
	Low 25	0.4455	84	0.204	36	**0.1485**	23
	Min	0.094	53	0.109	21	**0.079**	17

V. Finally, for Open Box VI, SOR-IM and PARTAN-IM are 2.35% and 26.16% faster respectively than IM. These results, which are also summarized in Figure 8, indicate that SOR-IM's performance advantage in comparison to IM is probably eroding, as the number of weights increase. A similar conclusion can probably be drawn for PARTAN-IM as well. However, the performance deterioration is much slower for the latter algorithm. Box plots of the execution time distributions for each algorithm on selected Open Box datasets are given in Figure 7.



(a) (b)

Fig. 7
THE BOX PLOTS OF OPEN BOX I AND OPEN BOX VI.

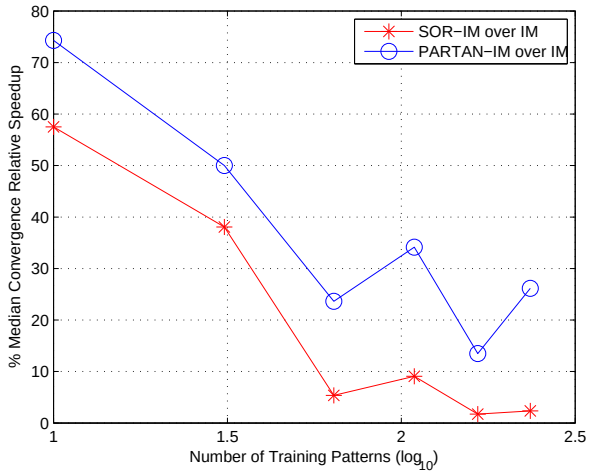
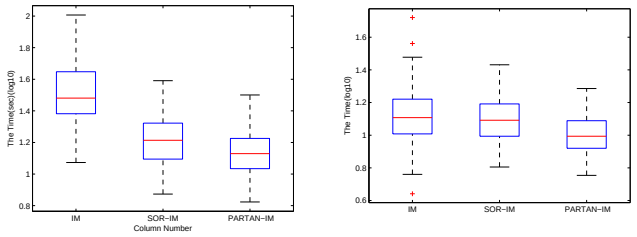


Fig. 8
MEDIAN CONVERGENCE TIME VS. TRAINING SET SIZE FOR THE OPEN BOX DATASETS.

For the Teapots data set, Table III reflects that median SOR-IM speed performance is 45.95% faster than the corresponding IM performance. On the other hand, PARTAN-IM has 55.48% faster median performance than IM. For Swiss Roll, the SOR-IM median slightly outperforms IM's by 3.72%, but PARTAN-IM is still 23.04% faster than IM. Next, for the Federalist Papers dataset, median SOR-IM performance is 15.98% less than the one of IM, while PARTAN-IM is faster by 37.28%. As for ORL Faces, median speedups for SOR-IM and PARTAN-IM are 54.2% and 68.7% respectively faster than IM's median performance. Box plots of the execution time distributions for the Teapots and Swiss Roll data sets are given in Figure 9. Similar Box plots are obtained for the other two remaining data sets as well, but are not shown here.



(a) (b)

Fig. 9
THE BOX PLOTS FOR THE TEAPOTS AND SWISS ROLL DATASETS.

V. CONCLUSIONS

We explored 2 acceleration methods, namely SOR Accelerated IM (SOR-IM) and PARTAN Accelerated IM (PARTAN-IM), to improve the convergence speed of the Iterative Majorization (IM) algorithm, which is used to estimate the projection weights of a Generalized Sammon Mapping (GSM) model. Both methods are relatively easy to understand and to implement. Based on our experiences, we can conclude that both methods indeed accelerate the original algorithm and exhibit robust behavior. We have shown experimentally, that PARTAN-IM always converges faster than SOR-IM, which, in turn, is typically faster than IM. This is despite the relatively increased complexity of these algorithms over IM. However, it seems that their speedup advantage over plain IM seems to decrease as the number of weights increase, a fact that is more or less expected. Nevertheless, they may still retain an advantage, albeit much smaller, even as the size of the dataset to be projected increases.

ACKNOWLEDGMENT

Yinjie Huang acknowledges partial support from NSF grants No. 0647120 and No. 0717680 and partial support from a UCF Graduate College Presidential Fellowship. Moreover, Michael Georgiopoulos acknowledges partial support from NSF grants No. 0525429, No. 0837332 and No. 0717680. Finally, Georgios C. Anagnostopoulos acknowledges partial support from NSF grants No. 0717674 and No. 0647018. 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. Finally, the authors would like to thank the anonymous reviewers of this manuscript for their time and helpful comments.

REFERENCES

[1] J. Sammon, "A nonlinear mapping algorithm for data structure analysis," *IEEE Transactions on Computers*, vol. 18(5), p. 401409, 1969. [Online]. Available: http://dx.doi.org/10.1109/T-C.1969.222678

[2] H. G. Boaz Lerner and M. Aladjem, "On Pattern Classification with Sammon's Nonlinear Mapping: An Experimental Study," *Pattern Recognition*, vol. 31, no. 371-381, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0031-3203(97)00064-2

[3] B. B. Balazs Feil and J. Abonyi, "Visualization of fuzzy clusters by fuzzy Sammon mapping projection: application to the analysis of phase space trajectories," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 11, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00500-006-0111-5

[4] C. Bharitkar, S.; Kyriakakis, "Visualization of multiple listener room response equalization using Sammon map," in *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, vol. 2, 2002, pp. 277 – 280 vol.2. [Online]. Available: http://dx.doi.org/10.1109/ICME.2002.1035575

[5] S. Bharitkar and C. Kyriakakis, "Visualization of Multiple Listener Room Acoustic Equalization With the Sammon map," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 542 –551, February 2007. [Online]. Available: http://dx.doi.org/10.1109/TASL.2006.881683

[6] G. Chicco, R. Napoli, and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 933 – 940, May 2006. [Online]. Available: http://dx.doi.org/10.1109/TPWRS.2006.873122

[7] C. Frueh, R. Sammon, and A. Zakhor, "Automated texture mapping of 3d city models with oblique aerial imagery," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, September 2004, pp. 396 – 403. [Online]. Available: http://dx.doi.org/10.1109/TDPVT.2004.1335266

[8] S. Kannappady, S. P. Mudur, and N. Shiri, "Visualization of Web Usage Patterns," in *Database Engineering and Applications Symposium, 2006. IDEAS '06. 10th International*, December 2006, pp. 220 –227. [Online]. Available: http://dx.doi.org/10.1109/IDEAS.2006.52

[9] G. Karemore, J. Mullick, R. Sujatha, M. Nielsen, and C. Santhosh, "Classification of protein profiles using fuzzy clustering techniques: An application in early diagnosis of oral, cervical and ovarian cancer," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 312010-sept.4 2010, pp. 6361 –6364. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2010.5627292

[10] T. Miyamoto, Y. Fujita, S. Uchimura, Y. Hamamoto, N. Iizuka, and M. Oka, "Visualization of transitions of developing of hepatitis C virus-associated hepatocellular carcinoma," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, December 2008, pp. 1 –4. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2008.4761751

[11] 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. [Online]. Available: http://dx.doi.org/10.1109/72.363467

[12] 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. [Online]. Available: http://dx.doi.org/10.1016/S0167-8655(97)00093-7

[13] A. R. Webb, "Multidimensional Scaling by Iterative Majorization Using Radial Basis Functions," *Pattern Recognition*, vol. 28, no. 5, 1995. [Online]. Available: http://dx.doi.org/10.1016/0031-3203(94)00135-9

[14] M. Ma, R. Gonet, R. Yu, and G. Anagnostopoulos, "Metric representations of data via the Kernel-based Sammon Mapping," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1 –7. [Online]. Available: http://dx.doi.org/10.1109/IJCNN.2010.5596662

[15] 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.

[16] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. Adaptive Computation and Machine Learning, T. Dietterich, Ed. Cambridge, Massachusetts, US: The MIT Press, 2002.

[17] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed., ser. International Series in Operations Research and Management Science, F. S. Hillier, Ed. Springer Science+Business Media, LLC, 2008, vol. 116.

[18] L. Armijo, "Minimization of Functions Having Lipschitz Continuous First-Partial Derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966. [Online]. Available: http://projecteuclid.org/euclid.pjm/1102995080

[19] J. B. Tenenbaum, "Mapping a manifold of perceptual observations," in *Advances in Neural Information Processing Systems 10*. MIT Press, 1998, pp. 682–688.

[20] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, pp. 2319 – 2323, 2000. [Online]. Available: http://dx.doi.org/10.1126/science.290.5500.2319

[21] G. Fung, O. Mangasarian, and J. Jay, "The Disputed Federalist Papers: SVM Feature Selection via Concave Minimization," in *in Proc. 2003 Conf. on Diversity in Computing, ACM*. ACM Press, 2003, pp. 42–46. [Online]. Available: http://dx.doi.org/10.1145/948542.948551

[22] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, December 1994, pp. 138 –142. [Online]. Available: http://dx.doi.org/10.1109/ACV.1994.341300