Visualizing pairwise similarity via semidefinite programming

Amir Globerson
Computer Science and Artificial Intelligence Laboratory
MIT
Cambridge, MA 02139
gamir@csail.mit.edu

Sam Roweis
Department of Computer Science
University of Toronto
Toronto, Canada
roweis@cs.toronto.edu

Abstract

We introduce a novel learning algorithm for binary pairwise similarity measurements on a set of objects. The algorithm delivers an embedding of the objects into a vector representation space that strictly respects the known similarities, in the sense that objects known to be similar are always closer in the embedding than those known to be dissimilar. Subject to this constraint, our method selects the mapping in which the variance of the embedded points is maximized. This has the effect of favoring embeddings with low effective dimensionality. The related optimization problem can be cast as a convex Semidefinite Program (SDP). We also present a parametric version of the problem, which can be used for embedding out of sample points. The parametric version uses kernels to obtain nonlinear maps, and can also be solved using an SDP. We apply the two algorithms to an image embedding problem, where it effectively captures the low dimensional structure corresponding to camera viewing parameters.

1 Learning from Binary Similarity Measurements

Often in data analysis our goal is to learn something about the relationship between entities despite having only limited quantitative measurements. In particular, in certain domains it may not be possible to naturally represent objects (e.g. proteins) in a vector space and thus we may not be able to associate objects with points in a feature space. Furthermore, it may be difficult to obtain real-valued labels or to measure a quantitative numerical "distance" or "dissimilarity" between objects. However, it is often possible to obtain sparse binary similarity measurements on a limited number of pairs of objects which tell us if the pair is known to be related/linked or known not to be (e.g. the presence or absence of functional/physical interactions between proteins).

Insight into such data may be obtained by associating each object with a point in some abstract representation space, which for visualization purposes is often two or three dimensional. This *embedding* should naturally reflect the known relations between the objects. Here we present a learning algorithm for such situations which, given a similarity matrix, delivers an embedding of the objects that strictly respects the known similarities. Namely, *objects known to be similar are always closer in the embedding space than those known to be dissimilar*. If input constraints are sparse or the embedding space has multiple dimensions, many such embeddings may exist; our method selects among them the one in which *the mean distance between dissimilar embedded points is as large as possible, and the mean distance between similar points is as small as possible*, an idea related to the Semidefinite Embedding (SDE) method of Weinberger and Saul (Weinberger & Saul, 2004).[1] This has the effect of *unfolding* the mapping and, interestingly, favoring embeddings with low effective dimensionality.

We formulate both parametric and non-parametric variants of the problem, showing that they both result in convex Semidefinite Programs (SDP) (Boyd & Vandenberghe, 2004). In the non-parametric version, objects may be mapped to arbitrary points in the representation space, whereas the parametric version assumes that objects have associated with them some feature vector and that the mapping is given by a function of this feature vector. Specifically we take this function to be a linear projection in some (possibly infinite dimensional) space where dot products are given by a kernel. The parametric version allows us to generalize the embedding to future unseen objects

[1] This method is also sometimes referred to as *Maximum Variance Unfolding* (Sun et al., 2006).

which are not part of the training procedure and which do not have known similarity relations with other objects, as long as the new objects have an associated feature vector from which we can compute their embedding.

We use our method to embed sets of images taken under different conditions, and show that it successfully captures the low dimensional manifold corresponding to camera position.

2 Pairwise Semidefinite Embedding - Problem Formalization

The input to our method is a set of n objects $\mathcal{X} = \{1, \ldots, n\}$ and a set of similarity/dissimilarity relations $s_{ij} \in \{-1, 0, 1\}$. When $s_{ij} = 1$, the pair (i, j) is considered similar, and when $s_{ij} = -1$ it is dissimilar. If no similarity data is available, we set $s_{ij} = 0$. The desired output is a set of points $\boldsymbol{\phi}_i \in \Re^p$ for $i \in \mathcal{X}$. The points should *respect* the similarity structure given by the pairwise binary relations. As in standard MDS (Cox & Cox, 1984), we will use Euclidean distances in $\Re^p$ to reflect the original similarities, although here the similarities are highly non-metric, taking on values of $\{\pm 1, 0\}$ only. Specifically, we will require that the embedding $\boldsymbol{\phi}_i$ of an object i will always be closer (using distance in $\Re^p$) to the embeddings of points similar to it than to the embeddings of points dissimilar to it. No constraint is placed on the distances between embeddings of points for which $s_{ij} = 0$.

Importantly, our method does not require as input any metric information other than binary similarity relations, and can thus be used with a much wider variety of inputs than traditional multidimensional scaling (Cox & Cox, 1984) or more advanced local-distance preservation methods (Tenenbaum et al., 2000; Weinberger & Saul, 2004).

We now formalize the constraints on the embedding $\boldsymbol{\phi}$. If all objects similar to i are embedded closer to it than dissimilar ones, there should exist a set of radii $b_i \geq 0$ such that $\boldsymbol{\phi}_j$ for all j similar to i lie within a sphere of radius b_i centered on $\boldsymbol{\phi}_i$, and $\boldsymbol{\phi}_k$ for all k dissimilar to i lie outside this sphere. If we denote distances in the embedding by $d_{ij}^2 = \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2$, we can capture the above geometric constraint (illustrated in Figure 1) with a simple algebraic inequality (Equation 1):

$$s_{ij} d_{ij}^2 \leq s_{ij} b_i \quad \forall ij \tag{1}$$

Note that the radii b_i are not known in advance and will also need to be found by the embedding algorithm.

We shall also find it useful to remove the translation and scaling invariance of the embedding, , by requiring it to be centered at the origin and have a bounded

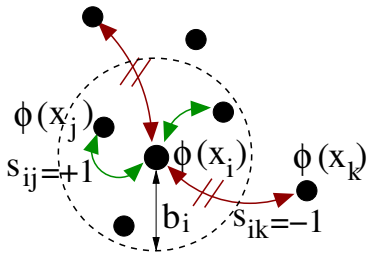


Figure 1: An illustration of the embedding objective. Similar points $(\mathbf{x}_i, \mathbf{x}_j)$ should be more closely mapped than dissimilar ones $(\mathbf{x}_i, \mathbf{x}_k)$.

mean distance from the origin. These two constraints can be represented as:

$$\sum_i \boldsymbol{\phi}_i = 0\,, \qquad \sum_{ij} d_{ij}^2 \leq 1 \tag{2}$$

Although the centering constraint is not strictly necessary, it improves the numerical behavior of some of the algorithms, and simplifies some of the derivations.

To summarize, we are seeking a mapping $\boldsymbol{\phi}$ which satisfies the constraints in Equations 1 and 2. Clearly, many embeddings may satisfy the pairwise constraints (e.g. mapping all points to the origin. Note however, that this solution can be easily avoided by adding a margin requirement such as $b_i \geq 1$ as in (Shalev-Shwartz et al., 2004). We shall see that this will not be needed in our method since we will be explicitly maximizing variance.). To enhance visualization, we shall prefer mappings which minimize distances between similar points, while maximizing distances between dissimilar points. We can achieve this by minimizing the following objective (subject to the constraints in Equations 1 & 2):

$$f(\boldsymbol{\phi}) = \frac{1}{n_S} \sum_{ij:s_{ij}=1} d_{ij}^2 - \frac{1}{n_D} \sum_{ij:s_{ij}=-1} d_{ij}^2 \tag{3}$$

where we define n_S, n_D to be the number of similar and dissimilar pairs respectively. In what follows we shall assume that $n_S = n_D$ so we can write $f(\boldsymbol{\phi}) = \sum_{ij} s_{ij} d_{ij}^2$. This is just for notational convenience, and all our results apply when $n_S \neq n_D$.

We note that the above objective has the effect of *stretching* the embedding, as in SDE (Weinberger & Saul, 2004), and will thus prefer low dimensional solutions (i.e. embeddings with intrinsic dimensionality lower than p) when those are available.

Thus, the complete pairwise embedding problem (PAIREMB) is to minimize $\sum_{ij} s_{ij} d_{ij}^2$ with respect to the n vectors $\boldsymbol{\phi}_i \in \Re^p$ and the radii $b_i \geq 0$, subject to the pairwise constraints $s_{ij} d_{ij}^2 \leq s_{ij} b_i$ and the centering and scaling constraints $\sum_i \boldsymbol{\phi}_i = 0$, $\sum_{ij} d_{ij}^2 \leq 1$.

Unfortunately, this optimization problem is highly non-convex and may have multiple local minima. The main source of non-convexity is the constraint $\boldsymbol{\phi}_i \in \Re^p$. In the next section we suggest an approach for approximating the above optimization by formulating a convex problem using semidefinite programming.

3 The Non-Parametric Pairwise Semidefinite Embedding Method

The close link between Euclidean distances and Positive Semidefinite (PSD) matrices allows us to recast the above problem in a manner simpler to solve, although the solution may be approximate in some cases. Define a PSD Gram matrix G (of rank p) with entries $g_{ij} = \boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j$; then $d_{ij}^2 = g_{ii} + g_{jj} - 2g_{ij}$. Conversely, any PSD matrix G of rank p can be written as $G = \Phi\Phi^T$ where Φ (the square root of G) is a matrix of size $n \times p$. Thus the mapping between PSD matrices of rank p and p dimensional embeddings is one to one. We can therefore recast the PAIREMB optimization above in terms of G instead of $\boldsymbol{\phi}$, replacing the constraint $\boldsymbol{\phi}_i \in \Re^p$ by the constraint that G is of rank p. The other constraints also have simple expressions under this new formulation:

$$\begin{array}{lll} \sum_i \boldsymbol{\phi}_i = 0 & \rightarrow & \sum_{ij} g_{ij} = 0 \\ \sum_{ij} d_{ij}^2 \leq 1 & \rightarrow & \sum_i g_{ii} \leq 1 \end{array} \tag{4}$$

This problem is still non-convex since the set of PSD matrices of rank $p < n$ is not convex. However for $p = n$ this set *is* convex, and thus the rank-relaxed PAIREMB problem becomes tractable, and can be solved using standard semidefinite program (SDP) solvers, since the objective and all the constraints are linear in the elements of G. We call this new optimization "Pairwise Semidefinite Embedding" (PSDE). It is equivalent to PAIREMB when $p = n$ (but not when $p < n$):

PSDE:

$$\begin{array}{lll} \min_{G,b} & \sum_{ij} s_{ij}(g_{ii} + g_{jj} - 2g_{ij}) & \\ s.t. & s_{ij}(g_{ii} + g_{jj} - 2g_{ij}) \leq s_{ij} b_i & \forall ij \\ & \sum_i g_{ii} \leq 1 & \\ & \sum_{ij} g_{ij} = 0 & \\ & G \succeq 0 \qquad b_i \geq 0 & \forall i \end{array}$$

To obtain a p dimensional embedding from G we use a standard spectral decomposition as in (Weinberger & Saul, 2004), and consider only the p leading eigenvalues. The main drawback of the PSDE problem is that it is cast in a dimension n which is usually much higher than the typical target dimension $p = 2, 3$. However, because of the form of the objective function, which unfolds embeddings, this problem often results in solutions with low effective dimensionality (i.e., the effective rank of G is small). Of course, if the optimal G is of rank p, we are guaranteed that it is also a solution to PAIREMB, since PSDE is less constrained than PAIREMB. In all the experiments we performed the effective rank of G was much lower than n.

4 Parametric Pairwise Embeddings and Kernel PSDE

In the previous two sections, we assumed that $\mathcal{X}$ was simply a set of objects, which were not necessarily represented as real vectors. In such cases, the embedding cannot easily be generalized to new objects. However, if the objects $\mathcal{X}$ do have feature vector representations, i.e., $\mathbf{x}_i \in \Re^q$, we may seek an embedding which is an explicit function of these features $\boldsymbol{\phi} : \Re^q \rightarrow \Re^p$.

Perhaps the simplest form of such a mapping is a linear map $\boldsymbol{\phi}(\mathbf{x}) = A\mathbf{x}$. While this would be quite limited in the original feature space, it can be made much more expressive using the well known *kernel trick* (Schölkopf & Smola, 2002), as we show below. The goal of parametric pairwise embedding is to find a matrix A such that the embedding $\boldsymbol{\phi}(\mathbf{x})$ solves the non-parametric PAIREMB problem. The SDP version of the problem is to find a matrix A which solves the PSDE problem.

Since we shall be interested in using kernels in what follows, we augment the PSDE problem by adding a regularization term $\gamma\|A\|^2$ to its objective the (γ is a positive regularization factor). It can be shown that the A which solves this regularized PSDE has the form $A = WX$, where the matrix X has $\mathbf{x}_i$ as rows and W is a matrix of size $p \times n$ (i.e. the "representer theorem" applies).

We can now define a new matrix $Q = W^T W$ such that the entire optimization problem is cast in terms of Q and dot products between vectors $\mathbf{x}_i$. The Gram matrix of the embedded points can be written as

$$g_{ij} = \boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_j) = \mathbf{x}_i^T X^T Q X \mathbf{x}_j \tag{5}$$

Define the "kernel matrix" K to be the matrix whose entries are $K_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$. The elements of the Gram matrix are then given by $g_{ij} = \mathbf{k}_i^T Q \mathbf{k}_j$, where $\mathbf{k}_i$ is the i^{th} column of K, i.e $\mathbf{k}_i = X\mathbf{x}_i$.

To express the regularization term $\|A\|^2$ using Q and K we exploit the cyclic property of the trace operator and note that

$$\begin{array}{rcl} \|A\|^2 & = & Tr(AA^T) = Tr(WXX^TW^T) = \\ & = & Tr(W^TWXX^T) = Tr(QK) \end{array}$$

The important property of the above transformation is that the optimization depends only on inner products between the feature vectors $\mathbf{x}_i$, and thus we can use kernel functions $K(\mathbf{x}_i, \mathbf{x}_j)$ to obtain non-linear maps, as is standard in kernel based algorithms (Schölkopf & Smola, 2002).

In order to obtain a tractable SDP problem, we assume as before that $p = n$. We thus have the following Kernel Pairwise SDE (KPSDE) optimization problem:

KPSDE:

$$\begin{array}{lll} \min_{Q,\xi,b} & \sum_{ij} s_{ij} d_{ij}^2 + \gamma Tr(QK) + \beta \sum_{ij} \xi_{ij} & \\ s.t. & s_{ij} d_{ij}^2 \leq s_{ij} b_i + \xi_{ij} & \forall ij \\ & Tr(KQK) \leq 1 & \\ & g_{ij} = \mathbf{k}_i^T Q \mathbf{k}_j & \\ & d_{ij}^2 = g_{ii} + g_{jj} - 2g_{ij} & \\ & Q \succeq 0 & \end{array}$$

Since it may not be possible to find a non-zero parametric mapping such that the constraints are exactly satisfied, we added slack variables ξ which allow constraints to be violated, and a term to the objective that minimizes these violations, weighted by a positive factor β. Note that in principle the non-parametric case may be non-feasible as well, so that slack variables may also required there. However, the non-parametric problem is much less constrained than the parametric one, and in the experiments we performed, it always had an exact solution with no slack violations.

Also note that we have dropped the constraint that points be centered at the origin. This constraint can be automatically satisfied by centering the input points $\mathbf{x}_i$ at the origin. This set would remain centered under any linear transformation. Importantly, centering can also be performed in the kernel space, as is done in Kernel PCA (Schölkopf & Smola, 2002).

The objective of the KPSDE problem is linear in the Q, ξ, b variables, and so are the constraints, so that the problem constitutes a standard SDP.

To recover the projection matrix A from the matrix Q, we need to perform a Singular Value Decomposition of the matrix $A^T A = X^T Q X$. Although this cannot be done explicitly when X is high or infinite dimensional (e.g., with RBF kernels), one can calculate low dimensional projections via a procedure similar to Kernel PCA. Here we use the procedure as described in (Globerson & Roweis, 2006).

In the case of the RBF kernel, the parametric form of the resulting low dimensional mapping is:

$$\phi(\mathbf{x}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \exp(-\|\mathbf{x}_i - \mathbf{x}\|/\sigma^2) \qquad (6)$$

where $\boldsymbol{\alpha}_i \in \Re^p$ are the low dimensional vectors obtained from the SVD. In effect, the points $\boldsymbol{\alpha}_i$ are a set of "basis embeddings". A new point $\mathbf{x}$ is mapped to a mixture of these points weighted by the proximity of $\mathbf{x}$ to the points in $\mathbf{x}_i$. Note that $\mathbf{x}_i$ itself will not be mapped exactly to $\boldsymbol{\alpha}_i$, but $\boldsymbol{\alpha}_i$ will always have the largest weight in the mixture corresponding to $\mathbf{x}_i$.

5 Convex Duality and Laplacian Eigenvalues

It is well known that convex optimization problems have equivalent duals (Boyd & Vandenberghe, 2004). It was shown in (Sun et al., 2006) that the convex dual of SDE is the minimization of mixing time in a continuous time Markov network. Here, using similar duality transformations, it can be shown that the dual of our PSDE problem is also an eigenvalue problem. Consider the PSDE convex optimization problem, and for simplicity assume there is only a single variable $b_i = b$, and $s_{ij} = s_{ji}$. The convex dual of this PSDE can be shown to be:

$$\begin{array}{ll} \min & \text{MaxEig}[L(S \circ (Y+1))] \\ \text{s.t.} & \text{Tr}[YS] = 0\,, \quad Y \geq 0 \end{array} \qquad (7)$$

Where the operator $L(G)$ is defined as $L(G) = \text{diag}(\mathbf{1}^\top G) - G$, and $\circ$ denotes element-wise multiplication. The operator $\text{MaxEig}(A)$ returns the large eigenvalue of A. The constraint on Y is element-wise non-negativity and *not* positive semidefiniteness. (Note that while $L(G)$ corresponds to the Laplacian for non-negative matrices G, in this case the argument of L may have negative entries, and thus it does not correspond exactly to the conventional Laplacian.) The zero trace constraint $\text{Tr}[YS] = 0$ may be viewed as a *flow constraint* where for each point $\mathbf{x}_i$ the weight on its similar points should equal that on its dissimilar points. It remains an interesting problem to find a natural graph flow, or mixing time problem, which corresponds to the above optimization problem.

6 Efficient Optimization

Our current implementation uses the CSDP package (Borchers, 1999) for solving SDPs. The PSDE optimization problem has potentially $O(n^2)$ constraints and variables. The associated SDP may be too large to solve using standard solvers. An effective solution to this problem, previously employed in SDE, is to first solve for a partial set of constraints, and then add

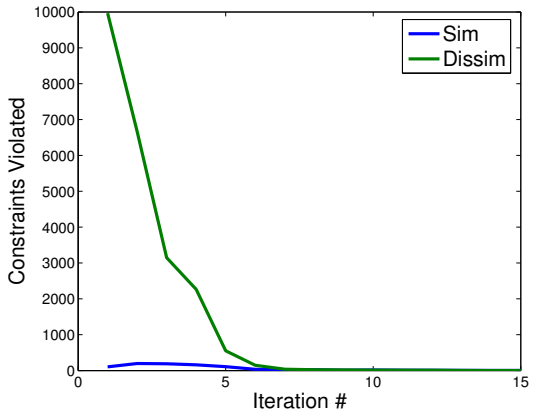


Figure 2: Number of constraint violations as a function of iteration number. One similarity and dissimilarity constraints are added to each point, per iteration. The count is over the un-optimized constraints.

a subset of the most violated constraints, until convergence. While in the worst case, this strategy may converge only after all the original constraints have been added, this will only occur in pathological examples. Figure 2 shows the number of violated similarity/dissimilarity constraints for the parametric embedding KPSDE run on the NORB dataset (Section 8), where $n = 162$.

In the parametric version of the algorithm, the constraints are no longer sparse (although they are low rank), and thus the problem becomes computationally costly. To reduce computational costs, we use a trick similar to that in (Weinberger et al., 2005). Recall that the projection matrix in the kernel case had the form $A = WX$ where the rows of X are the input points. Instead of using all of X, we take only a subset of $r < n$ of its rows. Thus the SDP matrix Q in this case has only size $r \times r$ and the problem is easier to solve. Note that we still use *all* the other $n - r$ points for specifying the constraints. In the implementation used here, we chose r random elements of X although one could also potentially optimize this set.

7 Related Work

Metric embedding algorithms seek an embedding which reproduces a given set of distances. The simplest example of such an algorithm is Metric Multidimensional Scaling (MDS) (Cox & Cox, 1984). Our method diverges from that approach since the exact values of distances are not known, only some constraints on their relative magnitudes. It is thus much more closely related to non-metric MDS, which seeks an embedding that preserves the ranking of distances in a given distance matrix. Our approach differs from non-metric MDS in several important respects. First, it does not assume any distance matrix as input, but rather only similarity relations. Second, our unfolding objective is an effective method for obtaining low dimensional solutions while maintaining convexity of the optimization. Third, we offer both parametric and non-parametric embeddings. This is an advantageous property, since one can first uncover the *ideal* pairwise similarity manifold via a non-parametric method (PSDE) and then proceed to find its functional form using the parametric method (KPSDE) as in Sec. 8.

Another related line of work is that of graph visualization or drawing (Kaufmann & Wagner, 2001). Given a graph G the goal is to draw it in two or three dimensions such that the resulting representation is *readable* or *aesthetic*. For example, one desired property of such a visualization is that the number of crossing edges is minimized, although this objective is hard to optimize directly. Our approach is related to graph drawing if one considers similar pairs as neighboring edges on a graph.[2] Indeed some graph drawing algorithms (e.g. spring models) also rely on minimizing the distance between neighboring vertices, but do not do so in a constrained optimization framework as we present here. Harel and Koren (2002) present a graph drawing method that is related to ours in that it first embeds vertices into a high dimensional space and then uses PCA to obtain a low dimensional embedding. It will be interesting study our algorithm in the context of graph drawing, and to obtain theoretical results regarding the *readability* of its embeddings.

The current method uses some ideas from recent works by Weinberger and Saul. Most importantly, it uses an objective similar to the maximum variance unfolding objective of the SDE method (Weinberger & Saul, 2004), but under a non-metric setting. Since the constraints here do not act as rigid rods as in SDE, we add a norm constraint on the embedding, so that it cannot expand to infinity. Our PKSDE algorithm is also related to recent metric learning methods for supervised classification (Weinberger et al., 2006; Shalev-Shwartz et al., 2004; Globerson & Roweis, 2006) which search for a metric under which vectors in the same class are mapped to nearby points. PKSDE differs from these in the structure of its constraints and objective, which are designed to obtain *low dimensional* embeddings for general pairwise similarity input. Brand (2003) also studies a kernel based embedding algorithm, but in the context of spectral clustering.

Finally, a recent work (Hadsell et al., 2006) addressed a setting similar to ours, where one is given pairwise sim-

[2]Since we allow for dissimilar pairs ($s_{ij} = -1$) as well as unrelated pairs ($s_{ij} = 0$) our input is more general than a graph adjacency matrix

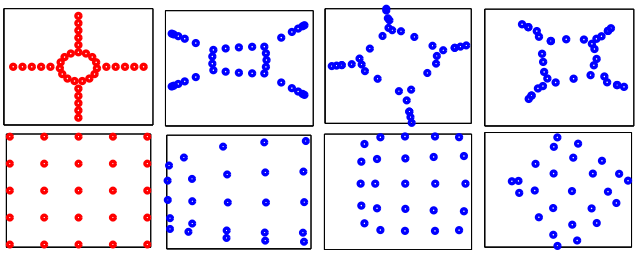


Figure 3: PSDE embedding of star and grid structures. The left column shows the original points (which the algorithm does not have access to). The right columns show the two dimensional embeddings recovered by the PSDE algorithm, based only on pairwise binary similarities. Different columns correspond to a different choice of the number of nearest neighbors used in constructing the similarity matrix ($k = 2, 3, 4$ from left to right).

ilarity as input, and seeks a low dimensional map preserving it. The algorithm in (Hadsell et al., 2006) used a convolutional network as the functional map, with a single objective which combines maximizing dissimilar distances and minimizing similar ones. Unlike our method, theirs does not impose any constraints, and thus the solution may violate a large set of the constraints in an uncontrolled manner. Furthermore, the optimization problem is non-convex and may be stuck in local minima. An advantage of (Hadsell et al., 2006) is that it works in a low dimensional space directly, and is thus likely to be less computationally intensive than our method, especially for large datasets.

8 Experimental Evaluation

The PSDE algorithm may be used to uncover underlying structure given only pairwise relations. We first illustrate this on simple *star* and *grid* structures, shown in Figure 3. The input to the algorithm is a similarity matrix generated by considering all the k nearest neighbors of a point $\mathbf{x}_i$ to be similar to $\mathbf{x}_i$, and all other points considered dissimilar (here we experiment with $k = 2, 3, 4$). The algorithm does not have access to the geometric positions of the original points. The resulting embeddings are shown in the right columns of Figure 3. It can be seen that the similarity measure of the original structure is well preserved by the recovered embedding, although the exact coordinates are not. This is to be expected, since the algorithm does not have any access to this representation. Also, different values of k, although they generate different similarity matrices, give qualitatively similar embeddings.

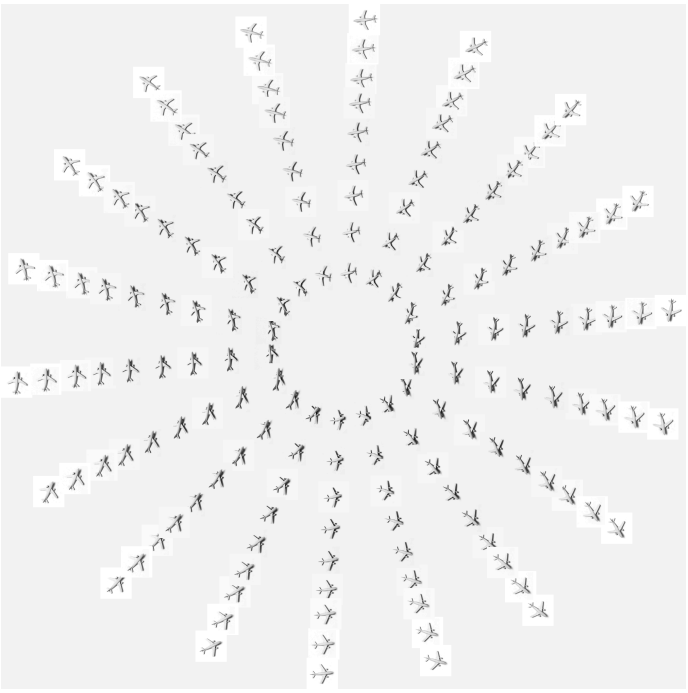
8.1 Image Embedding

Embedding algorithms are commonly used to study the manifold structure of image sets (Roweis & Saul, 2000; Tenenbaum et al., 2000). This is a challenging task since the standard Euclidean distance between pixel maps is usually not very indicative of semantic similarity between images. For example a translated image may be very different from the original, if one only considers pixel values. Here we apply our pairwise embedding method to the NORB dataset (LeCun et al., 2004), which has been used in other recent works on embedding (e.g. the method of Hadsell et al. (2006) mentioned in Section 7).

The small NORB dataset, which we use here, consists of images of a given object (e.g., an airplane) taken at different azimuths ($0, 20, \ldots, 340$ degrees), different elevations ($30, 35, \ldots, 70$ degrees) and four illumination conditions. The raw images are shown in Figure 4. Here we are interested in recovering the manifold structure related to azimuth and elevation. Intuitively, such a manifold may be represented by a cylinder. We wish to obtain this representation from pairwise similarity data alone.

For the experiment we used images of an airplane taken at all azimuths and elevations and at one illumination condition. This resulted in a total of 162 images. The pairwise similarity presented to the embedding algorithm was $s_{ij} = +1$ if two images differed by at most one level of azimuth *and* elevation, and $s_{ij} = -1$ otherwise. We first ran the non-parametric embedding algorithm (Section 3), to see what manifold structure is implied by the similarity data alone. The result, shown in Figure 8.1, is a nearly perfect cylinder which captures the expected azimuth/elevation structure.

The non-parametric result implies that the given similarity data may be represented without distortion in three dimensions. It is thus natural to apply KPSDE to obtain a parametric version of this embedding. We used KPSDE with an RBF kernel on 142 images, setting aside 20 points to test out of sample behavior. We also used a subset of $r = 100$ random data points to speed up the optimization, as explained in Section 6. The parameters σ, γ, β where chosen to maximize the weight of the first three eigenvalues of the Gram matrix. At the optimum, these three eigenvalues captured 94% of the sum of all 100 eigenvalues. The results for the training data and out of sample points are shown in Figure 6. It can be seen that the embedding also captures the azimuth/elevation parameterization of the manifold, and that out of sample points are mapped to areas in the map corresponding to their azimuth and elevation.



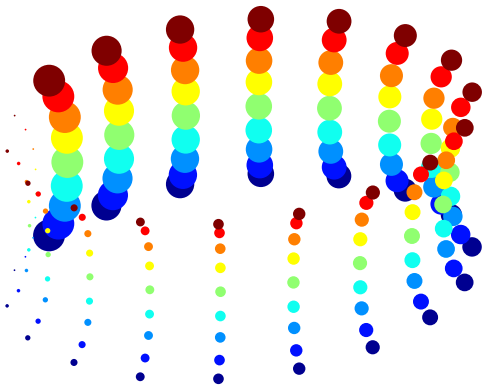


Figure 5: Non-parametric embedding (PSDE). **Above:** Points representing the embedded images in three dimensions. Different colors correspond to different elevations, and marker sizes correspond to different azimuths (smallest marker is azimuth 0 and largest is azimuth 340).
Left: the corresponding images embedded in two dimensions using the coordinates $(x\sqrt{z}, y\sqrt{z})$.

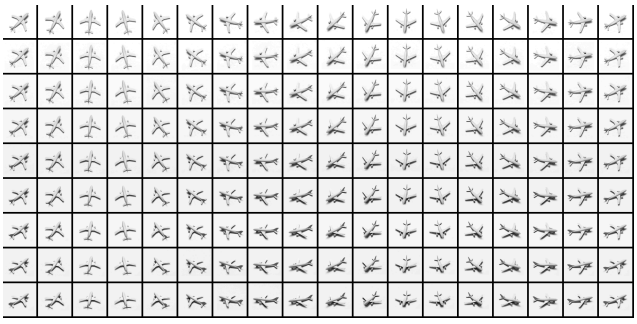


Figure 4: NORB airplane image data. Images with similar azimuth and elevation (neighbors in the rectangular layout above. The rightmost and leftmost columns also have a similar azimuth) were considered similar, and all other dissimilar.

9 Discussion & Conclusions

We have presented a novel method for obtaining low dimensional embeddings of objects based only on pairwise similarity data. Our algorithm employs a variant of semidefinite embedding (Weinberger & Saul, 2004) to generate a convex semidefinite program whose solution gives an embedding in a high dimensional space, which can then be projected to a low dimensional one. One advantage of this approach is that it may find an embedding which satisfies the set of constraints, even if such an embedding does not exist in a low dimensional space. This can aid in uncovering the dimensionality of the underlying manifold, and estimating the loss incurred in projecting it to low dimensional spaces.

The current formalism uses a binary measure of similarity. However in some cases one may have access to more complex relations such as an ordered similarity measure. For instance, our input may be in the following form: point x_i is more similar to x_j than to x_k, but also more similar to x_k than to a fourth point x_l (i.e. $x_j \leq_i x_k \leq_i x_l$). The PSDE method may be extended to reflect such relations. We are currently studying this extension and its applications.

Finally, we also presented a parametric method, which optimizes the same objective and constraints as the parametric one, but yields a functional map, which can be used for out of sample points. The two methods can be used jointly to first find a dimension where the given similarity relations can be faithfully represented when the mapping is not restricted, and then seek a functional form of this map. We note that our kernel extension could in principle be applied to other constraint based embedding algorithms such as the original SDE method, thus allowing them to generalize to out of sample points.

The generality of these new methods, and their reliance on a very minimal form of input, should make them applicable to a wide range of fields, from modeling protein-protein interactions, to mapping of social

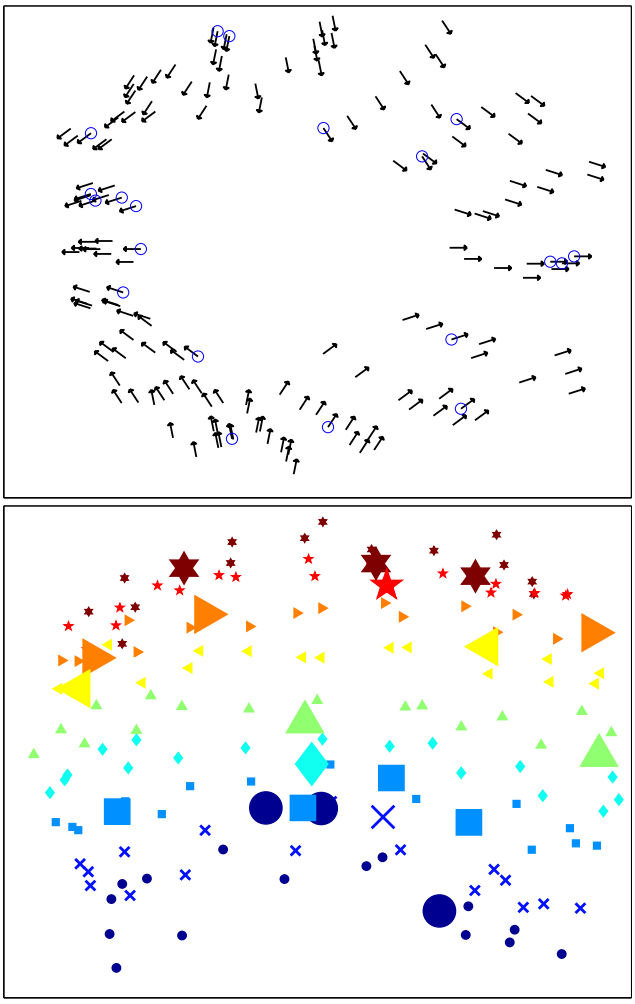


Figure 6: Parametric embedding (KPSDE), using an RBF kernel. Two projections of the $3D$ embedding are shown. The top panel shows a projection that illustrates the azimuth mapping. Arrows are drawn at each embedding point with a direction corresponding to the azimuth of the original image. For images in the test set, the base of the arrow is marked with a circle. The bottom panel shows a projection that illustrates the elevation mapping. Different elevations correspond to different colors and shape. Points in the test set are indicated by a marker which is larger than the training points.

networks.

Acknowledgments AG is supported by a fellowship from the Rothschild Foundation - Yad Hanadiv.

References

Borchers, B. (1999). CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, *11*, 613–623.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge Univ. Press.

Brand, M. (2003). Continuous nonlinear dimensionality reduction by kernel eigenmaps. *Proc. of Int. Joint Conf. Artif. Intel.*.

Cox, T., & Cox, M. (1984). *Multidimensional scaling*. London: Chapman and Hall.

Globerson, A., & Roweis, S. (2006). Maximally collapsing metric learning. *NIPS, Volume 18*.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *Proc. of CVPR*.

Harel, D., & Koren, Y. (2002). Graph drawing by high-dimensional embedding. *Int. Symp. on Graph Drawing*.

Kaufmann, M., & Wagner, D. (Eds.). (2001). *Drawing graphs :methods and models*. Springer.

LeCun, Y., Huang, F., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. *Proc. of CVPR*.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–6.

Schölkopf, B., & Smola, A. J. (Eds.). (2002). *Learning with kernels*. MIT Press.

Shalev-Shwartz, S., Singer, Y., & Ng, A. (2004). Online learning of pseudo-metrics. *Proc. of ICML*.

Sun, J., L., S. B., Xiao, & Diaconis, P. (2006). The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Review*, *48*, 681–699.

Tenenbaum, J., de Silva, V., & Langford., J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. *NIPS, Volume 18*.

Weinberger, K. Q., Packer, B. D., & Saul, L. K. (2005). Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. *Proc. of AISTATS*.

Weinberger, K. Q., & Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. *Proc. of CVPR*.