CLASSIFICATION CONSTRAINED DIMENSIONALITY REDUCTION

Jose A. Costa and Alfred O. Hero III

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109
Emails: {jcosta,hero}@umich.edu

ABSTRACT

In this paper, we propose a nonlinear dimensionality reduction method aimed at extracting lower-dimensional features relevant for classification tasks. This is obtained by modifying the Laplacian approach to manifold learning through the introduction of class dependent constraints. Using synthetic data sets, we show that the proposed algorithm can greatly improve both supervised and semi-supervised learning problems.

1. INTRODUCTION

Continuing technological advances in both sensing and media storage capabilities are enabling the development of systems that generate massive amounts of new types of data and information. However, the high dimensional nature of data sets produced by today's medical information systems or video surveillance applications, for example, poses challenging problems in the application of current signal processing tools. Nevertheless, such signals often contain fundamental features that are concentrated on lower dimensional subsets – curves, surfaces or, more generally, lower-dimensional manifolds – thus permitting substantial dimension reduction with little or no loss of content information. In the recent past, this subject has received substantial attention from researchers in machine learning, computer vision and statistics, leading to the introduction of several manifold learning algorithms (see webpage [1] for an extensive list of references).

Although dimensionality reduction is usually invoked as a tool to improve classification, regression, denoising or visualization tasks, among other applications, current algorithms do not use this information to find a particular lower dimensional representation of the data. For example, in the classification problem, the lower dimensional embeddings found by many popular algorithms generally induce a nonlinear mixing of the classes, resulting in an harder problem in the embedded domain than in the original high dimensional space. However, incorporating classification information in the specification of the data embedding can lead to improvements in classification performance. In particular, by designing a classifier based on a "good" lower dimensional embedding of the data, instead of the high dimensional space, one might break the well known *curse of dimensionality*.

The goal of this paper is to introduce a dimensionality reduction method where the class labels of data points having a manifold structure are incorporated in the construction of a lower dimensional data embedding. It seems intuitive that such class dependent manifold embedding algorithm can improve the performance of supervised and semi-supervised learning tasks.

Currently, the only other applications to classification that take advantage of the manifold structure of the data are from semi-supervised learning problems [2–4], although the perspective of these papers is one of regularization, instead of the dimensionality reduction approach followed here.

2. GRAPH LAPLACIANS AND MANIFOLD EMBEDDINGS

Let $\mathcal{X}_n = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ be a set of n points constrained to lie on an m-dimensional submanifold $\mathcal{M}$ of $\mathbb{R}^d$. The manifold learning problem consists in finding an embedding of $\mathcal{X}_n$ into a subset $\mathcal{Y}_n = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ of a lower m-dimensional space $\mathbb{R}^m$ (usually $m \ll d$), without any prior knowledge about $\mathcal{M}$ besides its finite sampling $\mathcal{X}_n$.

A common framework used to represent the geometric information about $\mathcal{M}$ carried by its sampling $\mathcal{X}_n$ is through the use of adjacency graphs. Let $G = (V, E)$ be an undirected weighted graph, whose vertex set V is given by the data points, i.e., $V = \mathcal{X}_n$, and E contains edges connecting *adjacent* vertices. The edge set E is associated with an $n \times n$ weight matrix W specifying adjacency relations between vertices, such that w_{ij} is a function of the similarity between points i and j. The weights are assumed nonnegative and symmetric. Although there are many choices for G, throughout this paper we consider nearest neighbor (NN) graphs with a weight matrix derived from the heat kernel [2]. The construction of this graph proceeds as follows:

This research was partially supported by the National Science Foundation under ITR grant CCR-0325571.

1. For a fixed neighborhood parameter $k \in \mathbb{N}$, construct a k-NN graph on $\mathcal{X}_n$, i.e., put and edge between points i and j if i is one of the k-NN's of j or j is one of the k-NN's of i.
2. For a fixed scale parameter $\epsilon > 0$, assign weight

   $$w_{ij} = \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\epsilon}\right\} ,$$

   if vertices i and j are connected and $w_{ij} = 0$ otherwise.

Following the Laplacian eigenmaps approach [5], we formulate manifold learning as the problem of minimizing the cost function

$$E(\mathcal{Y}_n) = \sum_{ij} w_{ij} \, \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \tag{1}$$

in the embedding points $\mathcal{Y}_n \subset \mathbb{R}^d$. This cost function naturally accounts for the geometry of $\mathcal{X}_n$, as mapping close points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the manifold to faraway points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ in $\mathbb{R}^m$ results in a large penalization. Equation (1) can be rewritten as

$$E(\mathcal{Y}_n) = 2 \, \mathrm{tr}\left(Y \, L \, Y^T\right) , \tag{2}$$

where $Y = [\boldsymbol{y}_1 \dots \boldsymbol{y}_n]$ and $L = D - W$, with D a diagonal matrix with entries $D_{ii} = \sum_j w_{ji}$. L is known as the graph Laplacian of G. After imposing appropriate constraints to remove arbitrary translations and scalings in the embedding, finding a lower dimensional embedding of $\mathcal{X}_n$ reduces to solving the following optimization problem:

$$\arg \min_{\substack{Y D \mathbf{1} = \mathbf{0} \\ Y D Y^T = I}} \mathrm{tr}\left(Y \, L \, Y^T\right) , \tag{3}$$

where I is the $n \times n$ identity matrix and $\mathbf{1}$ is a column vector of ones.

As L is positive semidefinite, the solution to problem (3) is given by the m generalized eigenvectors associated with the m smallest positive generalized eigenvalues that solve

$$L \, \boldsymbol{v} = \lambda \, D \, \boldsymbol{v} . \tag{4}$$

This is equivalent to solving a regular eigenvalue problem for matrix $\tilde{L} = D^{-1/2} \, L \, D^{-1/2}$, the so-called normalized graph Laplacian. If $V = [\boldsymbol{v}_1 \dots \boldsymbol{v}_m]$ is the collection of such eigenvectors, then the embedded points are given by $\boldsymbol{y}_i = (v_{i1}, \dots, v_{im})^T$, $1 \leq i \leq n$.

3. CONSTRAINING THE MANIFOLD EMBEDDING

Assume now that each point of $\mathcal{X}_n \in \mathcal{M}$ (or a subset of them) is associated with a class label, i.e., $\boldsymbol{x}_i$ has label $c_i \in \{-1, 1\}$. For simplicity, we only consider the problem of two classes, although the extension of the method proposed here to a multi-class scenario is straightforward.

We are interested in finding a lower dimensional embedding for $\mathcal{X}_n$ that, unlike common manifold learning algorithms, takes into account the class structure of the data. The goal is to obtain an embedding that tries to separate classes in order to improve training and generalization capabilities of a classifier fitted to the lower dimensional embedded data.

Although we also use graph Laplacians, we do not follow the approach advocated in [2], where dimensionality reduction and classification are seen as a function fitting problem, since the the eigenvectors of the Laplacian provide a natural basis to represent functions on the graph sampling of the manifold.

The method developed here is based on the idea of *maximum alignment* [6] between classes and data points. This idea proceeds as follows. Start by associating with each class a new node on the graph, called *class centers*, inserting an edge of unit weight between this node and all data points with the same class label. Now, if we view the graph edges as springs that pull together nodes in the graph, determining an embedding corresponds to finding data coordinates in an m-dimensional space that minimize the stresses in the system of springs. This will lead to points with the same class label trying to cluster together around the class center, while attempting to preserve the geometric neighborhood structure of the manifold. In this way, the class centers are maximally aligned with the data points.

We now formalize this idea. Let $\boldsymbol{z}_k \in \mathbb{R}^m$ be the class center associated with class k and C be the class membership matrix, i.e., $c_{ki} = 1$ if $\boldsymbol{x}_i$ has label k and $c_{ki} = 0$ otherwise. As before, we find the embedding by minimizing the cost function

$$E(\mathcal{Z}_n) = \sum_{ki} c_{ki} \, \|\boldsymbol{z}_k - \boldsymbol{y}_i\|^2 + \beta \sum_{ij} w_{ij} \, \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 , \tag{5}$$

where $\mathcal{Z}_n = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ and $\beta \geq 0$ is a regularization parameter. Large values of β will produce embeddings that ignore class labels, while small values will produce embeddings that ignore the manifold structure of the data. Of course, in the latter case, points will tend to collapse into the class centers, producing lower dimensional data with little value to train a classifier with good generalization performance.

By defining $Z = [\boldsymbol{z}_1 \, \boldsymbol{z}_2 \, \boldsymbol{y}_1 \dots \boldsymbol{y}_n]$, determining the lower dimensional embedding of $\mathcal{X}_n$ can be once again made equivalent to the following optimization problem:

$$\arg \min_{\substack{Z D \mathbf{1} = \mathbf{0} \\ Z D Z^T = I}} \mathrm{tr}\left(Z \, L \, Z^T\right) , \tag{6}$$

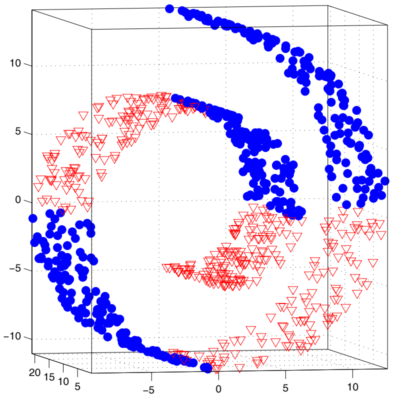


Fig. 1. Swiss roll manifold with 400 points from each of 2 classes, marked as '$\triangledown$' (red) and '$\bullet$' (blue).

where L is the $(n+2)\times(n+2)$ graph Laplacian associated with weight matrix

$$W' = \begin{bmatrix} I & C \\ C^T & \beta W \end{bmatrix} .$$

The solution of problem (6) is again given by the matrix of the generalized eigenvectors associated with the m smallest positive generalized eigenvalues of L, where the first rows correspond to the coordinates of the class centers and the following rows determine the embedding of the original data points.

We remark that the method proposed here extends naturally to the semi-supervised setting, where only partial labeling is available. In this case, the points $\boldsymbol{x}_i$ for which there are no label information will have the corresponding columns of matrix C set to zero, thus imposing no additional constraints.

4. EXAMPLES

We now show through simple examples how the proposed classification constrained dimensionality reduction (CCDR) algorithm works. All the simulations presented here have $\beta = 1$, neighborhood parameter $k = 12$ and the scale parameter ϵ of the heat kernel is set automatically according to [7]:

$$\epsilon = \frac{10}{n}\sum_{i=1}^{n} \min_{j:\,\boldsymbol{x}_j \neq \boldsymbol{x}_i} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2 .$$

Consider the standard 2-dimensional swiss roll manifold in $\mathbb{R}^3$. We sample 400 points uniformly on the manifold from each of two classes, as shown in figure 1. As it can be deduced, there is no linear projection of the data into a 2-dimensional subspace that separates the classes.

Figure 2 shows the results of applying standard manifold learning methods, ISOMAP [8] and Laplacian Eigenmaps, together with the proposed CCDR algorithm to the data set of Figure 1. Recall that both ISOMAP and Laplacian Eigenmaps do not take into account label information when computing the embedding. As a result, although ISOMAP (Figure 2(a)) is able to recover an isometric embedding of the data into the plane, it fails at finding a simple separation of the classes. The Laplacian eigenmaps method (Figure 2(b)) gives similar results, albeit finding an arc-length type parameterization of the data. On the contrary, the CCDR algorithm (Figure 2(c)) computes an embedding where classes are almost linearly separable.

Table 1. Error rates for classification using pre-processing dimensionality reduction versus full dimensional data

no. of train. samples	CCDR	Laplacian	3-NN
300	**4.4 %**	6.4 %	5.0 %
400	**3.6 %**	5.0 %	4.4 %
500	**2.6 %**	3.6 %	3.4 %

To quantify this behavior, we designed a very simple classifier. To classify a new sample, add it to the graph formed by the training set, with unknown label (add a zero column to matrix C), compute the constrained (or simple Laplacian) embedding. and then classify the sample using a simple NN-classifier on the embedded points. We compare this to a baseline NN-classifier on the full dimensional data set. In all the experiments a 3-NN classifier was used. We tested 50 sample points per training set and repeated for 20 random training sets. Table 1 shows the average error rates as a function of the number of training samples. As it can be seen, the CCDR algorithm outperforms the other methods. Supporting the claim that dimensionality reduction without guidance can harm classification performance, it can be observed that the full dimensional NN-classifier does better than a NN-classifier based on the Laplacian embedding.

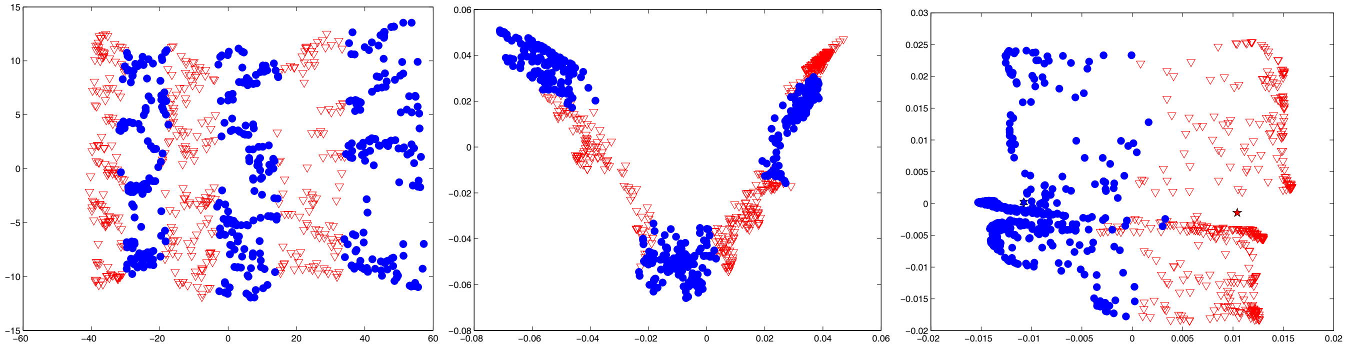
The algorithm proposed here can also be used to improve semi-supervised learning machines. Adopting the method proposed in [2], we have the following algorithm. firstly, compute the constrained embedding of the entire data set, inserting a zero column in C for each unlabeled sample. Secondly, fit a linear classifier to the labeled embedded points by minimizing the quadratic error loss:

$$\ell(\boldsymbol{a}) = \sum_{\substack{i\,:\,\boldsymbol{x}_i \text{ is} \\ \text{labeled}}} \left(c_i - \boldsymbol{a}^T\boldsymbol{y}_i\right)^2 .$$

Thirdly and finally, for an unlabeled point $\boldsymbol{x}_j$, label it using the fitted linear classifier:

$$c_j = \begin{cases} 1 & \text{if } \boldsymbol{a}^T\boldsymbol{y}_j \geq 0 \\ -1 & \text{if } \boldsymbol{a}^T\boldsymbol{y}_j < 0 \end{cases} .$$

Once again, we compare this algorithm to the simple Laplacian equivalent, where the embedding is found using Laplacian eigenmaps, and to a baseline NN-classifier,



(a) ISOMAP

(b) Laplacian Eigenmpas

(c) CCDR

Fig. 2. Applying dimensionality reduction algorithms to the Swiss roll data set of Figure 1. ISOMAP was computed using 8-NN, while both Laplacian Eigenmaps and CCDR used 12-NN.

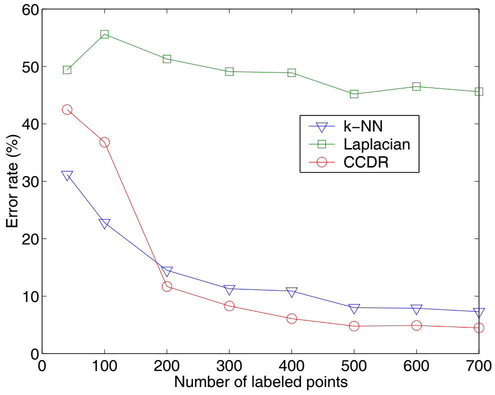


Fig. 3. Percentage of errors for labeling unlabeled samples as a function of the number of labeled points, out of a total of 1000 points on the Swiss roll.

where points are labeled according to the nearest labeled neighbors in the full dimensional space. We chose the best k-NN classifier for $k = 1, 2, 3$. Figure 3 shows the error rates as a function of the number of labeled points, for a total of 1000 points on the Swiss roll. For each fixed number of labeled points, we drew 20 independent data sets and randomly assigned labels, although guaranteeing balanced classes. As it can be seen, for a small number of labeled points, CCDR performs almost as bad as the Laplacian algorithm, as label information is not enough to produce an embedding substantially different from the original Laplacian eigenmaps, where classes are highly mixed (see Fig. 2(b)). However, as the number of samples increases beyond 100, the class constraints start to take effect, driving an embedding that can achieve almost linear separation of classes and thus outperforming all other methods.

5. FUTURE WORK

Several issues should be addressed in order to make the method proposed widely applicable to real life problems. Of prime importance is the development of out-of-sample extensions of the embeddings to new points. Also of interest, is the study of the influence of the regularization parameter β in the classification performance. We are currently applying this method to high-dimensional databases, such as the MNIST database of handwritten digits.

6. REFERENCES

[1] "Manifold learning resource page," `http://www.cse.msu.edu/~lawhiu/manifold/`.

[2] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, pp. 209–239, 2004, Special Issue on Clustering.

[3] X. Zhu, Z. Ghaharamani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. of Int. Conf. on Machine Learning*, Washington DC, August 2003.

[4] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *Advances in Neural Information Processing Systems 14*, 2002.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, June 2003.

[6] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, June 2004.

[7] S. Lafon, *Diffusion Maps and Geometric Harmonics*, Ph.D. thesis, Yale University, May 2004.

[8] 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.