Learning Isometric Separation Maps *

Nikolaos Vasiloglou, Alexander G. Gray, David V. Anderson
Georgia Institute of Technology
nvasil@ieee.org, agray@cc.gatech.edu, dva@ece.gatech.edu

Abstract

Maximum Variance Unfolding (MVU) and its variants have been very successful in embedding data-manifolds in lower dimensionality spaces, often revealing the true intrinsic dimensions. In this paper we show how to also incorporate supervised class information into an MVU-like method without breaking its convexity. We call this method the Isometric Separation Map and we show that the resulting kernel matrix can be used for a binary/multiclass Support Vector Machine in a semi-supervised (transductive) framework. We also show that the method always finds a kernel matrix that linearly separates the training data exactly without projecting them in infinite dimensional spaces.

1 Introduction

Support Vector Machines have been quite successful in separating classes of data that are not linearly separable. The kernel trick lifts the data in a high dimensional Hilbert space usually of infinite dimension [1]. Embedding datasets in infinite dimensional spaces gives the advantage of SVMs to separate data with linear hyperplanes in the lifted space, that otherwise were not separable in the original space. So far it is not clear how the dimensionality of the kernel affects the performance of SVMs. It is not known yet how many dimensions are sufficient for separating the classes. It is very likely that the minimum dimension required for linear separability is smaller than the original dimension of the data. This is because the data might already be embedded in a redundant manifold.

Maximum Variance Unfolding (MVU) [2] along with other manifold learning methods has addressed the problem of reducing the dimensionality of the data by preserving local distances. Most of the times the data end up living in a lower dimensional space. MVU explicitly finds the optimal kernel matrix for the data, by solving a semidefinite program. As a remark MVU usually gives the most compact spectrum [3], revealing the true intrinsic dimension of the dataset very well. The authors of the MVU point though that it has very poor performance when it comes to using the kernel matrix for SVM classification [3] as it doesn't include any information about the linear separability of the classes.

In this paper we introduce a variation of MVU that takes into consideration the linear separability of the classes. The result is a new algorithm, the Isometric Separation Mapping (ISM), that gives unfolding that preserves the class structure of the manifold. The algorithm can be seen as a transductive (semi-supervised SVM), since it requires the test data during training. Previous work on transductive SVMs has been also studied by several researchers. When the choice of the kernel is ad-hoc, the problem becomes very difficult as it boils down to mixed integer programming [4]. In [5] and [6] the authors train the kernel matrix over a set of predefined kernels. Finally in [7] the Laplacian Eigenmap framework is used for training SVMs. Our technique doesn't make any assumption on the kernel function, the only requirement is to preserve isometry on the data.

*This work is supported by Google Grants

In section 2 we give an overview of MVU along with its variants that make it scalable. In section 3 we present the ISM algorithm. Then in section 4 we present a transductive SVM based on the ISM and in the last section 5 we show some experimental results.

2 Maximum Variance Unfolding, Maximum Furthest Neighbor Unfolding.

Weinberger formulated the problem of isometric unfolding as a Semidefinite Programming algorithm [2]. In [8] Kulis presented a non-convex formulation of the problem that requires less memory than the Semidefinite one. He also claimed that the non-convex formulation is scalable. In [9] Vasiloglou presented experiments where he verified the scalability of this formulation by changing the objective function resulting in the Maximum Furthest Neighbor Unfolding (MFNU) algorithm. The latest formulation tends to be more robust and scalable than MVU, this is why we use it for our experiments. Both methods can be cast as a semidefinite programming problem.

Given a set of data $X \in \Re^{N \times d}$, where N is the number of points and d is the dimensionality, the dot product or Gram matrix is defined as $G = XX^T$. The goal is to find a new Gram matrix K such that $rank(K) < rank(G)$ in other words $K = \hat{X}\hat{X}^T$ where $\hat{X} \in \Re^{N \times d'}$ and $d' < d$. Now the dataset is represented by $\hat{X}$ which has fewer dimensions than X. The requirement of isometric unfolding is that the euclidian distances in the $\Re^{d'}$ for a given neighborhood around every point have to be the same as in the $\Re^{d}$. This is expressed in:

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} = G_{ii} + G_{jj} - G_{ij} - G_{ji}, \forall i, j \in I_i$$

where I_i is the set of the indices of the neighbors of the ith point. From all the K matrices MFNU chooses the one that maximizes the distances between furthest neighbor pairs and MVU the one that maximizes the variance of the set (equivalently the distances of the points from the origin). So the algorithm is presented as an SDP:

$$\max_{K} \sum_{i=1}^{N} B_i \bullet K \tag{1}$$

$$\begin{aligned} &\text{subject to} \\ A_{ij} \bullet K &= d_{ij} \quad \forall j \in I_i \\ K &\succeq 0 \end{aligned}$$

where the $A \bullet X = \text{Trace}(AX^T)$ is the dot product between matrices.

$$d_{ij} = G_{ii} + G_{jj} - G_{ij} - G_{ji} \tag{2}$$

B_i has the same structure of A_{ij} and computes the distance of the i_{th} point with its furthest neighbor for MFNU, while for MVU it is just the unit matrix (computes the distance of the points from the origin). The last condition is just a centering constraint for the covariance matrix. The new dimensions $\hat{X}$ are the eigenvectors of K. In general MVU/MFNU gives Gram matrices that have compact spectrum, at least more compact than traditional linear Principal Component Analysis (PCA). The method behaves equally well with MVU. Unfortunately this method can handle datasets of no more than hundreds of points because of its complexity. A non-convex formulation suggested in [10] and tested in [9] is preferred, because it scales better and it has the same global optimum with the convex one.

3 Isometric Separation Maps

Although MVU and its variant MFNU give kernel matrices, experiments published by Weinberger [3] show that they are performing more poorly when it comes to SVM classification. In this section we will show that MVU/MFNU can be modified so that the kernel matrix can be used for classification too.

In traditional SVMs the kernel is chosen ad-hoc and the goal is to find a hyperplane that can linearly separate the classes. The kernel is chosen in such a way that it lifts the data in a high dimensional space hoping that data would be linearly separable. In our approach we have the hyperplane given and we are trying to find the kernel matrix that separates the data along the hyperplane. Finding a kernel matrix like that is trivial as it suffices to add one extra dimension on the data that will be either -1 or 1. What is sort of interesting though is to find the minimum dimensionality of the data, that keeps them linearly separable. In other words, unfold the data to the extent they are linearly separable.

The solution of the problem is the following. We pick one of the points x_A to be normal to the separating hyperplane. The choice of the point doesn't matter since it will just change the orientation of the points in space. Let $x_i \in C_1$ be the points that belong in the same class with x_A, then $k(x_A, x_i) \geq 0$, where $k(x_A, x_i)$ is the generalized dot product between $x_A$andx_i. For points that belong to the opposite class $x_i \in C_2$, $k(x_A, x_i) \leq 0$. Now the problem of MVU/MFNU with linear separability constraints can be cast as the following Semidefinite Program:

$$
\begin{aligned}
\max_{K} \; & \sum_{i=1}^{N} B_i \bullet K \qquad (3) \\
& \text{subject to} \\
A_{ij} \bullet K &= d_{ij} \quad \forall j \in I_i \\
K_{A,i} &\geq 0, \quad i \in C_1 \\
K_{A,i} &\leq 0, \quad i \in C_2 \\
K &\succeq 0
\end{aligned}
$$

Using the same formulation as in [9] we can solve the above problem in a non-convex framework that scales better. Extending the problem for more classes is pretty straight forward. The only modification is to use more anchor points that will serve as normal vectors to the separating hyperplanes. A remarkable result of this formulation is that the above method guarantees zero classification error [1](on the training data) in contrast to other algorithms proposed for learning the kernel in SVM (mentioned in section 1 , where the kernel is learnt as a convex combination of preselected kernels). Another remark on the ISM is that it is not a max margin classifier because it doesn't regularize the norm of the normal vector. It is not possible to do it since we need to also preserve the local distances. If instead we preserve the ranks between the distances [12] and not the exact distances then it is possible to maximize the margin.

4 Transductive SVMs

The method described above can also be used as a transductive SVM in a semi-supervised setting. Transductive SVMs are in general difficult problems. If the kernel is preselected then a mixed integer problem has to be solved. If the kernel is learnt from the data then as we mentioned before no guarantee can be given that the training data are linearly separable. In ISM the kernel is trained over all data, using all neighborhood information. After solving the optimization problem, the classification information for the test data will be on the sign of $K_{A,i}$ $\forall i \in T$, where T is the test set.

5 Example

In order to verify our algorithm we tested in on the swiss roll dataset (1500 points). Two classes were defined on the swiss roll that they were not linearly separable. ISM was performed on the dataset. Embedding in 2 dimensions was not possible as the isometry cannot be preserved (the algorithm terminated with 2% error on the isometry). Embedding was though possible in 3 dimensions where the algorithm terminated with 0.01% error in the distance isometry. In both cases the classification error was zero. As we see in fig. 1 MVU unfolds the dataset in strip where the classes are not linearly separable. The ISM on the contrary transforms the manifold in a set that preserves the distances (k neighborhood=10) and divide the two classes in a linearly separable way.

[1]The above problem is always feasible provided that $k \ll N$. As long as the k neighbors belong to the tangent space and the manifold is smooth, a folding (isometric transform) of the manifold along a hyperplane always exists [11]. If all pair distances are given then the Gram matrix is uniquely defined and the problem might be infeasible.

In the future we intend to test the method in larger datasets and compare its performance with other techniques mentioned earlier. Also a more rigorous bound for the maximum number of distances that can be preserved so that the manifold remains linearly separable needs to be formed. Interesting theorems helping in that direction can be found in the area of Euclidean Distance Geometry [13].

Figure 1: Top row shows a gradient-colored swiss roll and also colored in a bimodal way along with the MVU unfoldings. Bottom row shows the ISM in a gradient and bimodal colored way from different views.

References

[1] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press New York, NY, USA, 2004.

[2] K.Q. Weinberger and L.K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," *Proceedings of the Twenty First National Conference on Artificial Intelligence (AAAI-06)*, 2006.

[3] K.Q. Weinberger, F. Sha, and L.K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the twenty-first international conference on Machine learning*. ACM New York, NY, USA, 2004.

[4] K. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," *NIPS*, 1999.

[5] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," *The Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[6] S.J.Z. Kim, A. Magnani, A.K. Koh, and S. Boyd, "Learning the kernel via convex optimization," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2008, pp. 1997–2000.

[7] M. Belkin, P. Niyogi, and V. Sindhwani, "On manifold regularization," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*, 2005.

[8] B. Kulis, A.C. Surendran, and J.C. Platt, "Fast Low-Rank Semidefinite Programming for Embedding and Clustering," in *Eleventh International Conference on Artifical Intelligence and Statistics, AISTATS 2007*, 2007.

[9] D. Anderson N. Vasiloglou, A. Gray, "Scalable Semidefnite Manifold Learning," *MLSP*, 2008.

[10] S. Burer and R.D.C. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, 2003.

[11] J.M. Lee, *Introduction to Smooth Manifolds*, Springer, 2003.

[12] H. Ouyang and A. Gray, "Learning dissimilarities by ranking: from SDP to QP," in *ICML*, 2008, pp. 728–735.

[13] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*, Lulu. Com, 2006.