Learning Large Margin Mappings

Andrew G. Howard
Department of Computer Science
Columbia University
New York, NY 10027
ahoward@cs.columbia.edu

Tony Jebara
Department of Computer Science
Columbia University
New York, NY 10027
jebara@cs.columbia.edu

Abstract

We present a method to simultaneously learn a mixture of mappings and large margin hyperplane classifier. This method learns useful mappings of the training data to improve classification accuracy. We first present a simple iterative algorithm that finds a greedy local solution and then derive a semidefinite relaxation to find an approximate global solution. This relaxation leads to the matrix mixture of mappings formulation which generalizes both the mixture of mappings and the mixture of kernels framework. More efficient algorithms based on the extragradient method are introduced to work on larger problems and to extend the basic framework to a multitask setting. We then apply our method to the novel task of learning monotonic transformations which cannot be easily addressed in a kernel learning framework. Experiments directly comparing kernel learning and mapping learning demonstrate the usefulness of the technique.

1 Introduction

In the past decade, large margin classifiers based on the support vector machine (SVM) formulation have been very successful in applied domains. A large part of their success can be attributed to the application of nonlinear mappings to the training data either explicitly or implicitly with kernels. Often, these preprocessings are applied in an ad hoc manner guided by a combination of intuition and exhaustively searching over a set of commonly used mappings and kernels.

More recently, a line of research has set out to learn these kernels or mappings directly from data. In [4], optimization algorithms were proposed to find a mixture of base kernels $K(\vec{x}, \vec{x}') = \sum_i m_i K_i(\vec{x}, \vec{x}')$. In this paper, rather than a learning a mixture of kernels, we will investigate mixtures of mappings $\Phi(\vec{x}) = \sum_i m_i \phi_i(\vec{x})$ as depicted in figure 1. This was originally proposed in [3] to learn monotonic transformations and expanded upon in [2]. This leads to learned kernels of the form: $K(\vec{x}, \vec{x}') = \sum_{i,j} m_i m_j \phi_i(\vec{x}) \phi_j(\vec{x}')$. We also explore a generalization, the matrix mixture of mappings: $K(\vec{x}, \vec{x}') = \sum_{i,j} M_{i,j} \phi_i(\vec{x}) \phi_j(\vec{x}')$. This subsumes the previous two settings where the mixture of kernels can be replicated with a diagonal M and the mixture of mappings can be replicated with a rank one M.

2 Learning Mixtures of Mappings

The basic problem proposed in this paper is to learn both a maximum margin hyperplane and an unknown mapping $\Phi(\vec{x})$ simultaneously where $\Phi(\vec{x}) = \sum_i m_i \phi_i(\vec{x})$ is defined as a mixture of base mappings $\phi_i(\vec{x})$. This leads to the following SVM like optimization problem

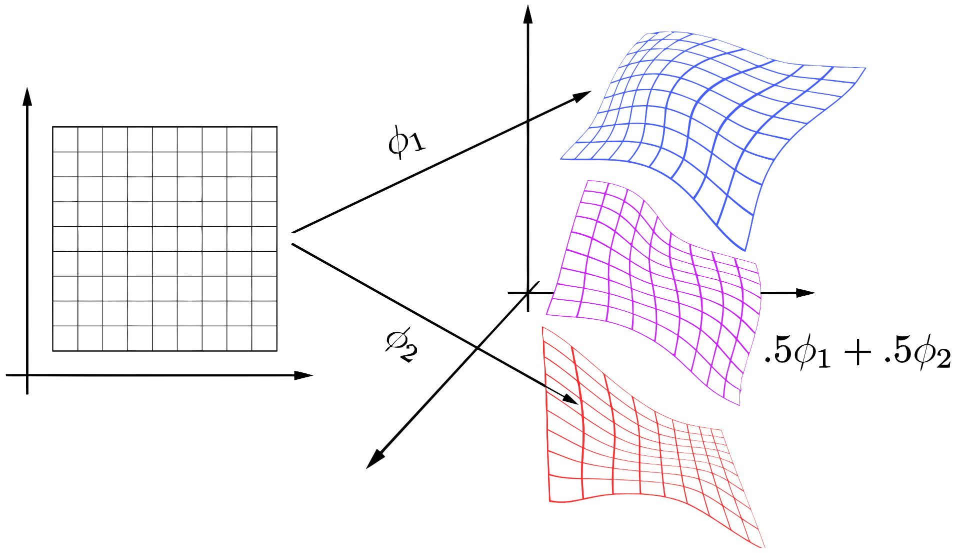


Figure 1: Two mappings ϕ_1, ϕ_2 and their mixture $.5\phi_1 + .5\phi_2$.

augmented with the task of learning $\Phi(\vec{x})$:

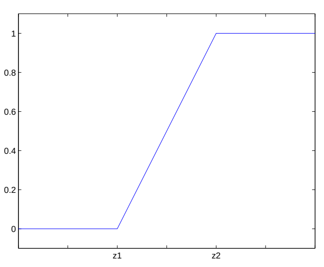
$$\begin{aligned} \min_{\vec{w},\vec{\xi},b,\vec{m}} \quad & \|\vec{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i\left(\left\langle \vec{w}, \sum_{j=1}^{J} m_j\phi_j(\vec{x_i})\right\rangle + b\right) \geq 1 - \xi_i \;\forall i \\ & \xi_i \geq 0, m_j \geq 0, \sum_j m_j \leq 1 \;\forall i, j \end{aligned}$$

where $\vec{\xi}$ are the standard SVM slack variables, $\vec{w}$ and b are the maximum margin hyplerplane for the training set that has been transformed via $\Phi(\vec{x})$ with learned $\vec{m}$, the vector of mixing weights. The formulation is a nonconvex problem which can then be solved with a simple greedy coordinate descent algorithm. The algorithm iteratively solves for $\vec{w}, b$ and $\vec{\xi}$ with $\vec{\Phi}$ fixed by using an SVM solver and then optimizes $\vec{m}$ and $\vec{\xi}$ with $\vec{w}$ and b fixed via a linear program. This EM like algorithm efficiently finds a locally optimal solution in around ten iterations.

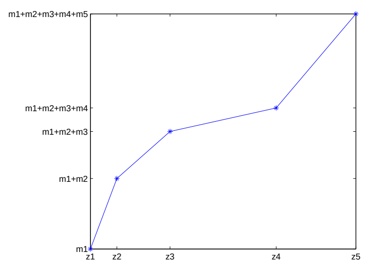
3 Convex Relaxation

The previous algorithm often finds good solutions but sometimes gets trapped in local minima. One method to overcome this issue is to relax the nonconvex terms transforming the optimization into a semidefinite program (SDP) which can be solved for a global (approximate) solution. In the dual formulation, the mixing weights are replaced with a rank one positive semidefinite mixing matrix. The problem is relaxed by dropping the rank constraint yielding the now convex SDP:

$$\begin{aligned} \min_{M,t,\lambda,\vec{\nu},\vec{\delta}} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} Y\sum_{i,j} M_{i,j}\phi_i(X)^T\phi_j(X)Y & \vec{1} + \vec{\nu} - \vec{\delta} + \lambda\vec{y} \\ (\vec{1} + \vec{\nu} - \vec{\delta} + \lambda\vec{y})^T & t - 2C\vec{\delta}^T\vec{1} \end{pmatrix} \succeq 0 \\ & M \succeq 0, \sum_{j=1}^{J} M_{j,j} \leq 1, M_{0,j} \geq 0, \sum_{j=1}^{J} M_{0,j} \leq 1, M_{0,0} = 1, \vec{\nu} \geq \vec{0}, \vec{\delta} \geq \vec{0} \end{aligned}$$



(a) Ramp function $\phi_1(x)$



(b) $\Phi(x) = \sum_{j=1}^{5} m_j \phi_j(x)$

Figure 2: Building blocks for piecewise linear functions.

where M is the relaxed mixing matrix and λ, $\vec{\nu}$, $\vec{\delta}$ arise from the dual transformation. This relaxation would be exact if M were rank one. For learning with large data sets, we use the extragradient algorithm as an alternative to a general SDP solver because semidefinite programs can be computationally expensive. The extragradient method solves a simpler saddle point problem that is equivalent to the proposed SDP. This formulation also allows us to efficiently extend the method to more complicated scenarios such as a multitask setting where a single mixture mapping is learned over multiple related tasks.

4 Monotonic Transformations

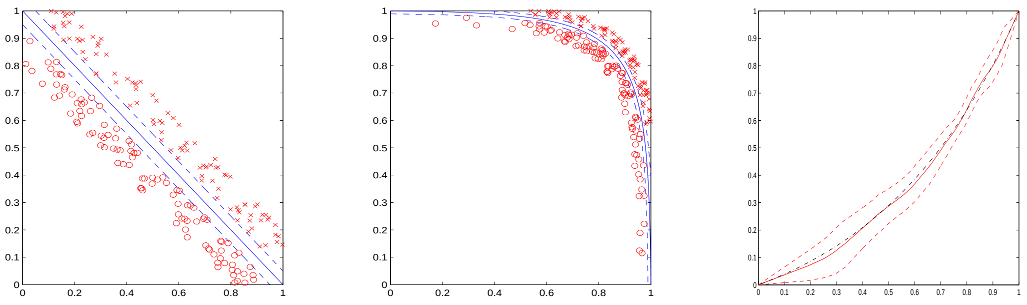
One application of the mapping learning framework is to learn monotonic transformations. In many domains, such as document classification, image histogram classification and gene microarray experiments, fixed monotonic transformations can be useful as a preprocessing step which can often substantially improve results. Our method can learn these monotonic transformations automatically while training a large margin classifier without any prior domain knowledge. Truncated ramp functions, as in figure 2(a), can be used as base mappings that are combined in a mixture to form piecewise linear monotonic functions as in figure 2(b).

5 Experiments

In these experiments we investigated the efficacy of the greedy mixture of mappings algorithm (Greedy) and the convex matrix mixture of mappings algorithm (Convex). A synthetic experiment was first investigated to demonstrate intuitively the application of the mixture of mappings to the problem of learning monotonic transformations. The next experiment involves classification of image histograms and demonstrates the usefulness of the learned mappings in the single task setting and also the utility of estimating them jointly in a multitask setting. More experiments, including document and gender image classification, can be found in [2, 3].

In the first experiment, synthetic data was generated by sampling near a linear decision boundary as in figure 3(a). It was then transformed with a monotonic transformation such as the normalized logarithm in figure 3(b). To correctly classify this data with a linear hyperplane, the correct inverse monotonic transform must be learned by the proposed methods. The recovered transformation is plotted in red in figure 3(c), the standard deviation over ten repetitions in dashed red and the true target transformation is plotted in black. The proposed methods are compared to linear SVM and a mixture of kernels based on the truncated ramp functions in table 1. The linear SVM clearly suffers in this nonlinear setting. The two mapping learning algorithms perform best because they are searching over the correct set of functions which cannot be represented by the kernel learning algorithm.

The second experiment classified image histograms from the Corel dataset. Parameterized monotonic transformations of the form x^a were shown to substantially outperform commonly used kernels in [1]. Four classes of animals were chosen, and all six pairwise classification tasks were solved. The proposed methods were compared to linear, RBF and polynomial SVM as well as the x^a transforms and the multiple kernel learning algorithm. The mapping learning algorithms again outperform the competition although the greedy algorithm tied the kernel mixture most likely due to problems with local minima. The multitask learning algorithms were also applied to this set of problems yielding improvements over independently trained models. All results are reported in table 2.



(a) Sampled data (b) Transformed data (c) Learned transform

Figure 3: Synthetic data experiment.

	Linear	**Exponential**	**Square Root**	**Total**
Linear	**0.05**	3.25	5.55	2.95
Kernel Mixture	1.15	1.05	1.45	1.22
Greedy	0.55	0.45	0.55	0.52
Convex	0.20	**0.35**	**0.50**	**0.35**

Table 1: Percent testing error rates for the synthetic experiments.

	1 vs 2	**1 vs 3**	**1 vs 4**	**2 vs 3**	**2 vs 4**	**3 vs 4**	**Total**
Linear	20.0	12.0	4.0	26.0	15.0	20.0	16.16
Poly	19.0	12.0	4.0	25.0	14.0	18.0	15.33
RBF	14.0	12.0	4.0	18.0	14.0	17.0	13.17
x^a	6.0	**1.0**	5.0	3.0	6.0	10.0	5.17
Kernel Mixture	9.0	2.0	3.0	**1.0**	7.0	7.0	4.83
Greedy	7.0	2.0	4.0	2.0	9.0	5.0	4.83
Convex	**5.0**	2.0	**1.0**	2.0	**4.0**	**2.0**	**2.67**
Greedy Multitask	5.0	2.0	2.0	1.0	3.0	3.0	2.67
Convex Multitask	4.0	1.0	0.0	2.0	3.0	1.0	2.17

Table 2: Percent testing error rates on Corel dataset. Bold entries are the best results in the single task setting.

References

[1] O. Chapelle, P. Hafner, and V.N. Vapnik. Support vector machines for histogram-based classification. *Neural Networks, IEEE Transactions on*, 10:1055–1064, 1999.

[2] A. Howard and T. Jebara. Large margin transformation learning. *In Preparation.*

[3] A. Howard and T. Jebara. Learning monotonic transformations for classification. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20.* MIT Press, Cambridge, MA, 2008.

[4] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.