Learning via Linear Operators: Maximum Margin Regression; Multiclass and Multiview Learning at One-class Complexity

Sandor Szedmak ss03v@ecs.soton.ac.uk

John Shawe-Taylor jst@ecs.soton.ac.uk
ISIS Group
Electronics and Computer Science
University of Southampton
Southampton, U.K.

Emilio Parado-Hernandez emipar@tsc.uc3m.es
Department of Signal Processing and Communications
Universidad Carlos III de Madrid
Leganes, Spain

Editor:

Abstract

We introduce a maximum margin framework realizing a regression type learning in an arbitrary Hilbert space whilst the corresponding dual problem preserving the structure and, therefore, the complexity that of the binary Support Vector Machine(SVM). We demonstrate via some examples this learning framework is broadly applicable in several seemingly different problems. One example is the multiclass classification problem which, in this way, can be implemented with the complexity of a binary SVM. The reduction of the complexity does not involve diminishing performance but, in some cases this approach can improve the classification accuracy. The multiclass classification is realized where the output labels are vector valued. Other examples implement multiview learning problems.

Keywords: Maximum Margin Robot, Vector labels, Multiclass learning, Multiview learning

1. Introduction

Our original motivation to develop a maximum margin based regression framework comes from the multiclass classification problem which has been considered a more complex problem than the well-known implementations of binary classification. There are two main streams among the attempts to tackle these kind of problems. The first one decomposes the multiclass problem into a certain combination of binary problems, e.g. "one versus all", "one versus one" approaches built upon some kind of binary classifiers. The second one derives a regression based solution framework exploiting the multivariate capability of the Canonical Correlation Analysis and Partial Least Squares Regression, see at Rosipal and Trejo (2001), Barker and Rayens (2003) and Rosipal et al. (2003). The Support Vector

Machine seems to be a good candidate to find a maximum margin framework to solve the multiclass classification. Some approaches in Weston and Watkins (1998), Crammer and Singer (2001)and Franc and Hlavac (2002) mostly considered a classification based formulation. The authors in Tsochantaridis et al. (2005) discuss the multiclass learning as a subcase of classification of objects with special structure.

Recently Evgeniou et al. (2005), Micchelli and Pontil (2004) and Micchelli and Pontil (2005) presented a synthesis of the kernel learning approach with a general form of regression, where vector labeled output items are also learned by a machine that is an extension of the Support Vector learner. In Evgeniou et al. (2005) the complexity issue is mentioned as a weakness of the presented approach. In this paper we show that there is an implementation of this kind of maximum margin based machine with computational complexity independent of the number of classes and that it requires no more computation than a single binary Support Vector Classifier. The multiclass learning is then expressed as an application.

First, we formulate the Support Vector Machine with vector output that we call as Maximum Margin Robot (MMR). Then we present two examples, multiclass and multiview learning are presented. Following this we provide primal and dual perceptron learners for vector labels and present a Novikoff type theorem for this algorithm.

The notations that we use are summarized in Table 1. Note that we assume every mentioned Hilbert space has finite dimension and it is defined above the real numbers; furthermore there is a fixed orthogonal basis in every space, thus every object, vector and linear operator can be represented in matrix, might be high dimensional, form. When we talk about a vector we mean it is an object in a Hilbert space and, in this sense any matrix and high dimensional hyper-matrix behave as vectors.

2. Formulation of the SVM with vector output

The Support Vector Regression (SVR), described by Vapnik (1998) can be a candidate for vector label learning, but the extension of its capability is seemingly restricted. In its base formulation it uses the difference of two scalars the output value and the predictor in each constraint. Including vectors We have two possibilities to include label vectors: increase the number of the constraints up to the product of the sample size and the dimension of the label vectors or incorporate a distance function into the constraints but this kind of functions are generally nonlinear and hardly invertible. Thus, both alternatives blow up the complexity of the underlying optimization problem.

The idea underlying implementation for the vector valued Support Vector Machine stems from a simple reinterpretation of the normal vector of the separating hyperplane. We say this vector is a projection operator of the feature vectors into an one-dimensional subspace. An extension of the range of this projection into multi-dimensional subspace gives the solution for vector labeled learning.

Assume we have a sample S of pairs $\{(\mathbf{y}_i, \mathbf{x}_i) : \ \mathbf{y}_i \in \mathcal{H}_y, \ \mathbf{x}_i \in \mathcal{X}, \ i = 1, \ldots, m\}$ independently and identically generated by an unknown multivariate distribution, and an embedding of the input objects into a Hilbert space called feature space by the function $\phi : \mathcal{X} \to \mathcal{H}_\phi$.

Symbol	Explanation
$\mathcal{X}$	space of the possible input vectors,
$\mathcal{H}_\phi$	Hilbert space comprising the feature vectors, the images of the input vectors with respect to the embedding $\phi()$,
$\mathcal{H}_y$	space of the output(label) vectors,
$\mathcal{H}_\psi$	Hilbert space comprising the image of label vectors with respect to the embedding $\psi()$
$\mathbf{W}$	matrix represented linear operator projecting the feature space $\mathcal{H}_\phi$ into $\mathcal{H}_\psi$,
$\langle .,. \rangle_{\mathcal{H}_z}$, $\\|.\\|_{\mathcal{H}_z}$	inner product and norm defined in the Hilbert space $\mathcal{H}_z$,
$\mathbf{tr}(\mathbf{W})$	trace of the matrix $\mathbf{W}$,
$\mathbf{dim}(\mathcal{H})$	dimension of the space $\mathcal{H}$.
$\mathbf{x}_1 \otimes \mathbf{x}_2$	tensor product of the vectors $\mathbf{x}_1 \in \mathcal{H}_1$ and $\mathbf{x}_2 \in \mathcal{H}_2$ and it represents a linear operator $\mathbf{A} : \mathcal{H}_2 \rightarrow \mathcal{H}_1$ which acts on a vector $\mathbf{z} \in \mathcal{H}_2$ as $(\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{z} \overset{\text{def}}{=} (\mathbf{x}_1\mathbf{x}_2^T)\mathbf{z} = \mathbf{x}_1\langle \mathbf{x}_2, \mathbf{z}\rangle_{H_2}$.
$\langle \mathbf{A}, \mathbf{B} \rangle_F$	Frobenius inner product of matrix represented linear operators $\mathbf{A}$ and $\mathbf{B}$ and it is defined by $\mathbf{tr}(\mathbf{A}^T\mathbf{B})$.
$\\|\mathbf{A}\\|_F$	Frobenius norm of a matrix represented linear operator $\mathbf{A}$ and defined by $\sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F}$.
$\mathbf{A} \cdot \mathbf{B}$	element-wise(Schur) product of the matrices $\mathbf{A}$ and $\mathbf{B}$.

Table 1: Notation used in the paper

The Maximum Margin Robot a certain type of Support Vector Machine with vector output is realized on this sample by the following optimization problem

$$\begin{array}{ll}
\min & \frac{1}{2}\mathbf{tr}(\mathbf{W}^T\mathbf{W}) + C\mathbf{1}^T\boldsymbol{\xi} \\
\text{w.r.t.} & \{\mathbf{W} \mid \mathbf{W} : \mathcal{H}_{\phi(x)} \rightarrow \mathcal{H}_y, \mathbf{W} \text{ linear operator}\}, \\
 & \{\mathbf{b} \mid \mathbf{b} \in \mathcal{H}_y, \text{ bias vector}\}, \\
 & \{\boldsymbol{\xi} \mid \boldsymbol{\xi} \in \mathbb{R}^m, \text{ slack or error vector}\} \\
\text{s.t.} & \left\langle \mathbf{y}_i, (\mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) + \mathbf{b}) \right\rangle_{\mathcal{H}_y} \geq 1 - \xi_i, \ i = 1, \ldots, m, \\
 & \boldsymbol{\xi} \geq \mathbf{0}.
\end{array} \tag{1}$$

where $\mathbf{0}$ and $\mathbf{1}$ denote the vectors with components 0 and 1 respectively.

Introducing dual variables $\{\alpha_i | i = 1, \ldots m\}$ to the margin constraints and based on the Karush-Kuhn-Tucker theory we can express the linear operator $\mathbf{W}$ by the direct products of the output and feature vectors, that is

$$\mathbf{W} = \sum_{i=1}^{m} \alpha_i \mathbf{y}_i \boldsymbol{\phi}(\mathbf{x}_i)^T. \tag{2}$$

The dual then gives

$$\begin{array}{ll} \min & \sum_{i,j=1}^{m} \alpha_i \alpha_j \overbrace{\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle}^{\kappa_{ij}^{\phi}} \overbrace{\langle \mathbf{y}_i, \mathbf{y}_j) \rangle}^{\kappa_{ij}^{y}} - \sum_{i=1}^{m} \alpha_i, \\ \text{w.r.t.} & \{\alpha_i | \alpha_i \in \mathbb{R}\}, \\ \text{s.t.} & \sum_{i=1}^{m} (\mathbf{y}_i)_t \alpha_i = 0, \ t = 1, \ldots, \dim(\mathcal{H}_y), \\ & 0 \leq \alpha_i \leq C, \ i = 1, \ldots, m, \end{array} \tag{3}$$

where we can write the values of inner products in the objective as kernel items

$$\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle \langle \mathbf{y}_i, \mathbf{y}_j) \rangle = \kappa_{ij}^{\phi} \kappa_{ij}^{y}, \tag{4}$$

where κ_{ij}^{ϕ} and κ_{ij}^{y} stand for the elements of the kernel matrices for the feature vectors and for the label vectors respectively. Hence, the vector labels are kernelized as well. The synthesized kernel is the element-wise product of the input and the output kernels, an operation that preserves positive semi-definiteness.

2.1 Some remarks about Maximum Margin Robot

The meaning of the regularization term $\mathbf{tr}(\mathbf{W}^T\mathbf{W})$ in the primal problem implies we are looking for a linear operator with the smallest sum of the squared singular values.

The complexity of the dual moderately increases relative to the base SVM since the structure of the objective remains the same where we have constraints with the same content but the number of them is increased to the dimension of the output space. However, using a proper optimization technique all the constraints except the box one can be included into the objective function as a penalty term, then we need to solve only a quadratic problem over a box constraint. For most practical cases the bias $\mathbf{b}$ can be ignored implying no other constraints than the box one is included.

The formulation of the Maximum Margin Robot can be extend further realizing that the simplicity of the dual problem is still preserved if in the primal problem the left hand side of the margin constraints comprise a real valued function F being linear in $\mathbf{W}$ an approach similar to the paper of Tsochantaridis et al. (2005). To allow the regularization term in the objective to work properly we need to assume that F is a monotonic increasing function of $\mathbf{tr}(\mathbf{W}^T\mathbf{W})$.

The MMR can be an efficient base method for structural learning since it can process any abstract Hilbertian labels. The structure of the output just like the input objects can be embedded into an appropriate Hilbert space preserving most of the original properties and then the relationship between the input and output can be discovered. The embedding of the outputs implies an inversion problem which requires us to find the original structure from the predicted Hilbertian image. One approach is shown when the multiclass learning is detailed. It applies an enumeration of the possible outcomes to find the best. Obviously, it can only be implemented if the cardinality of the label set is small.

2.2 Relation to other approaches

We would like to emphasize our learning model has deep roots in recent machine learning researches. The papers of Evgeniou et al. (2005), Micchelli and Pontil (2004) and Micchelli and Pontil (2005) mentioned in the introduction gave the direction of using regression instead of extending the classification approach. Crammer and Singer (2001) applied the kernel of the output vectors and the trace minimization via Frobenius norm minimization. Tsochantaridis et al. (2005) following Taskar et al. (2003) try to integrate the maximum margin and the inversion problem of the outputs into one optimization problem. Our objective is to handle them separately keeping the computational complexity of the maximum margin problem at a low level.

3. Multiclass classification

The multiclass classification can be implemented within the framework of the MMR. Let us assume the label vectors are chosen out of a finite set $\{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_T\}$ in the learning task. The decision function predicting one of these labels can be expressed by using the predicted vector output

$$\begin{aligned} d(x) &= \arg \max_{t=1,\ldots,T} \langle \hat{\mathbf{y}}_t, \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) + \mathbf{b} \rangle_{\mathcal{H}_y} \qquad (5) \\ &= \arg \max_{t=1,\ldots,T} \sum_{i=1}^{m} \alpha_i \kappa^y(\hat{\mathbf{y}}_t, \mathbf{y}_i) \kappa^\phi(\mathbf{x}_i, \mathbf{x}) + \langle \hat{\mathbf{y}}_t, \mathbf{b} \rangle_{\mathcal{H}_y}, \end{aligned}$$

where the bias vector $\mathbf{b}$ is the corresponding Lagrangian of the constraint $\sum_{i=1}^{m} (\mathbf{y}_i)_t \alpha_i = 0,\ t = 1, \ldots, \dim(\mathcal{H}_y)$ in the dual.

Now we are able to set up a multiclass classification. Some promising versions of the label selection are:

- The label vectors are chosen as indicator vectors of the classes following the rule

$$(\mathbf{y}_i)_t = \begin{cases} 1 & \text{if item } i \text{ belongs to category } t, t = 1, \ldots, T, \\ 0 & \text{otherwise.} \end{cases} \qquad (6)$$

- Let the label vectors be defined on a unit ball. We are looking for a configuration of these vectors in which the correlation between any two distinct vectors is the same and minimized. These vectors span a hyper-tetrahedron defined in a space with dimension $T-1$. For example if $T=2$ there are two one dimensional vectors $1, -1$, if $T=3$ then there are three vectors spanning a equilateral triangle. The hyper-tetrahedron is a generalization of these shapes in a higher dimensional Euclidean space.

  This kind of configuration can be derived as follows. Let a matrix $\mathbf{A}$ with size $T \times T$ be given by

$$A_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\frac{1}{(T-1)} & \text{otherwise.} \end{cases} \qquad (7)$$

  The eigenvalue decomposition of $\mathbf{A}$ equals to $\mathbf{U}\boldsymbol{\Gamma}\mathbf{U}^T$, where $\mathbf{U}$ is the matrix of the eigenvectors and $\boldsymbol{\Gamma}$ is a diagonal matrix comprising the eigenvalues. Let $\hat{\mathbf{U}} = \mathbf{U}\sqrt{\boldsymbol{\Gamma}}$, where $\sqrt{\boldsymbol{\Gamma}}$ is evaluated component-wise. One eigenvalue of $\mathbf{A}$ is 0 so the corresponding column of $\hat{\mathbf{U}}$ can be deleted, thus we received a matrix with size $T \times (T-1)$ and satisfying $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{U}}^T$. Hence the length of column vectors is 1 and the correlations between them are the same and equal to $-\frac{1}{T-1}$. The reader can refer to Appendix A. for the proof of the correctness of this procedure.

- A simple and straightforward labeling technique can be deducted using the class centers. The mean or the median vectors of the classes arise as good candidates. They may be computed on the raw data or after a certain kind of normalization.

If the label vectors are chosen as indicator vectors given in (6), then we have T special maximum margin machines realizing a set of one class SVMs for each of the classes. This statement follows from the fact multiplying the projection matrix $\mathbf{W}$ from the left with the transpose of a label vector with only one non-zero component selects one row of $\mathbf{W}$ and this row can be considered as a normal vector of the hyperplane cutting the feature space into two parts such that one part contains the corresponding class with the smallest error. The speciality of our implementation comes from the structure of the objective function where the sum of the squared norm of the normal vectors is minimized allowing the subproblems to influence each other. In this way we can balance the occurring differences in the classification errors among the classes which might give a better overall performance.

If the indicator type label vectors from (6) are applied then the bias has to be excluded from the model, since the dual constraint contains only non-negative components of $\{\mathbf{y}_i\}$, therefore, the only feasible solution for α is $\mathbf{0}$. Hence, the separating hyperplanes are linear subspaces of the feature space.

3.1 Experiments with the multiclass learning

In order to test the MMR we used multiclass classification problems from the UCI Repository of machine learning datasets, the details are given by Blake and Merz (1998). The data sets chosen mostly correspond to those used by Rifkin and Klautau (2004) to give a well-defined benchmark environment for comparison. Table 2 shows these sets and their descriptors.

We used similar configurations to those was described in the paper of Rifkin and Klautau (2004), and a Gaussian kernel as well. The accuracies are computed in the following way

- If the original dataset is split into given training and test sets, we use them; otherwise, a 5-fold cross-validation was applied.
- The size for the Gaussian kernel was evaluated by a 5-fold cross-validation procedure applied only on the training set. The set $\{\sigma|0.001 \cdot 2^i, i = 0, \ldots, 20\}$ of candidate parameters was scanned and the value producing the best average performance on the five validation subsets was chosen.

We computed the accuracies when the input vectors are normalized by projecting them onto a unit ball, and when the input vectors are normalized component-wise by subtracting the mean and dividing with the standard deviation. In both cases we also considered indicators and vertices of a hyper-tetrahedron for output coding.

In the optimization task we dropped the bias term, hence the box constraint was only active in the dual. The solver used a simple coordinate descent method and it was implemented in pure Matlab code. It requires only one column of the kernel matrix in each step of the algorithm allowing very large problems to be processed. Table 4 demonstrates the average solution time for every dataset when the data was normalized and when was not. It shows the practical performance of MMR guarantees the efficiency in huge multiclass classifications.

	Number of			
Name	Training	Test	Classes	Numerical/
	Items	Items		Nominal attr.
abalone	3133	1044	29	8/1
glass	214	*	7	9/0
optdigits	3823	1797	10	64/0
page-blocks	5473	*	5	10/0
satimage	4435	2000	6	36/0
spectrometer	531	*	48	101/0
yeast	1484	*	10	8/0

Table 2: Parameters of the data sets used in the experiment. * denotes the datasets with no dedicated training and test subsets.

	Test error rate (%)							
	SVM		MMR					
	all	one	hyper-tetrahedron			indicator		
	vs.	all	Normalized on					
Name			—	item	variable	—	item	variable
abalone *	**72.3**	79.7	73.0	73.0	73.4	73.9	73.0	74.1
glass	30.4	30.8	27.3	27.6	29.2	**26.4**	29.0	29.0
optdigits *	3.8	2.7	2.0	**1.6**	3.3	2.1	1.9	3.3
page-blocks	3.4	3.4	4.4	3.4	3.7	4.5	3.6	**3.3**
satimage *	8.2	**7.8**	8.2	17.5	8.6	8.7	17.7	9.1
spectrometer	42.8	53.7	99.5	**37.5**	53.9	99.6	38.4	53.3
yeast	41.0	**40.3**	41.6	40.6	**40.3**	42.6	41.6	40.9

Table 3: Test error rates (%). If the data set has dedicated training and test subsets, marked with *, then the table shows the accuracy computed on the given test subset otherwise the presented accuracies are averages computed via 5-fold cross-validation.

In Table 3 the values for the methods “one versus all” and “one versus one” are borrowed from Rifkin and Klautau (2004) as well. We should emphasize that if the computational complexity of a learner is small then a systematic scanning of the parameter space for an optimal configuration remains sufficiently cheap, so that using any validation procedure better (and sometimes much better) accuracies can be achieved.

The accuracy and computational time result shows that the tetrahedron configuration of the multiclass labels performs very well and in average twice as fast as the indicator case. If there is no a prior information to make a distinction between the output labels then a highly symmetric Euclidean embedding is probably the best choice.

	Times (s)					
	MMR					
	hyper-tetrahedron			indicator		
	Normalized on					
Name	—	item	variable	—	item	variable
abalone *	0.457	0.231	0.220	0.185	0.493	0.511
glass	0.009	0.015	0.008	0.010	0.025	0.018
optdigits *	0.435	0.253	0.244	0.220	0.565	0.575
page-blocks	0.402	0.232	0.291	0.267	0.630	0.724
satimage *	0.511	0.468	0.320	0.272	0.687	0.717
spectrometer	0.016	0.019	0.012	0.009	0.050	0.028
yeast	0.155	0.051	0.047	0.042	0.115	0.111

Table 4: The average solution time of the dual problem on the training

4. Multiview learning

In the multiview learning we are given a compound sample S of pairs $\{\Big(\mathbf{y}_i, (\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{N_k})\Big) : \mathbf{y}_i \in \mathcal{H}_y,\ \mathbf{x}_i^k \in \mathcal{X}_k,\ i = 1, \ldots, m,\ k = 1, \ldots, N_k\}$ independently and identically generated by an unknown multivariate distribution, and given a set of embedding of the inputs into Hilbert spaces by the functions $\phi_k : \mathcal{X}_k \rightarrow \mathcal{H}_{\phi_k},\ k = 1, \ldots, N_k$. We may consider a setting where the input vectors $(\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{N_k})$ in one sample item are the same however the embedding functions are distinct.

This learning problem arises when there are several sources of the input vectors chosen from distinct distributions. In case of identical distributions a simple concatenation of the inputs can work, but when it is not true, source specific embedding can help.

We present two models, an additive and a multiplicative ones, to synthesize the effect of the inputs considered. For the sake of simplicity the models do not include the bias term. The additive primal model and its dual are as follows

$$
\begin{array}{lll}
 & \min & \frac{1}{2}\sum_{k=1}^{N_k} \mathbf{tr}(\mathbf{W}_k^T \mathbf{W}_k) + C\mathbf{1}^T\boldsymbol{\xi} \\
 & \text{w.r.t.} & \mathbf{W_k} : \mathcal{H}_{\phi_k} \rightarrow \mathcal{H}_y \text{ set of linear operators,} \\
\text{Primal:} & \text{s.t.} & \big\langle \mathbf{y}_i, \sum_{k=1}^{N_K} \mathbf{W}_k \boldsymbol{\phi}_k(\mathbf{x}_i^k) \big\rangle_{\mathcal{H}_y} \geq 1 - \xi_i, \\
 & & \quad i = 1, \ldots, m, \\
 & & \boldsymbol{\xi} \geq 0.
\end{array}
\tag{8}
$$

$$
\begin{array}{lll}
 & \min & \frac{1}{2}\boldsymbol{\alpha}^T(\mathbf{K}_y \cdot \sum_{k=1}^{N_k} \mathbf{K}_{\phi_k})\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\
\text{Dual:} & \text{w.r.t.} & \boldsymbol{\alpha} \in \mathbb{R}^m \\
 & \text{s.t.} & 0 \leq \boldsymbol{\alpha} \leq C.
\end{array}
$$

Here we use a multivariate linear regression approach which leads to a summation of the corresponding input kernels in the dual.

The multiplicative case and its dual reads as

$$
\begin{array}{lll}
 & \min & \frac{1}{2}\mathbf{tr}(\mathbf{W}^T\mathbf{W}) + C\mathbf{1}^T\boldsymbol{\xi} \\
 & \text{w.r.t.} & \mathbf{W} : \bigotimes_{k=1}^{N_k} \mathcal{H}_{\phi_k} \rightarrow \mathcal{H}_y \text{ linear operator,} \\
\text{Primal:} & \text{s.t.} & \left\langle \mathbf{y}_i, \mathbf{W}\left(\bigotimes_{k=1}^{N_k} \boldsymbol{\phi_k}(\mathbf{x}_i^k)\right) \right\rangle_{\mathcal{H}_y} \geq 1 - \xi_i, \\
 & & \qquad i = 1, \ldots, m, \\
 & & \boldsymbol{\xi} \geq 0. \\
 & & \\
 & \min & \frac{1}{2}\boldsymbol{\alpha}^T(\mathbf{K}_y \cdot \mathbf{K}_{\phi_1} \cdot\ \ldots\ \cdot \mathbf{K}_{\phi_{N_k}}))\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\
\text{Dual:} & \text{w.r.t.} & \boldsymbol{\alpha} \in \mathbb{R}^m \\
 & \text{s.t.} & 0 \leq \boldsymbol{\alpha} \leq C.
\end{array}
\tag{9}
$$

In this model the direct product of the inputs collects the information provided by the sources and it works in a nonlinear fashion. The dual comprises the element-wise product of the kernels. In both models the final structure of the duals are the same.

Several other formulation can be assumed to connect the input vectors, where one can combine the additive and the multiplicative models into an element-wise polynomial of the input kernels.

4.1 Experiments with multiview learning

The MMR based multiview learning has been tested on image classification. The dataset[2] is commonly used for generic object recognition, for example by Fergus et al. (2003). The three object classes in this dataset are; motorbikes, aeroplanes and faces. It also contains an additional background class to give the negative examples for each class.

For each image two sets of low level features were computed. One[3] used the affine invariant Harris detector developed by Mikolajczyk and Schmid (2001) to find interest points within an image and to compute Invariant Moments as patch descriptors. The other was introduced by Lowe (1999) as a keypoint detector[4] to recognize interesting patches with the so-called SIFT affine invariant patch descriptors. These sets of image patch descriptors form the basis of the feature generation.

Since different images have different numbers of interest points vector quantization was used to map these sets of points into a fixed length feature vector. In our experiment, k-means clustering was computed to learn K cluster centers based upon the features from all images. For each image a fixed length feature vector was then created by using a histogram corresponding to the distribution of the interest points with respect to the clusters on that image. In all the following experiments, the parameter for clustering was chosen as $K = 400$. We compared the performance of the additive and multiplicative models with the two binary SVMs trained on only one feature vector, moment or SIFT, and when these vectors concatenated one. Following the general practice in machine vision we use the receiver-operating characteristic (ROC) curve related accuracy measure, so called, Equal

[2] Available at http://www.robots.ox.ac.uk/~vgg/data/

[3] Available at $http://lear.inrialpes.fr/people/Mikolajczyk/$.

[4] Available at $http://www.cs.ubc.ca/\sim lowe/keypoints/$

	Equal Error Rate (%)					
	Ferguson	Binary SVM			MMR	
Dataset	et al.	Moment	SIFT	Concatenated	Additive	Multiplicative
Airplanes	90.2	92.4	97.0	97.3	98.0	**98.2**
Faces	96.4	98.0	96.5	**98.5**	98.2	98.2
Motorbikes	92.5	95.3	95.0	95.1	**96.5**	94.7

Table 5: Classification accuracies for the multiview learning compared with the performance of binary SVM processing feature sets separately and in concatenation.

Error Rate (EER). The EER shows the accuracy when the proportions of the false negatives and false positives are the same in the prediction. The base line values for the binary SVMs are borrowed from Meng et al. (2005).

5. Perceptron algorithm for Maximum Margin Robot

The formulation of MMR also suggests an implementation of a perceptron type algorithm for maximum margin regression. Here we aim to demonstrate the transparency of the formulation of the MMR, which allows us to inherit most of the machine learning techniques developed earlier.

Consider the optimization problem in (1) when only the error term is minimized

$$\begin{array}{ll} \min & \sum_{i=1}^{m} h(\lambda - \langle \mathbf{y}_i, \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i)\rangle_{\mathcal{H}_y}) \\ \text{subject to} & \{\mathbf{W} | \mathbf{W} : \mathcal{H}_x \rightarrow \mathcal{H}_y, \mathbf{W} \text{ a linear operator}\}, \end{array} \tag{10}$$

where λ is a prescribed margin, and the function $h(u)$ denotes the Hinge loss, that is

$$h(u) = \begin{cases} u & \text{if } u > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

The error function that we are going to minimize has subgradient with respect to $\mathbf{W}$ and this can be computed independently in an incremental way for each term occurring in the summation (10). The reader can consult to Bertsekas (1999) and Kiwiel (2004) for details of incremental subgradient methods. The term-wise subgradient is equal to

$$\partial h(\lambda - \langle \mathbf{y}_i, \mathbf{W}\phi(x_i)\rangle_{\mathcal{H}_y})|_{\mathbf{W}} = \begin{cases} -\mathbf{y}_i\phi(x_i)^T & \text{if } \lambda - \langle \mathbf{y}_i, \mathbf{W}\phi(x_i)\rangle_{\mathcal{H}_y} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We can define the learning speed with a step size, denoted by s, and we obtain the perceptron-like algorithm given in Figure 1.

The departure from the original perceptron algorithm is very moderate. Here we need to learn a matrix realizing the projection of the input vectors into the output space. The incremental subgradient based update employs the direct product of the corresponding output and input vectors to update the projection matrix.

```
**Input of the learner:** The sample $S$, step size $s$
**Output of the learner:** $\mathbf{W} \in \mathbb{R}^{dim(\mathcal{H}_y)\times dim(\mathcal{H}_x)}$
**Initialization:** $\mathbf{W}_t = \mathbf{0};\ t = 0;$
**Repeat**
  **for** $i = 1, 2, \ldots, m$ **do**
    read input: $\mathbf{x}_i \in \mathbb{R}^n;$
    **if** $\langle \mathbf{y}_i, \mathbf{W}_t \boldsymbol{\phi}(x_i)\rangle_{\mathcal{H}_y} < \ \lambda$ **then**                (13)
      $\mathbf{W}_{t+1} = \mathbf{W}_t + s\mathbf{y}_i \boldsymbol{\phi}(x_i)^T$
      $t = t + 1$
    **end if**
  **end for**
**until**
```

Figure 1: Vector perceptron algorithm

A dual version of perceptron algorithm can be derived to learn vector outputs. Assume $\mathbf{W}$ is expressible by the sample items in the form of (2) then we have the optimization problem

$$\begin{array}{ll} \min & \sum_{i=1}^{m} h(\lambda - \sum_{j=1}^{m} \alpha_j \overbrace{\langle \mathbf{y}_i, \mathbf{y}_j \rangle}^{\kappa_{ij}^{y}} \overbrace{\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle}^{\kappa_{ij}^{\phi}}) \\ \text{subject to} & \alpha_j \geq 0,\ j = 1, \ldots, m, \end{array} \tag{14}$$

The partial derivatives for $\alpha_i,\ k = 1, \ldots, m$ equals to

$$\partial h(\lambda - \sum_{j=1}^{m} \alpha_j \kappa_{ij}^{y} \kappa_{ij}^{\phi})|_{\alpha_\mathrm{i}} = \begin{cases} -\kappa_{ij}^{y}\kappa_{ij}^{\phi} & \text{if } h(\lambda - \sum_{j=1}^{m} \alpha_j \kappa_{ij}^{y} \kappa_{ij}^{\phi}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Finally the dual perceptron algorithm is formulated according to Figure 2.

An analogue of the standard Novikoff theorem provides an upper bound on the number of updates and a lower bound on the achievable margin in the primal formulation. Here we follow the derivation that was presented in Li et al. (2002).

Let us define the margin for perceptron learner as

$$\gamma(\mathbf{W}, S, \phi) = \min_{(\mathbf{y}_i, \mathbf{x}_i) \in S} \frac{\langle \mathbf{y}_i, \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i)\rangle_F}{\|\mathbf{W}\|_F}. \tag{17}$$

Then we can claim the following statement:

Theorem 1 *Let $S = \{(\mathbf{y}_i, \mathbf{x}_i)\} \subset (\mathcal{Y} \times \mathcal{X}),\ i = 1, \ldots$ be a sample independently and identically drawn from an unknown distribution and let $\phi : \mathcal{X} \to \mathcal{H}_\phi$ be an embedding into a Hilbert space, furthermore assume that $\|\boldsymbol{\phi}(\mathbf{x}_i)\| = 1$ and $\|\mathbf{y}_i\| = 1$ for all i, and that the learning rate, the step size, s is a fixed positive real number in (1).*

Suppose there exists a linear operator $\mathbf{W}^*$ such that $\|\mathbf{W}^*\|_F = 1$ and

$$\gamma(\mathbf{W}^*, S, \phi) \geq \Gamma, \tag{18}$$

and the algorithm stops when the functional margin 1 is achieved.

```
Input of the learner: The sample $S$, step size $s$,
Output of the learner: $(\alpha_j)$, $j = 1, \dots, m$,
Initialization: $\alpha_j = \mathbf{0}$; $j = 1, \dots, m$,
Repeat
  for $i = 1, 2, \dots, m$ do
    read input: $\mathbf{x}_i \in \mathbb{R}^n$;
    if $\langle \sum_{j=1}^{m} \alpha_j \kappa_{ij}^{y} \kappa_{ij}^{\phi} ) < \lambda$ then                (16)
      for $j = 1, 2, \dots, m$ do
        $\alpha_j = \alpha_j + s\kappa_{ij}^{y} \kappa_{ij}^{\phi}$
      endif
    end if
  end for
until
```

Figure 2: Dual vector perceptron algorithm

1. *Then the number of updates made by Algorithm (1) is bounded by*

$$t \leq \frac{1}{\Gamma^2} \left(1 + \frac{2}{s} \right). \tag{19}$$

2. *Then for the solution* $\mathbf{W}_t$ *of (1) we have*

$$\gamma(\mathbf{W}_t, S, \phi) \geq \frac{\Gamma}{s+2}. \tag{20}$$

Proof

1. Following the Novikoff pattern we first upper bound the norm of the matrix $\mathbf{W}_t$ obtained after t updates:

$$\begin{aligned} \|\mathbf{W}_t\|_F^2 &= \|\mathbf{W}_{t-1}\|_F^2 + 2s\langle \mathbf{y}_i \mathbf{W}_{t-1} \boldsymbol{\phi}(x_i) \rangle_{\mathcal{H}_y} + s^2 \|\mathbf{y}_i \boldsymbol{\phi}(x_i)^T\|_F^2 \\ &\leq \|\mathbf{W}_{t-1}\|_F^2 + 2s + s^2 \|\mathbf{y}_i\|^2 \|\boldsymbol{\phi}(x_i)\|^2 \\ &\leq \|\mathbf{W}_{t-1}\|_F^2 + 2s + s^2 \\ &\leq ts(s+2). \end{aligned} \tag{21}$$

We now provide a reverse inequality for the inner product with $\mathbf{W}^\star$:

$$\begin{aligned} \langle \mathbf{W}_t, \mathbf{W}^\star \rangle_F &= \langle \mathbf{W}_{t-1}, \mathbf{W}^\star \rangle_F + s \left\langle \mathbf{y}_i \boldsymbol{\phi}(x_i)^T, \mathbf{W}^\star \right\rangle_F \\ &= \langle \mathbf{W}_{t-1}, \mathbf{W}^\star \rangle_F + s \left\langle \mathbf{y}_i, \mathbf{W}^\star \boldsymbol{\phi}(x_i) \right\rangle_{\mathcal{H}_y} \\ &\geq \langle \mathbf{W}_{t-1}, \mathbf{W}^\star \rangle_F + s\Gamma \\ &\geq ts\Gamma. \end{aligned}$$

Then we can create the squeezing inequality:

$$ts(s+2)\|\mathbf{W}^\star\|_F^2 \geq \|\mathbf{W}_t\|_F^2 \|\mathbf{W}^\star\|_F^2 \geq \langle \mathbf{W}_t, \mathbf{W}^\star \rangle_F^2 \geq (ts\Gamma)^2. \tag{22}$$

implying the result.

2. Taking the bound (19) for t and substituting into (21) we arrive at

$$\|\mathbf{W}_t\|_F \leq \frac{s+2}{\Gamma}. \tag{23}$$

Then for the margin we have

$$\begin{aligned} \gamma(\mathbf{W}_t, S, \phi) &\geq \min_{(\mathbf{y}_i,\mathbf{x}_i)\in S} \frac{\langle \mathbf{y}_i, \mathbf{W}_t\boldsymbol{\phi}(\mathbf{x}_i)\rangle_F}{\|\mathbf{W}_t\|_F} && (24) \\ &\geq \frac{1}{\|\mathbf{W}_t\|_F} && (25) \\ &\geq \frac{\Gamma}{s+2}, && (26) \end{aligned}$$

which proves the statement.

■

Sparsity bounds Graepel et al. (2000) can also be used to translate this bound on the number of updates into a corresponding bound on the generalization of the resulting classifier.

6. Conclusions

In this paper we have shown an algebraic generalization of the well-known Support Vector Machine to solve regression type vector labeled learning problems. In an application of the Maximum Margin Robot we demonstrate that multiclass learning is expressible in a simple optimization framework and this sort of simplicity not only preserves the accuracy but may improve it. In another application some approaches to the multiview learning have been represented. A vectorized version of the perceptron algorithm with margin (or τ-perceptron) has been shown for which the number of updates can be bounded in terms of the optimal margin obtainable.

In further research we plan to make a similar reduction of the complexity for structural learning. The simplicity and transparency of the learning methods in this formulation can give strong support to the generalization theory as well by removing unnecessary technical complications.

An interesting and fruitful extension of our approach is to use objects in infinite dimensional Hilbert spaces, that is to learn when the input and the output are real valued functions exploiting the simplicity and finiteness of the dual problem.

Using vector outputs the maximum margin is applicable to solve regression type problems and, thus, realizes an alternative of the maximum likelihood and least square methods for several well-known statistical methods, e.g. for multivariate regression and for variance analysis.

Acknowledgments

This work was supported by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

Apendix A.

Here we illuminate the background of the method presented in Section 3

Proposition 2 *Given a set of vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_k \in \mathbb{R}^k\}$ all different from the zero vector. If the Euclidean norm of these vectors equal to 1 and the inner product of any two distinct vectors is equal to and achieves the minimum then these vectors span a $k-1$ dimensional subspace in $\mathbb{R}^k$ and the inner-products of distinct vectors equal to $-\frac{1}{k-1}$.*

Proof Let the vectors $\mathbf{x}_1, \dots, \mathbf{x}_k$ be rows of a matrix $\mathbf{X}$. The inner-products between the vectors is given by $\mathbf{C} = \mathbf{X}\mathbf{X}^T$. Because for every $i, j = 1, \dots, k$ $\|x_i\| = 1$ and $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ are the same the components of C satisfy

$$C_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -t & \text{otherwise.} \end{cases} \tag{27}$$

We can assume computing the minimum of the common value for the inner products that $t \geq 0$. It is true since $t = 0$ gives a feasible inner product matrix where the vectors constitute a normalized orthogonal basis of the space $\mathbb{R}^k$, so, we can claim t at least 0.

The matrix $\mathbf{C}$ can be written as $\mathbf{C} = (1+t)\mathbf{I} - t\mathbf{1}\mathbf{1}^T$, where $\mathbf{I}$ the k dimensional identity matrix and $\mathbf{1}$ is a vector with components 1 in $\mathbb{R}^k$. Because of the construction of $\mathbf{C}$ it is a symmetric and positive (semi)definite matrix. Let its eigenvalue decomposition equal to $\mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$ where $\mathbf{U}$ is an orthogonal matrix of the eigenvectors and $\mathbf{\Gamma}$ a diagonal matrix comprising the eigenvalues. We have the equality

$$(1+t)\mathbf{I} - t\mathbf{1}\mathbf{1}^T = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T \tag{28}$$

that we can multiply from left by $\mathbf{S}^T$ and from right by $\mathbf{S}$ which gives

$$(1+t)\mathbf{I} - t\mathbf{S}^T\mathbf{1}\mathbf{1}^T\mathbf{S} = \mathbf{\Gamma}. \tag{29}$$

Since the matrices $(1+t)\mathbf{I}$ and $\mathbf{\Gamma}$ are diagonal, thus the matrix equality (29) holds if the matrix $\mathbf{S}^T\mathbf{1}\mathbf{1}^T\mathbf{S}$ is a diagonal matrix as well. However $\mathbf{S}^T\mathbf{1}\mathbf{1}^T\mathbf{S}$ is a direct product of the vector $\mathbf{S}^T\mathbf{1}$ with its transpose therefore the rank of this matrix is at most 1. From this fact and the diagonality of this matrix we can conclude that only one diagonal component can depart from 0.

Assume first this diagonal component differs from 0. Without hurting the generality we can fix the non-zero component into the upper-left position of the matrix. The value of this non-zero component is equal to $\mathbf{1}^T\mathbf{S}\mathbf{S}^T\mathbf{1}$ which gives k. Now we have

$$(1+t)\mathbf{I} - \mathbf{\Gamma} = \begin{bmatrix} tk & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{30}$$

We can express the eigenvalues in $\mathbf{\Gamma}$ by unfolding the matrix equality (30)

$$\gamma_1 = 1 + t - tk, \tag{31}$$

$$\gamma_2 = 1 + t, \tag{32}$$

$$\vdots = \vdots \tag{33}$$

$$\gamma_k = 1 + t. \tag{34}$$

The positive semi-definiteness of $\mathbf{C}$ demands non-negative eigenvalues. γ_1 is non-negative if $t \leq \frac{1}{k-1}$. The minimization of the pairwise inner-product implies maximization of t so we can conclude $t = \frac{1}{k-1}$ and then $\gamma_1 = 0$ but because all other eigenvalues equal to $1 + \frac{1}{k-1}$ the rank is $k-1$ hence the vectors in $\mathbf{X}$ live in a $k-1$ dimensional subspace of $\mathbb{R}^k$.

If there is no non-zero diagonal component in $\mathbf{S}^T\mathbf{1}\mathbf{1}^T\mathbf{S}$ then it can happen if $\mathbf{S}^T\mathbf{1} = \mathbf{0}$ implying $\mathbf{C} = 0$ and $\mathbf{X} = \mathbf{0}$ which contradicts with our assumptions.

■

References

M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

D.P. Bertsekas. *Nonlinear Programming.* Athena Scienctific, second edition edition, 1999.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research, 6(Apr)*, pages 615–637, 2005.

R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

V. Franc and V. Hlavac. Kernel representation of the kesler construction for multi-class svm classification. In H. Wildenauer and W. Kropatsch, editors, *Proceedings of the CVWW'02.* 2002.

T. Graepel, R. Herbrich, and J. Shawe-Taylor. Generalisation error bounds for sparse linear classifiers. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 298–303. Morgan Kaufmann Publishers Inc., 2000.

K.C. Kiwiel. Convergence of approximate and incremental subgradient methods for convex optimization. *Journal of Optimization*, 14, 3:807–840, 2004.

Yaoyong Li, Hugo Zaragoza, Ralf Herbich, John Shawe-Taylor, and Jaz Kandola. The perceptron algorithm with uneven margins. In *Proceedings of the International Conference of Machine Learning (ICML'2002)*. 2002.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision, Corfu*, 1999.

H. Meng, J. Shawe-Taylor, S. Szedmak, and J.R.D. Farquhar. Support vector machine to synthesise kernels. In *Sheffield Machine Learning Workshop Proceedings, Lecture Notes in Computer Science*. Springer, 2005.

C.A. Micchelli and M. Pontil. Kernels for multi-task learning. In *Proc. of the 18-th Conf. on Neural Information Processing Systems (NIPS'04)*. 2004.

C.A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *In Computer Vision and Pattern Recognition*, volume 2, pages 96–101, 2001.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

R. Rosipal and L.J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space, 2001.

R. Rosipal, L. J. Trejo, and B. Matthews. Kernel pls-svc for linear and nonlinear classification. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003) Washington DC*. 2003.

B. Taskar, C. Guestrin, and D. Koller. Max margin markov networks. In *NIPS 2003*. 2003.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453–1484, 2005.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.