The Support Vector Decomposition Machine

Francisco Pereira[1,3] FPEREIRA@CS.CMU.EDU
Geoffrey Gordon[2] GGORDON@CS.CMU.EDU
1) Computer Science Dept. 2) Machine Learning Dept. 3) Center for the Neural Basis of Cognition, Carnegie Mellon University

Abstract

In machine learning problems with tens of thousands of features and only dozens or hundreds of independent training examples, dimensionality reduction is essential for good learning performance. In previous work, many researchers have treated the learning problem in two separate phases: first use an algorithm such as singular value decomposition to reduce the dimensionality of the data set, and then use a classification algorithm such as naïve Bayes or support vector machines to learn a classifier. We demonstrate that it is possible to combine the two goals of dimensionality reduction and classification into a single learning objective, and present a novel and efficient algorithm which optimizes this objective directly. We present experimental results in fMRI analysis which show that we can achieve better learning performance and lower-dimensional representations than two-phase approaches can.

1. Introduction

Learning problems in biomedical image analysis are often characterized by having tens of thousands of features and only a small number of labelled training examples. In the domain we are considering, functional neuroimaging, this situation arises because of the difficulty of collecting and labelling independent training examples: temporal autocorrelation in fMR image sequences is very high and it is expensive to collect more than a few image sequences.

In order to obtain good learning performance in these domains, researchers often perform dimensionality reduction before learning a classifier. Two typical approaches to dimensionality reduction are feature selection (*e.g.*, forward stepwise selection) and feature synthesis (*e.g.*, singular value decomposition). In this paper we will focus on feature synthesis.

In a typical feature synthesis approach we build features based on the distribution of the independent variables in the training data, using algorithms like singular value decomposition (SVD) or independent component analysis (ICA). Unfortunately, feature spaces produced in this manner may not necessarily be good for classification, since they are derived without reference to the quantity we are trying to predict.

For example, consider fMR images taken while a subject was thinking about items of different semantic categories, and suppose that our task is to decide which semantic category was present in each example. If we perform an SVD of this data, the components extracted will capture image variability due to awareness, task control, language use, the visual form of the cues given, and many other factors. Most of these dimensions of variability will have little information about the semantic category, and if their variance is too high they may prevent the SVD from noticing the directions of variation which would be useful for classification.

On the other hand, the basic idea of the SVD—trying to find a small set of features which accurately describe the test data—seems sound. The problem is only that the SVD performs its data reduction without paying attention to the classification problem at hand. So, we pose the question of whether we can combine dimensionality reduction and classification into a single learning algorithm.

In this paper we propose one such learning algorithm, the Support Vector Singular Value Decomposition Machine (SVDM). To design the SVDM, we combine the goals of dimensionality reduction and classification into a single objective function, and present an

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

efficient alternating-minimization algorithm for optimizing this objective. Like an SVM, the SVDM can be viewed as trading off between a classifier's hinge loss and the norm of a learned weight vector; however, instead of regularizing with the 2-norm like the SVM, the SVDM regularizes with a norm derived from the goal of reconstructing the design matrix. So, the "simple" points on the SVDM's regularization frontier correspond to classifiers which split the data across directions of high variability. We present experiments showing that we can achieve better learning performance and find lower-dimensional representations than approaches that separate dimensionality reduction and classification into independent steps.

2. Methods

2.1. Singular Value Decomposition

The goal of the singular value decomposition algorithm is to find a representation of our matrix of training data as a product of lower-rank matrices. Our dataset is a matrix of n examples (rows) with m features (columns)

$$X_{n\times m} = \begin{bmatrix} x_1(1) & x_1(2) & \dots & x_1(m) \\ x_2(1) & x_2(2) & \dots & x_2(m) \\ \dots & & & \\ x_n(1) & x_2(2) & \dots & x_n(m) \end{bmatrix} = \begin{bmatrix} x_1' \\ x_2' \\ \dots \\ x_n' \end{bmatrix}$$

The SVD approximates X as a product of two[1] lower-rank matrices,

$$X_{n\times m} \approx Z_{n\times l} W_{l\times m}$$

Here l is the rank of the approximation. W is a *basis* matrix: each of its l rows is a direction of variability of the training examples. And Z is a matrix of *coordinates*: the ith row of Z gives the coefficients necessary to reconstruct the ith training example as a linear combination of the rows of W.

More precisely, the SVD minimizes the sum of squared approximation errors: we can compute Z and W by solving the optimization problem

$$\min_{Z,W} \|X - ZW\|_{\text{Fro}}^2 \tag{1}$$

Here $\|A\|_{\text{Fro}}$ stands for the Frobenius norm of the matrix A, $\sqrt{\sum_{ij} A_{ij}^2}$.

[1] Often the SVD is written as a product of 3 matrices, $X = U\Sigma V'$, and constraints are imposed on U, Σ, and V to make the solution unique. It is easy to convert back and forth between the two representations if desired.

2.2. Classification

We will suppose that there are $k \geq 1$ classification problems which we wish to solve. The target labels for these problems are given in the matrix $Y_{n\times k}$, with $y_{i,j} \in \{-1, 1\}$. The reason for allowing k classification problems instead of just one is that one problem may give information about features which are useful for solving another (this is a form of Multitask Learning (Caruana, 1997)).

To solve these classification problems using the learned low-dimensional representation from the SVD, we can seek parameters $\Theta_{l\times k}$ such that the matrix $\text{sgn}(Z\Theta)$ is a good approximation to Y. Here $\text{sgn}(\cdot)$ is the componentwise sign function, so for example if some element of Y is 1 we want the corresponding element of $Z\Theta$ to be positive. For convenience, we will constrain all the entries in the first column of Z to be 1, so that the corresponding entries in the first row of Θ will act as biases for the k classification problems.

2.3. Optimization Problem

As discussed above, the SVD computes Z and W without reference to Y or Θ. To improve on the SVD we will simultaneously search for values of Z, W, and Θ which minimize the following objective:

$$\|X - ZW\|_{\text{Fro}} + \sum_{i=1:n, j=1:k} h(\rho_{ij}, \mu, D)$$

where $\rho_{ij} = y_{ij} Z_{i,:}\Theta_{:,j}$. Here h is the hinge loss function with slope $-D$ and breakpoint μ:

$$h(\rho, \mu, D) = \begin{cases} 0 & \rho \geq \mu \\ D(\mu - \rho) & \text{otherwise} \end{cases}$$

This objective trades off reconstruction error (the first term) with a bound on classification error (the second term). The parameter D controls the weight of the hinge loss term, and the parameter μ controls the target classification margin.

The relative norms of Z, W, and Θ are not constrained by the above objective; for example, if we multiply Θ by some constant λ, we can compensate by dividing Z by λ and multiplying W by λ. So, we will impose the arbitrary constraints

$$\begin{aligned} Z_{i,1} &= 1 \\ \|Z_{i,2:end}\|_2 &\leq 1 \\ \|\Theta_{:,j}\|_2 &\leq 1 \end{aligned} \tag{2}$$

to pick one solution out of the many possible ones. We have constrained $Z_{i,1}$ to be 1 for two different reasons:

the first is to make $\Theta_{1,j}$ be a bias term for the jth linear discriminant, and the second is to make the first row of W correspond to the mean of the dataset.

The SVDM optimization problem is strongly related to both the SVD and the SVM. We will make part of this relationship more formal in Section 2.5; but, to gain an intuition, we can examine what happens when we set D to its extreme values. As D rises the objective will be dominated by the hinge loss term. Thus, the problems of finding Z and Θ will be similar to linear SVM problems. On the other hand, if we set D to 0, the hinge term vanishes, leaving only the Frobenius norm term. The resulting optimization is identical to the SVD problem (1), save for the constraints (2). The only one of these constraints that makes a difference is $Z_{i,1} = 1$; its effect is equivalent to centering the data by subtracting its mean before performing the SVD.

2.4. Optimization Procedure

We can solve the SVDM optimization problem by iterating the following steps: minimize the objective with respect to the variables W while holding Z and Θ fixed, then minimize with respect to Θ with Z and W fixed, then finally minimize with respect to Z with W and Θ fixed. Holding two of the three matrices fixed at each step of the optimization procedure simplifies the optimization problem in different ways, described in the following subsections. Because each step reduces the overall objective value, and because the objective is positive, this alternating minimization procedure will cause the objective value to converge to a local minimum or saddle point.

Each minimization problem was solved using either our own Matlab implementation or the SeDuMi optimization package (`http://sedumi.mcmaster.ca`). We stopped when an iteration (a set of three minimizations, one each with respect to W, Z, and Θ) yielded less than an 0.1% decrease in the objective. For the experiments described below, this translated into between 1 and 20 iterations.

2.4.1. Given Θ and Z, solve for W

As there is only one term involving W, we want to minimize

$$\|X - ZW\|_{\text{Fro}}$$

with no additional constraints. That is, we wish to predict X as a linear function of Z using coefficients W, minimizing the sum of squared prediction errors. We can find the best W by solving a linear regression problem for each column of X:

$$X_{:,j} \approx ZW_{:,j}$$

For stability we can add a tiny ridge term to each regression (Hastie et al., 2001). It would be conceptually simple to add external constraints on the rows of W, such as sparsity or spatial smoothness, without affecting the other subproblems.

2.4.2. Given W and Z, solve for Θ

Since W and Z are fixed, we can drop the first part of the objective as well as the constraints that don't involve Θ. The rest of the problem is then

$$\begin{array}{rll} \min_{\Theta,\rho_{ij}} & \sum_{ij} h(\rho_{ij}, \mu, C) & \\ \text{subject to} & \rho_{ij} = y_{ij} Z_{i,:}\Theta_{:,j} & i = 1 \ldots n,\ j = 1 \ldots k \\ & \|\Theta_{:,j}\|_2 \le 1 & j = 1 \ldots k \end{array}$$

This optimization problem tells us to choose Θ so that $\text{sgn}(Z_{i,:}\Theta_{:,j})$ is a good predictor of Y_{ij}; that is, it is a linear threshold classification problem. We can divide the optimization into k subproblems, each of which tries to find weights $\theta = \Theta_{:,j}$ which predict the jth column of Y from the features Z:

$$\begin{array}{rll} \min_{\theta,\rho_i} & \sum_i h(\rho_i, \mu, C) & \\ \text{subject to} & \rho_i = y_{ij} Z_{i,:}\theta & i = 1 \ldots n \\ & \|\theta\|_2 \le 1 & \end{array}$$

Since the hinge loss h is piecewise linear and convex, we can replace each term $h(\rho_i, \mu, C)$ in the objective by a variable h_i and the additional constraints

$$\begin{array}{rcl} h_i & \ge & 0 \\ h_i & \ge & D(\mu - \rho_i) \end{array}$$

and hence the final problem is

$$\begin{array}{rll} \min_{\theta,\rho_i,h_i} & \sum_i h_i & \\ \text{subject to} & \rho_i = y_{ij} Z_{i,:}\theta & i = 1 \ldots n \\ & h_i \ge 0 & i = 1 \ldots n \\ & h_i \ge D(\mu - \rho_i) & i = 1 \ldots n \\ & \|\theta\|_2 \le 1 & \end{array}$$

This problem is similar to a standard SVM optimization, but not identical: it has a constraint $\|\theta\| \le 1$ instead of a penalty proportional to $\|\theta\|_2^2$. See Section 2.5 for a more detailed comparison.

2.4.3. Given Y and Θ, solve for Z

Z is the hardest variable to minimize over, since it appears in both terms of the objective We can divide the optimization for Z into n subproblems, one per example. The ith subproblem is to predict the ith row of Y using the ith row of Z as adjustable weights:

$$Y_{i,:} \approx \text{sgn}(Z_{i,:}\Theta)$$

The Frobenius norm term in the objective is quadratic in Z; if it were just $\|Z\|_{\text{Fro}}^2$ we would have essentially a standard SVM again, but instead we have shifted and scaled the quadratic so that it is more expensive to increase Z along a direction that hurts reconstruction accuracy.

As in Section 2.4.2 we can eliminate $h(\rho_{ij}, \mu, C)$ from the objective by adding variables h_j with appropriate constraints. With this replacement, the ith problem is to find the $\zeta = Z_{i,:}$ which solves

$$\begin{array}{lll} \min_{\zeta,\rho_j,h_j} & \|X_{i,:} - \zeta W\|_{\text{Fro}}^2 + \sum_j h_j & \\ \text{subject to} & \rho_j = y_{ij}\zeta\Theta_{:,j} & j = 1\ldots k \\ & h_j \geq 0 & j = 1\ldots k \\ & h_j \geq D(\mu - \rho_j) & j = 1\ldots k \\ & \zeta_1 = 1 & \\ & \|\zeta_{2:end}\|_2 \leq 1 & \end{array}$$

2.5. Relationship to the SVM

In this section we compare our procedure for optimizing Θ (Section 2.4.2) to a standard linear SVM. We show that the the two optimization problems are highly similar: given appropriate parameter settings (which will not in general be known in advance), they will have equivalent solutions. We believe that it is possible to reformulate the SVDM so that the optimizations for Θ and Z become even more similar to standard SVM problems; we plan to experiment with such reformulations in future work.

The usual SVM formulation for solving a classification problem $Y_i \approx \operatorname{sgn} Z_{i,:}\,\theta$ for a vector of parameters θ is

$$\begin{array}{lll} \min_{\theta,\epsilon_i} & \|\theta\|_2^2 + Q\sum_i \epsilon_i & \\ \text{subject to} & (Y_i Z_{i,:}) \cdot \theta \;\geq\; 1 - \epsilon_i & i = 1\ldots n \\ & \epsilon_i \geq 0 & i = 1\ldots n \end{array}$$

Suppose that the optimal solution to the above problem is θ_{opt}. (θ_{opt} is unique, since the objective is strictly convex in θ.) Write $k = \|\theta_{\text{opt}}\|_2$. If we change variables to $\phi = \frac{\theta}{k}$ and $h_i = \frac{\epsilon_i}{k}$, we get the equivalent problem

$$\begin{array}{lll} \min_{\phi,h_i} & k^2\|\phi\|_2^2 + kQ\sum_i h_i & \\ \text{subject to} & (Y_i Z_{i,:}) \cdot \phi \;\geq\; 1/k - h_i & i = 1\ldots n \\ & h_i \geq 0 & i = 1\ldots n \end{array}$$

We know that this problem has an optimal solution ϕ_{opt} with $\|\phi_{\text{opt}}\|_2 = 1$ (namely, $\phi_{\text{opt}} = \theta_{\text{opt}}/k$). So, adding the constraint

$$\|\phi\|_2 \leq 1$$

doesn't change the solution. But, with this constraint, the first term in the objective ($k^2\|\phi\|_2^2$) becomes unnecessary: this term favors values of ϕ with smaller norm, but we already know reducing $\|\phi\|_2$ below 1 would result in a suboptimal solution. So, we can drop this term from the objective, leaving $\min kQ\sum_i h_i$.

Now, setting $\mu = \frac{1}{k}$ and $D = 1$ and dividing the objective by the constant kQ (which leaves the optimal solution $\phi_{\text{opt}} = \theta_{\text{opt}}/k$ unchanged), we obtain

$$\begin{array}{lll} \min_{\phi,h_i} & \sum_i h_i & \\ \text{subject to} & h_i \;\geq\; D(\mu - (Y_i Z_{i,:}) \cdot \phi) & i = 1\ldots n \\ & h_i \geq 0 & i = 1\ldots n \\ & \|\phi\|_2 \leq 1 & \end{array}$$

This optimization problem is the same as the one described in Section 2.4.2.

In other words, we have just shown that for any value of the SVM regularization parameter Q, there are corresponding values of the SVDM parameters μ and D which result in an equivalent optimal solution. (By equivalent, we mean that the solutions have the same direction but possibly different norms, resulting in the same classification boundary.)

2.6. Related Work

The papers that are most closely related to this work are (Weinberger et al., 2005) and (Globerson & Roweis, 2005). Both introduced algorithms for learning a Mahalanobis distance metric for use in nearest-neighbour classification, differing in the objective used and its relation to the classification error. This is in contrast with other metric learning methods insofar as it optimizes directly for classification instead of other related criteria (e.g. the clustering measure in (Xing et al., 2002)). Our work is more related to (Globerson & Roweis, 2005) in that the metric introduced there can be used to do a linear projection of the data into a low dimensional space, much as we aim to do by finding a basis and then reducing each example vector to its coordinates in that basis.

We depart from that work in that we minimize a hinge loss function on the reduced dimensionality space, hence doing a form of support vector machine classification rather than nearest neighbour. Our work differs also from a typical support vector machine in that it adds regularization via the reconstruction error using the low dimensional coordinates of the examples in the basis learnt. This approach was inspired by the maximum-margin matrix factorization described in (Srebro, 2004).

Guyon et al. (2002) suggested that the weights of a successful linear discriminant can provide a better indication of a feature's relevance for classification than can criteria that rely on the feature by itself. Our work shares this intuition. However, instead of looking at the original feature space directly, we learn a classifier on a low-dimensional subspace and propagate the weights back to the original space. Another difference between our work and that of Guyon et al. is that they wish to select a few features by eliminating correlated ones, while we wish to combine and summarize the collective activity of correlated features rather than eliminating them. This difference allows us to provide a visualization which displays the information contained in multiple features.

3. Experiments

3.1. Datasets

To evaluate the SVDM, we tested it on data from an fMRI experiment. In this experiment, the subject observes a word displayed on a screen for 3 seconds, followed by 8 seconds of a blank screen. Each word describes either a type of tool or a type of building, and the subject's task is to think about the word and its properties while it is displayed. During an experiment the task repeats 84 times, and a 3D image of the fMR signal is acquired every second. Each image contains $64 \times 64 \times 16 = 65536$ voxels, but only approximately 16000 of those contain cortex, hence we only consider this latter number as features (for more details about fMRI please refer to (Mitchell et al., 2004)). The dataset thus contains 84 examples; each example is the average image during a 4 second span while the subject is thinking about a word shown a few seconds earlier. The classification task is to decide which of the two semantic categories, tool or building, the subject was thinking about. We trained three separate SVDMs, one per experimental subject.

Figure 1 shows, for each of the two categories, one slice of activation in the temporal cortex of a subject, overlaid on the corresponding structural image. The data for this figure comes from another experiment, where the task was done many times in a row and all the images acquired during that period were averaged. With the reduction in noise due to averaging it is easy to see a difference between Tools and Buildings; our interest is to do the same for the current dataset, which is noisier since there was less averaging. That is, we wish to decode the "cognitive state" (Mitchell et al., 2004) of the subject from a brief interval of fMRI data.

We also tested the SVDM on synthetic datasets created by generating random matrices Z, W and Θ and using them to produce X and Y matrices for training and test sets ($X_{\text{train}} = Z_{\text{train}}W$ with $N(0,1)$ noise added to every entry, and $Y_{\text{train}} = \text{sgn}(Z_{\text{train}}\Theta)$ with a given percentage of the labels flipped to introduce error). Performance on these datasets was excellent, better than on real datasets for comparable numbers of training examples; hence these results are not included in order to leave room for discussion of the real data.

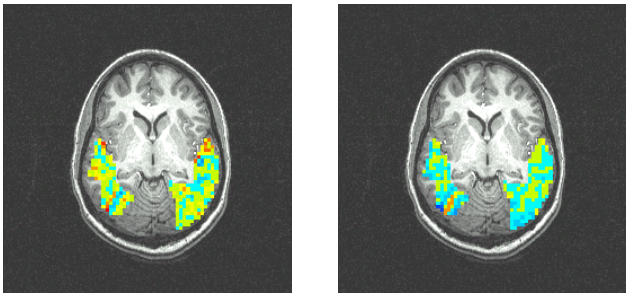

Figure 1. 2D slices from the average 3D fMRI image acquired while the subject was thinking of either "Tools" (left) or "Buildings" (right) many times in a row (more red/dark means more active)

3.2. Parameters

For a given dimensionality l (that is, Z has one column of 1s and l other columns) we ran a 6-fold cross-validation, corresponding to a natural division of the dataset into 6 cycles of task performance by a subject. Within the training set in each fold, we ran a secondary cross-validation to set the value of D. The D picked was the lowest one that produced an accuracy slightly below 100% on the secondary training sets. We always left μ fixed at 1. We tried $\mu = 0.5$ (which allows a smaller margin for each example) but found that the results were systematically worse.

3.3. Classification Experiments

We ran a comparison of several combinations of classifier, dimensionality reduction method, and number of dimensions used. We learned features using either SVDM, SVD or Independent Component Analysis (ICA) (Hastie et al., 2001), and trained a classifier using Gaussian Naïve Bayes (GNB) (Mitchell et al., 2004), a linear Support Vector Machine (linearSVM, using `libSVM` (Chang & Lin, 2001)), or an SVDM. The SVDM always learns both a set of features and a classifier; in a combination such as GNB+SVDM, we discarded the SVDM classifier and trained a GNB classifier on SVDM's features. We varied the number

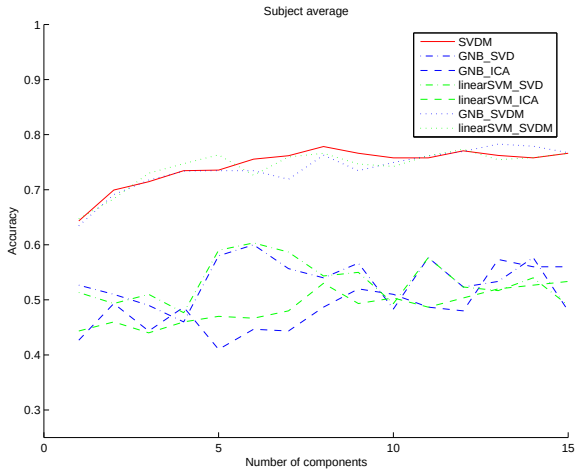

Figure 2. Classification accuracy of SVDM against other classifiers/dimensionality reduction methods, for subjects A, B and C, averaged over three subjects.

of features learned from 1 to 15 for the SVDM, and from 1 to 50 for the other dimensionality reduction methods.

Figure 2 displays the results. Each line shows the accuracy of one method averaged over all subjects; the numbers for SVDM are in addition averaged over 10 restarts of the algorithm per subject using different random seeds. (We obtained virtually the same result for all seeds.) Error bars are omitted to reduce visual clutter; however, the performance difference between the SVDM-based methods and the non-SVDM-based methods is statistically significant, while differences within either of the two groups of methods are generally not significant.

Table 1 shows the best accuracies obtained using each combination of classifier and dimensionality reduction method, together with the number of features used, again averaged over subjects. One point to note here is the rather good score obtained using linearSVM with all the voxels in the image. This is partly due to one of the subjects scoring much higher (82) than the other two (70/75); the SVDM's scores are more consistent. The high score of linearSVM indicates that the SVM's built-in regularization is helping it learn a reasonable discriminant despite the high dimensionality of the feature space; to evaluate the quality of the regularization imposed by different methods, in the next section we will examine the learned discriminants.

Table 1. Best accuracies obtained using each combination of classifier and dimensionality reduction method, together with the number of features used, averaged over three subjects.

classifier+method	accuracy	# components
SVDM	78	8
GNB+SVD	60	6
GNB+ICA	66	40
linearSVM+SVD	60	6
linearSVM+ICA	71	50
GNB+SVDM	78	13
linearSVM+SVDM	77	12
GNB	66	all voxels
linearSVM	76	all voxels

3.4. Decompositions Learnt

In order to interpret the learned discriminants, we mapped them back into equivalent discriminants on the original feature space, then plotted them to see which voxels they assigned the highest weights to. An example is shown in Figure 3. The four slices depict the size of the weight at each voxel for discriminants learnt with SVDM, GNB, or linearSVM. Each discriminant was scaled to have norm 1 prior to taking the absolute value, hence the weight magnitudes are comparable. The three discriminants all place weight in locations that have previously been identified as relevant to similar tasks, including the parahippocampal gyrus and the precentral gyrus (Kanwisher, 2003) among others (note that the Kanwisher study used picture rather than text stimuli). The difference between the SVDM discriminant and the other two is that it places more weight in those locations, while placing very little weight elsewhere.

4. Discussion and Conclusions

The experimental results in Figure 2 demonstrate that we can achieve better accuracy by learning a low-dimensional representation and a classifier simultaneously than we can by learning the two separately. The SVDM either achieves the best accuracy among competing dimensionality reduction methods, or achieves comparable accuracy using fewer components. Competing dimensionality reduction methods also seem to show fluctuations in accuracy when we add or remove even a single feature.

Moreover, the low-dimensional representation learnt with SVDM is more informative about the variable being predicted than that produced by SVD or ICA (compare the accuracies of GNB and linearSVM trained on the SVDM representation (GNB_SVDM,

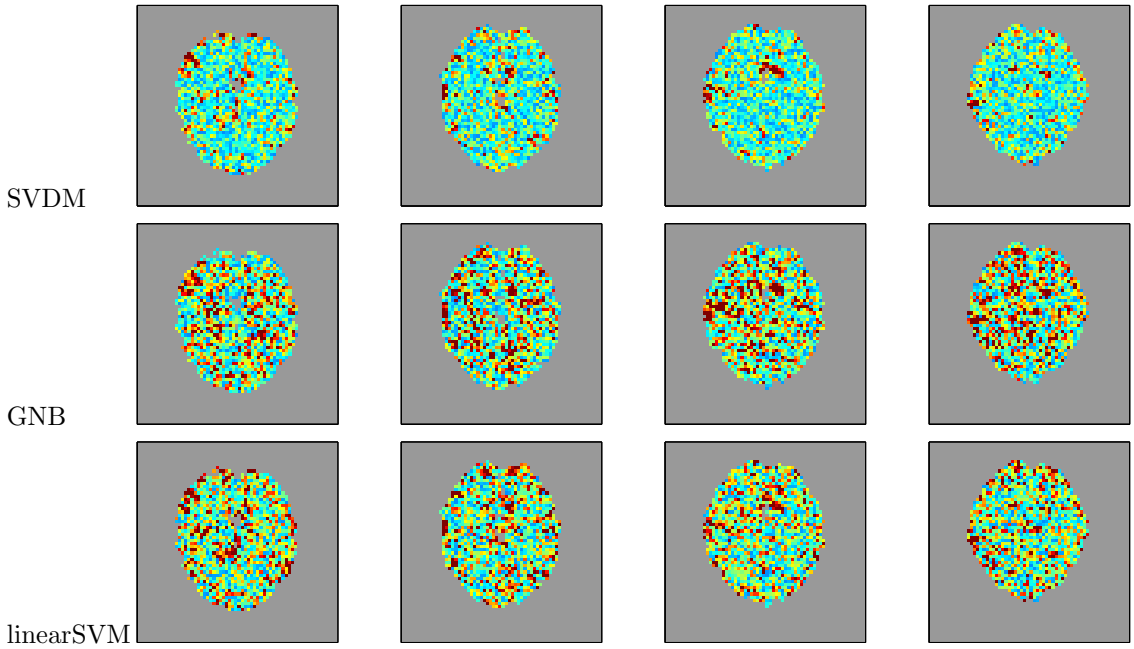

Figure 3. The four slices shown in each row depict the absolute value of discriminant weights at each voxel, for a discriminant learnt with each classifier on the same typical dataset. Slices start at the height of temporal cortex (left) and go from inferior to superior, with the back of the head at the top of each slice. Darker shades of red/gray correspond to higher values.

linearSVM_SVDM) against those obtained training on ICA/SVD (GNB_SVD, GNB_ICA, linearSVM_SVD, linearSVM_ICA)). The accuracy of the classifiers learnt on SVDM features tracks the SVDM results very closely, indicating that the features learnt, rather than the classifier, are responsible for the accuracy.

The optimization problem proposed is non-convex (Boyd & Vandenberghe, 2004). However, in practice the SVDM converges to the same solution regardless of the random matrices it is initialized with, hence it may turn out that the non-convexity is not a serious issue. This is a topic we plan to investigate further.

In related work (Pereira et al., 2006), we selected a few hundred voxels using an accuracy measure computed over a small region around each voxel. The best-scoring combinations of features yielded accuracies around 80% for our three subjects. The voxels thus identified are in the same regions that the SVDM's discriminant assigns the most weight to. This, together with the relative noisiness of the SVDM's discriminant (it is less noisy than the alternatives), leads us to think that the algorithm would fare better with priors favouring spatial smoothness and sparsity over voxels. It is conceptually easy to add such priors in the step where the W matrix is learnt, though not so simple in practice and thus work in progress.

Finally, upcoming datasets will have words from several different semantic categories. There are reasons to believe the spatial pattern of activation over temporal cortex is structured according to certain semantic features, and that different categories have different involvements of those features, while at the same time related categories will share some of them (Hanson et al., 2004). We expect that the ability to learn components useful for multiple classification problems will allow us to find components corresponding to semantic features and further constrain the learnt components.

5. Further work

We are currently working on a version of the algorithm where sparsity and smoothness constraints are added in the basis matrix W learning step. Simultaneously, we are developing an alternative formulation of the optimization problem that makes the subproblems in Section 2.4 become canonical SVM problems, allowing the use of off-the-shelf packages and the tackling of larger problems.

We also intend to run a comparison between classification results and weightings over features in the dis-

tance metric methods referred to in Section 2.6 and our own results and learnt components/discriminants. Amir Globerson has kindly run the method described in (Globerson & Roweis, 2005) over the data for a single subject and found that the classification accuracy was comparable to our results, for the most favourable setting of parameters of his method.

Finally, it has been brought to our attention by Nathan Srebro that a relaxation of this problem (with the dimensionality constraints on Z, W and Θ replaced by trace norm penalties) is a convex problem expressible as a semidefinite program, albeit possibly not tractable for the solvers we currently use. We shall pursue this direction as well.

Acknowledgments

The authors would like to thank Tom Mitchell, Amir Globerson and Nathan Srebro for useful discussions and feedback. This work was supported in part by NSF grant EF-0331657. Francisco Pereira was funded by a PRAXIS XXI scholarship from Fundação para a Ciência e Tecnologia, Portugal (III Quadro Comunitário de Apoio, comparticipado pelo Fundo Social Europeu), a PhD scholarship from Fundação Calouste Gulbenkian, Portugal, an NSF training grant from the Center for the Neural Basis of Cognition, NSF and NIH under the CRCNS program and a grant from the Keck Foundation."

References

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

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Globerson, A., & Roweis, S. (2005). Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*.

Guyon, I., Weston, J., Barnhill, S., & Vapnik., V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*, 389–422.

Hanson, S. J., Matsuka, T., & Haxby, J. V. (2004). Combinatorial codes in ventral temporal lobe for object recognition: Haxby(2001) revisited: is there a 'face' area? *Neuroimage*, *23*.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag.

Kanwisher, N. (2003). The ventral visual object pathway in humans: Evidence from fmri. In L. Chalupa and J. Werner (Eds.), *The visual neurosciences*. MIT Press.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., , & Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, *57*, 145–175.

Pereira, F., Mitchell, T., Mason, R., Just, M., & Kriegeskorte, N. (2006). "spatial searchlights for feature selection and classification of functional mri data". *to appear in the proceedings of the* 12^{th} *Conference on Human Brain Mapping*.

Srebro, N. (2004). *Learning with matrix factorizations*. Doctoral dissertation, Department of Electrical Engineering and Computer Science, MIT.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbour classification. *Advances in Neural Information Processing Systems*.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems*.