The Support Vector Decomposition Machine

Francisco Pereira[1] and Geoff Gordon[2], Computer Science Department and Center for the Neural Basis of Cognition[1], Center for Automated Learning and Discovery[2], Carnegie Mellon University, Pittsburgh, PA 15213, fpereira@cs.cmu.edu, ggordon@cs.cmu.edu

Introduction

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 a support vector machine (SVM) to learn a classifier. Instead, we combine the two goals of dimensionality reduction and classification into a single learning objective, and present the Support Vector Decomposition Machine (SVDM), a novel and efficient algorithm which optimizes this objective directly. 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 experimental results in fMR image analysis which show that the SVDM can achieve better learning performance and lower-dimensional representations than two-phase approaches can.

Algorithm

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

$$X_{n\mathtt{x}m} = \begin{bmatrix} x_1(1) & x_1(2) & \dots & x_1(m) \\ x_2(1) & x_2(2) & \dots & x_2(m) \\ \dots & & & \\ x_n(1) & x_2(2) & \dots & x_n(m) \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ \dots \\ x'_n \end{bmatrix}$$

The SVD approximates X as a product of two rank-l matrices, $X_{n\mathtt{x}m} \approx Z_{n\mathtt{x}l} W_{l\mathtt{x}m}$. More precisely, the SVD minimizes the sum of squared approximation errors: Z and W solve the optimization problem

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

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 $Y_{n\mathtt{x}k}$, with $y_{i,j} \in \{-1, 1\}$. To solve these classification problems using the learned low-dimensional representation from the SVD, we can seek parameters $\Theta_{l\mathtt{x}k}$ such that the matrix

$$f(Z\Theta) = \hat{Y}$$

is a good approximation to Y, where f is a linear threshold function. We will constrain the entries in the first column of Z to be 1, so that the first row of Θ is a vector of bias weights.

To improve on the above SVD-SVM combination 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 we have defined $\rho_{ij} = y_{ij}\xi_{ij}$ and $\xi_{ij} = \hat{Z}_{i,:}\hat{\Theta}_{:,j}$, and h is a hinge loss function

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

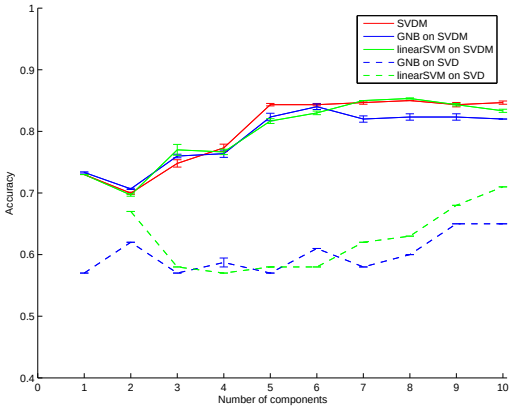

Figure 1: Accuracy of classifiers trained on low-dimensional representations.

D and μ are positive parameters, with $\mu D \geq 1$.

This objective function trades off reconstruction error (the first term) with an upper bound on classification error (the second term) using the parameter D. The relative sizes of Z, W, and Θ are not constrained by the above objective; so, we will impose the arbitrary constraints $\|Z_{i,:}\|_2 \leq 1$ and $\|\Theta_{:,j}\|_2 \leq 1$ on the norms of the rows of Z and the columns of Θ to pick one solution out of the many possible ones.

We can gain some intuition for the regularization parameter D by examining the three optimization problems which result when we solve for one set of parameters while holding the other two sets fixed. The optimization for W is unaffected by D, and just tends to force W to be as good a basis as possible for the rows of X. The optimization for Θ is essentially a standard SVM, and in this problem D is a rescaling of the usual SVM regularization parameter.

The most interesting case is when we solve for Z while holding W and Θ fixed. In this case D controls the relative weighting of the hinge loss and the $\|X - ZW\|_{\text{Fro}}^2$ term. This term 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.

Results and Discussion

We tested the SVDM on data from a functional magnetic resonance imaging (fMRI) experiment. This data consists of 84 training and 84 test examples, each of which is the average fMR image during a 4 second span, and contains approximately 16000 voxels. A few seconds prior to recording each example, the subject was shown a word on a screen, either the name of a tool or the name of a type of building. The classification task is to decide which of the two semantic categories was shown. In the training examples the words were presented in English while in the testing examples the words were presented in Portuguese.

We ran a comparison of several combinations of dimensionality reduction methods and classifiers: we learned features using either SVD or SVDM, and trained a classifier using Gaussian Naïve Bayes (GNB), a linear Support Vector Machine (linearSVM), or SVDM. (SVDM always learns both a set of features and a classifier; in a combination such as GNB on SVDM, we discarded the SVDM classifier and trained a GNB classifier on SVDM's features.) We varied the number of features learned from 1 to 10. The results are shown in Figure 1 and, in our view, allow us to conclude

- Learning a low-dimensional representation and a classifier simultaneously can result in better accuracy than learning the two separately.
- The lower-dimensional representation learnt with SVDM is more informative about the variable being predicted than that produced by SVD. This can be concluded by comparing the accuracies of GNB or SVMlinear when trained on the SVD and SVDM representations.