The Support Vector Decomposition Machine: a New Formulation

Francisco Pereira Geoff Gordon

Abstract

Recently, the authors introduced the Support Vector Decomposition Machine (SVDM), a learning algorithm which combines the two goals of dimensionality reduction and classification into a single optimization problem. The SVDM inherits the benefits of both the Singular Value Decomposition and the Support Vector Machine: it achieves the interpretability of the SVD along with the classification accuracy of the SVM. However, the original formulation of the SVDM required solving a general convex program; it was therefore tricky to implement, and could only handle small training sets. In the current paper we introduce a new formulation of the SVDM which works by calling a standard SVM optimizer as a subroutine. This new formulation can take advantage of efficient SVM optimizers to handle large training sets, and is simpler to implement because it does not require solving a general convex program. We present experiments showing that the new formulation achieves performance similar to that of the original formulation on an fMR brain image interpretation problem.

1 Background

Our dataset is a matrix X of n examples (rows) with m features (columns). We have $k \geq 1$ classification problems, with label matrix $Y \in \mathbb{R}^{n \times k}$; each label y_{ij} is in $\{-1, +1\}$. We wish to find matrices $Z \in \mathbb{R}^{n \times l}$, $W \in \mathbb{R}^{l \times m}$, and $\Theta \in \mathbb{R}^{l \times k}$ such that

$$X \approx ZW \qquad Y \approx \operatorname{sgn}(Z\Theta)$$

A singular value decomposition would pick Z and W to minimize the sum-squared reconstruction error,

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

And, given a feature matrix Z, a support vector machine would pick the matrix Θ which minimizes

$$D\|\Theta\|^2_{\text{Fro}} + \sum_{ij} h(\rho_{ij})$$

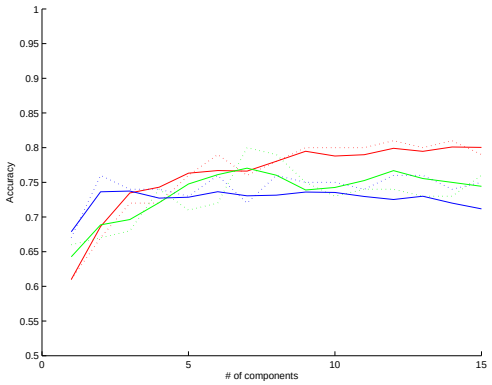

Figure 1: Accuracy of new (solid) vs. old (dashed) formulation of SVDM on 3 subjects (different colors) using 1-15 components.

where D is a parameter, $h = \max\{1 - r, 0\}$ is the hinge loss, and ρ is the matrix of scaled margins

$$\rho = Y \star Z\Theta$$

Here $\star$ stands for componentwise multiplication, so that ρ_{ij} is the scaled margin for example i and label j. (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 The SVDM

The original formulation [1] of the SVDM sought the matrices Z, W, and Θ which minimized

$$\|X - ZW\|^2_{\text{Fro}} + C \sum_{ij} h(\rho_{ij})$$

subject to the constraints

$$Z_{i,1} = 1 \qquad \|Z_{i,2:\text{end}}\| \leq 1 \qquad \|\Theta_{:,j}\| \leq 1$$

for examples i and labels j. C controls the tradeoff between hinge loss and reconstruction error. This optimization problem is not globally convex; but it

is convex in each of Z, W, and Θ with the other two matrices held fixed. So, we can efficiently find a local optimum by minimizing with respect to Z, W, and Θ in turn. In experiments, this alternating minimization algorithm converges reliably to a good local minimum in a moderate number of iterations.

The SVDM optimizations are similar to SVM optimizations, but not identical. To take advantage of efficient SVM optimization software, we propose a modification to the SVDM objective: we will search for Z, W, and Θ which simultaneously minimize

$$\frac{1}{k_X}\|X - ZW\|^2_{\text{Fro}} + \frac{1}{k_H}\sum_{\substack{i=1:n \\ j=1:k}} h(\rho_{ij}) + D\left(\frac{1}{k_Z}\|Z\|^2_{Fro} + \frac{1}{k_\Theta}\|\Theta\|^2_{Fro}\right)$$

subect to

$$Z_{i,1} = 1 \qquad \forall i$$

This optimization is similar to the original SVDM optimization, but instead of imposing constraints on the norms of Z and Θ, it adds quadratic penalties on these norms to the objective. The parameter D controls the weight of the parameter-norm term, while the weight of the reconstruction error term can be controlled implicitly by scaling the input matrix X. The constants k_X, k_H, k_Z and k_Θ are the sizes of the respective matrices (e.g., $k_X = nm$). The constraint $Z_{i,1} = 1$ makes $\Theta_{1,j}$ a bias term for the jth linear discriminant, and makes the first row of W correspond to the mean of the rows of X. With this new objective, we can now solve for Z, W, or Θ using off-the-shelf software. W is simplest, needing only a linear regression. Finding Θ with Z and W fixed is now a standard SVM problem, trading a hinge loss term against a squared-norm term. Z is most complicated: to use a standard SVM solver, we must change variables so that the quadratic penalty term is spherically symmetric and centered (details in the long version of this paper).

3 Experiments

To demonstrate that the two formulations of the SVDM have similar accuracy, we tested them on the fMRI interpretation problem from [1]. In this problem we must decide whether the subject was thinking of a tool (such as a hammer) or a building (such as a hospital) by looking at an fMR image of his or her brain. This problem is high-dimensional (thousands of voxels), very noisy, and data-poor (40 training and 40 test examples per subject). Figure 1 shows that the new formulation achieves similar accuracy to the old formulation, on three different subjects. Other methods, such as SVD followed by SVM, were less accurate for the same number of components, as visible in Figure 3.

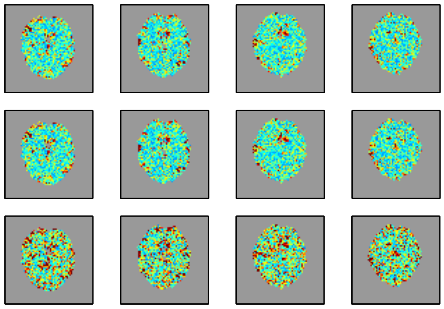

Figure 2: Learned discriminants: new SVDM (top), old SVDM (middle), plain SVM (bottom). Each plot shows the absolute value of the discriminant weight at each voxel (red=larger). The two SVDM discriminants are nearly identical, and much less noisy than the one from the SVM.

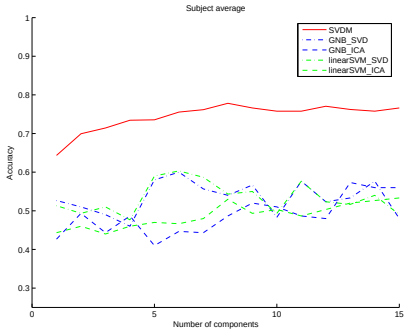

Figure 3: Comparison between SVDM and either Gaussian Naive Bayes (GNB) or Linear SVM on SVD or ICA low-dimensional representations.

In previous work [1] we showed that the original SVDM formulation achieved similar accuracy to a plain SVM on this problem but produced a much cleaner and more interpretable discriminant, placing weights mostly in brain locations consistent with the task being performed. Figure 2 shows that the new formulation achieves similar interpretability.

References

[1] F. Pereira and G. Gordon, "The Support Vector Decomposition Machine," Proceedings of the 23^{rd} International Conference on Machine Learning, 2006.