Alternative Ways to Solve Optimization Problem in Support Vector Decomposition Machine

Mid-Term Project Report

Yongxin (Taylor) Xi
April 27th, 2007

Overview

Note:
Part I through III have been done, while Par IV still in the phase of proposal

Part I Introduction and Problem Presentation

1.1 Background:

To cope with machine learning problems with high dimension of features (thousands and up) and limited independent training examples (dozens to hundreds), 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 support vector machines to learn a classifier. Recently it has been demonstrated that it is possible to **combine the two goals of dimensionality reduction and classification into a single learning objective**. And fMRI analysis results from Francisco06 suggest that the combined-goal single phase optimization (SVDM) achieves better learning performance than the two-phase approaches.

In this project, we will construct SVDM from fMRI data obtained from Psychology Dept., Princeton. The data contains 392 training samples, and each sample is the brain response of a certain person (7 in all, which from now on referred to as 7 subjects) when observing a certain stimulus (7 categories in all, which are man face, woman face, monkey face, dog face, chair, shoe, table). Here we measure brain response by fMRI (function Magnetic Resonance Imaging), and we only collected data from IT (Inferior Temporal) Cortex, which consists of 2048 voxels in our case (according to cognitive neuroscience, the voxels on IT cortex are in charge of this vision task, therefore we consider 2048 voxels from IT cortex as features). In other words, the fMRI data is just a 392 by 2048 matrix X, and its associated 392 by 7 label matrix Y (it takes value in 1 or -1), which tells us what category of stimulus is the subject viewing: For example, if Y(1,1)=1, and Y(1,k)=-1 for k≠1, then it means that sample 1 is observing the first stimulus, which is a man face.

We train the SVDM by these labeled data. After learning the sample data, the classifier should be able to tell which stimulus a subject is looking at merely by the fMR data obtained from the subject's brain.

$$X_{nxm} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} \text{ and } Y_{nxK} = \begin{pmatrix} Y_{11} & \cdots & Y_{1K} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \cdots & Y_{nK} \end{pmatrix}$$

Where n=392 (7 subjects viewing 7 categories, and experiment runs 8 times); m=2048 (number of voxels in 3D IT cortex, the value of each voxel measures the brain response at that voxel); K=7 (number of categories)

After training phase (equivalently: construction phase, or optimization phase), we test the SVDM classifier with new samples (vectors in R^{2048}) and see how well it can predict the category of stimulus although it does not know it.

1.2 Problem Present:

As we mentioned in the background part, we combine two goals (dimension reduction: SVD and classification: SVM) into a single objective.

a) Dimension reduction

The first goal is dimension reduction of the data. Given X, find Z and W so that:

$$X_{nxm} \approx Z_{nxl} W_{lxm} \, , \, |W(i,:)|_2 = 1 \qquad (1)$$

is essentially a SVD (Singular Value Decomposition) problem, where l<<m is the reduced dimension. The reason we would like each row of W to have unit norm is that W serves as a basis matrix: each of its l rows is a direction of variability of the training examples. Then Z is a just a matrix of coordinates. Linear combination of l rows of W represents the approximated data. To achieve (1), first obtain the full decomposition $X_{nxm} = U_{nxn} \Sigma_{nxn} V^T{}_{nxm}$ where $U^T U = I \;\; V^T V = I$ and Σ_{nxn} is a diagonal matrix with singular values decreasing along the diagonal. Then reduce X to dimension l: $X_{nxm} \approx U(:,1:l)\Sigma(1:l,1:l)V^T(1:l,:)$, the product of first l columns of U, the shrunk lxl diagonal matrix Σ, and the first l rows of transposed V. Finally we can set $Z_{nxl} = U(:,1:l)\Sigma(1:l,1:l)$, and $W_{lxm} = V^T(1:l,:)$, so that (1) is satisfied.

b) Classification

Suppose there are K > 1 classification problems we wish to solve. The label matrix Y_{nxK} is given, and takes value in {1, -1}. To solve the classification problem, we can use the low-dimensional representation of data (matrix Z) from a), can look for matrix Q_{lxK} such that $\mathrm{sgn}(ZQ)_{nxK}$ is a good approximation to Y_{nxK}. Here sgn() is the component-wise sign function, so for example if Y_{ij} is 1 we wish the corresponding element of $(ZQ)_{ij}$ to be positive and large.

c) Optimization Problem:

Simultaneously search for Z, W and Q which minimize the following objective: (Note that the first term contains matrix variables Z and W, while the second term contains Z and Q)

$$\underset{Z,W,Q}{Min} \| X - ZW \|_F^2 + \sum_{i=1}^{n} \sum_{j=1}^{K} \max(0, \lambda(\mu - Y_{ij}[ZQ]_{ij})) \qquad (2)$$

$$Subject\ to: \quad |W(i,:)|_2 = 1 \;\; , i = 1,2,...,l \qquad (3)$$

$$|Q_{i,j}| \le 1, \;\; i = 1,2,...,l; \;\; \text{j=1,2,....,k} \qquad (4)$$

Variables: $Z \in R^{nxl}, W \in R^{lxm}, Q \in R^{lxK}$

Constants: $X_{nxm} \in R^{nxm}, Y_{nxK} \in \{1,-1\}^{nxK}$

Parameters: $\lambda > 0, u > 0, l \in Z^{++}$

This objective trades off reconstruction error (the first term) with a measure of classification error (the second term). Parameter λ controls the weight of the penalty of classification error, and parameter u controls the margin of SVM in the classification task.

The objective function (2) is clearly not convex, because all the variables appear in bilinear forms. Neither is constraint (3) convex. So this is a non-convex nonlinear optimization problem. In Part II we use sequential optimization technique to solve Z, W, and Q one at a time, and this alternating minimization procedure will converge to a local minimum. In Part III we utilize linear algebra to largely reduce complexity of the problem, and show the reformulated problem obtains the same results as the original problem (yet still using sequential optimization). In Part IV we analyze the drawbacks of sequential optimization, and try to propose different ways to tackle the non-convex bilinear terms, by transforming or relaxing them into convex terms, and hence solve the whole problem overall.

d) SVDM Testing

After optimization based on training samples, SVDM is constructed. Given a new sample x (in our case, $x \in R^{1x2048}$), the classifier should be able to predict what category the new sample belongs to (in our case, there are 7 categories). The percentage of accuracy is a measure of how good the SVDM is. Usually, cross validation (also is called: leave-one-out), independent testing are two major testing methods, so we will show results based on both.

Part II Construction of SVDM via Sequential Optimization

2.1 Optimization Procedure

We can solve the SVDM optimization problem by solving Q, Z, W sequentially and iteratively. Holding two of the three matrices fixed at each step simplifies the optimization problem: It makes each problem convex, and reduces the problem complexity at the same time. Because each step reduces the overall objective procedure, and because the objective is always positive, this sequential optimization procedure will converge to a local optimum.

2.1.1 Initialization

Follow the SVD procedure introduced in 1.2 a), initialize matrices Z, W based on the first term of I (2).

2.1.2 Given Z and W, solve for Q

Since Z and W are fixed, the first term of the objective can be dropped, so the rest of the problem is then:

$$\underset{Q}{Min} \sum_{i=1}^{n} \sum_{j=1}^{K} \max(0, \lambda(\mu - Y_{ij}[ZQ]_{ij})) \qquad (5)$$

$$Subject\ to: \quad |Q_{i,j}| \le 1,\ i = 1, 2, ..., l; \quad \text{j=1,2,....,k} \qquad (6)$$

This is essentially a LP, because there are just linear terms and max functions in the objective. It will be clearer if we use two inequality constraints to replace max function, and divide the problem into K sub-problems, the jth of which is a problem of the jth column of matrix variable Q. Denote $Q(:, j)$ as the jth column of Q, then its associated problem is:

$$\underset{Q(:,j),hi}{Min} \quad \sum_{i=1}^{n} hi \qquad (7)$$

$$Subject\ to: \ |Q_{i,j}| \le 1,\ i = 1, 2, ..., l \qquad (8)$$

$$hi \ge 0,\ i = 1, 2, ..., n \qquad (9)$$

$$hi \ge \lambda(\mu - Y_{ij}[ZQ]_{ij}) \qquad (10)$$

This problem is solved by MOSEK LP.

2.1.3 Given W and Q, solve for Z

This problem is harder to solve, since it contains both items of the objective. However it is convex and is a QCQP problem.

$$\underset{Z}{Min} \| X - ZW \|_F^2 + \sum_{i=1}^{n} \sum_{j=1}^{K} \max(0, \lambda(\mu - Y_{ij}[ZQ]_{ij})) \qquad (11)$$

We can divide the problem into n sub-problems, so that the ith sub-problem optimizes respect to the ith row of Z.

$$\underset{Z(i,:)}{Min} \| X(i,:) - Z(i,:)W \|_F^2 + \sum_{j=1}^{K} hj \qquad (12)$$

$$Subject\ to: \ hj \ge 0,\ j = 1, 2, ..., K \qquad (13)$$

$$hj \ge \lambda(\mu - Y_{ij}[Z(i,:)Q]_j),\ j = 1, 2, ..., K \qquad (14)$$

This problem is solved by MOSEK QCQP.

2.1.4 Given Z and Q, solve for W

Since only the first term involves W, we drop the second term, which turns the problem into:

$$\underset{W}{Min} \| X - ZW \|_F^2 \qquad (15)$$

$$Subject\ to: \ |W(i,:)|_2 = 1 \ \ , i = 1,2,...,l \qquad (16)$$

This is a non convex problem, yet with relaxation of the constraint to:

$$|W(i,:)|_2 \le 1 \ \ , i = 1,2,...,l \qquad (17)$$

It becomes a convex problem (QCQP) and can be solved.

Interestingly enough, if we set the initial value for W via 2.1.1, the norm of the rows of W will remain very close to one at each iteration. Also, it has been shown from Franciso06 that SVDM with non-unit norm W still performs quite well, so we might as well drop the constraint (16). If so, the problem can be further decomposed into m sub-problems:

$$X(:,j) \approx ZW(:,j), \ j = 1,2,...,m \qquad (18)$$

It means we can find best W(:,j) by solving a linear regression problem for each column of X, and the answer to this problem is well known:

$$W(:,j)=Z^{-1}X(:,j), \ j = 1,2,...,m \qquad (19)$$

Where Z^{-1} is the pseudo-inverse of Matrix Z.

This problem can be well-solved by Matlab.

2.2 SVDM Testing Results

Given a new data $x \in R^{1x2048}$, first find its reduced dimensional representation by $z = x\, W^{-1}$, where W^{-1} is the pseudo inverse of matrix W. Then the prediction vector is $y = zQ$, $y \in R^K$, and the SVDM would vote on the one out of K categories which has the highest prediction score.

Test of SVDM on new data		**7 Categories (K=7)**	**2 Categories (K=2)**
Random guessing		14.28%	50%
Independent among subjects	Generalization on new subject	30%	69%
	Generalization on new sample	48%	93.9%
Independent within subjects		53%	92%
Cross-Validation		50%	81%

A classifier is effective if it does better than random guessing. Since fMRI data is usually quite noisy, the classification results from the table are quite good, and accord with a cognitive neuroscience explanation about human visual mechanism of distinguishing different objects.

Part III Reduction of Problem Complexity by linear algebra theory

3.1 QR Factorization

QR decomposition of a real square matrix *A* is a decomposition of *A* as

$$A = QR$$
$$A \in R^{mxn}, m > n;\ Q \in R^{mxn}, Q^T Q = I;\ \ R \in R^{nxn}$$

Where *Q* is an orthogonal matrix and *R* is an upper triangular matrix.

In our case, data matrix is a n by m matrix. Take $A = X^T$, and conduct QR factorization on matrix A: $A = QR$, therefore $X = LQ^T$, where $L = R^T$ is the n by n lower triangular matrix.

Now, $X = LQ^T = \begin{bmatrix} L & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ \tilde{Q}^T \end{bmatrix}$, where $\tilde{Q}^T$ forms the other (m-n) orthonormal bases so that $B = \begin{bmatrix} Q\ \tilde{Q} \end{bmatrix}$ forms the full orthonormal bases in column for R^{mxm}.

Take following transformations:

$$\begin{aligned}
\| X - ZW \|_F^2 &= \| \begin{bmatrix} L & 0 \end{bmatrix} B^T - ZW \|_F^2 \\
&= \| (\begin{bmatrix} L & 0 \end{bmatrix} - ZWB) B^T \|_F^2 \\
&= \| (\begin{bmatrix} L & 0 \end{bmatrix} - ZWB) \|_F^2 \\
&= \| (\begin{bmatrix} L & 0 \end{bmatrix} - Z[\tilde{W}_1\ \ \tilde{W}_2]) \|_F^2 \\
&= \| L - Z\tilde{W}_1 \|_F^2 + \| 0 - Z\tilde{W}_2 \|_F^2
\end{aligned}$$

In order to minimize the norm, second term can be made zero by setting $\tilde{W}_2$ to be zeros. In this way, the complexity of $\| X - ZW \|_F^2$ reduces from n by m (392 by 2048) to n by n (392 by 392). Therefore, the optimization problems in 2.1.3 and 2.1.4 can be both solved much faster.

3.2 Implementation of QR factorization
The sizes of matrices Z and Q stay the same while W is reduced to n by n. We implemented the newly formulated optimization problem, and as expected, get the same results while achieves approximately 4 times faster in speed.

Part IV Challenges and Proposals

4. Challenges

Although the sequential optimization introduced in Part II can solve this problem, but it only achieves local optimum, because during each step we constrain two out of three matrix variables to be constant. If we want to get global optimum, the three matrix variables need to be optimized at the same time.
Let us revisit the modified problem (see Part III):

$$\underset{H,P,Z,W,Q}{Min} \| A - H \|_F^2 + \sum_{i=1}^{n} \sum_{j=1}^{K} \max(0, \lambda(\mu - Y_{ij} P_{ij}))$$

Subject to:

$$H_{ij} = \sum_{k=1}^{l} Z_{ik} W_{kj}, \ i = 1,2,...,n, \ j = 1,2,...,n \qquad (1)$$

$$P_{ij} = \sum_{k=1}^{l} Z_{ik} Q_{kj}, \ i = 1,2,...,n, \ j = 1,2,...,K \qquad (2)$$

$$\| W(i,:) \| = 1, \ i = 1,2,...,l \qquad (3)$$

Where A is a constant n by n matrix, and Y_{ij}, λ, μ are constant scalars. While the objective function is convex, none of the constraints are convex.

The bilinear equality constraints in (1) and (2) are especially annoying, and they form the majority of constraints. Therefore, if we want to solve the problem globally, bilinear constraint should be tackled first.

Some methods so far I have thought of:
4.1 GP

Take the log of the bilinear variables as new variables, so that the bilinear equality constraints change into linear constraints. To see how this works, first reformulate the problem by adding new variables $s_k^{(i,j)}$ and:

Minimize:

$$\| A-H \|_F^2 + \sum_{i=1}^{n}\sum_{j=1}^{k} \max(0, \lambda(\mu - Y_{ij}P_{ij}))$$

Subject to:

$$H_{ij} = \sum_{k=1}^{l} s_k^{(i,j)},\ i=1,2,...,n,\ j=1,2,...,n \qquad (4)$$

$$s_k^{(i,j)} = Z_{ik}W_{kj},\ k=1,2,...,l \qquad (5)$$

$$P_{ij} = \sum_{k=1}^{l} d_k^{(i,j)},\ i=1,2,...,n,\ j=1,2,...,K \qquad (6)$$

$$d_k^{(i,j)} = Z_{ik}Q_{kj},\ k=1,2,...,l \qquad (7)$$

$$\| W(i,:) \| \le 1,\ i=1,2,...,l \qquad (8)$$

Note that: 1) constraints (4) and (6) are now affine; 2) we relax the last constraint by making it an inequality constraint.

If matrices Z, W and Q have all positive entries, then we can take log at both sides of equation (2) and (4), this brings to:

$$\log s_k^{(i,j)} = \log Z_{ik} + \log W_{kj}$$

$$\log d_k^{(i,j)} = \log Z_{ik} + \log Q_{kj}$$

Set:

$$r_k^{(i,j)} = \log s_k^{(i,j)}$$

$$f_k^{(i,j)} = \log d_k^{(i,j)}$$

Then:

$$s_k^{(i,j)} = e^{r_k^{(i,j)}}$$

$$d_k^{(i,j)} = e^{d_k^{(i,j)}}$$

So (1) and (3) will change to:

$$H_{ij} = \sum_{k=1}^{l} e^{r_k^{(i,j)}},\ i = 1,2,...,n,\ j = 1,2,...,n \qquad (9)$$

$$P_{ij} = \sum_{k=1}^{l} e^{f_k^{(i,j)}},\ i = 1,2,...,n,\ j = 1,2,...,K \qquad (10)$$

This is still not convex, but if we further relax (9) and (10) into inequality constraint, then this becomes a convex problem.

Although this seems quite reasonable, a huge assumption we are making here is that all the matrices are point-wise positive. It is clearly not the case, because matrices A and Y can have negative entries. Even with shift of these matrices into positive orthant, it is solvable but will not give good results because we are asking too much in the constraint.

4.2 Convex enclosure relaxation

Suppose the bilinear constraint is $z = xy$ and we need to find a convex relaxation for it. By using the Reformulation-Linearization Technique (RLT, SA92, She02), we have following inequalities:

$$(x - x^L)(y - y^L) \geq 0$$

$$(x - x^L)(y^U - y) \geq 0$$

$$(x^U - x)(y - y^L) \geq 0$$

$$(x^U - x)(y^U - y) \geq 0$$

Which, on substituting xy with z, imply the following linear enclosure for the bilinear surface:

$$z \geq x^L y + x y^L - x^L y^L \qquad (11)$$

$$z \leq x^L y + x y^U - x^L y^U \qquad (12)$$

$$z \leq x^U y + x y^L - x^U y^L \qquad (13)$$

$$z \geq x^U y + x y^U - x^U y^U \qquad (14)$$

where L and U stand for the lower bound and upper bound of the variable. In this way each bilinear equality constraint in (5) and (7) turns into four affine therefore convex constraints such as (11)~(14).

The advantage of this method is that it does not require variables to be positive. The drawbacks are 1) Four times of constraints are introduced so that it increases the problem complexity; 2) The relaxation will be very loose if the bounds of variables are loose. In our case, W is bounded in a way that $\| W(i,:) \| \le 1,\ i = 1,2,...,l$, while Z and Q are not. To solve this problem, we may introduce additional constraint on the range of variables. When the point is close to optimum point, we can set the upper and lower bounds of variable x and y according to their values in the last iteration. For example, we set:

$$x^L = (1-\alpha)x, y^L = (1-\alpha)y, x^U = (1+\alpha)x, y^U = (1+\alpha)y \qquad (15)$$

If we take $\alpha = 0.5$, it can be shown from (11), (12) that $0.75xy \le z \le 1.25xy$. The smaller α is, the more accurate z will be and hence the tighter the inequality constraints (11) ~ (14) become. Therefore, in our algorithm, the value α should be a non-increase function of iteration number as we gradually reach optimal point.

4.3 Taylor Expansion Approximation

To overcome the very high number of new constraints in 2., we can take first order Taylor's expansion to approximate $z = xy$:

$$z = f(x,y) = xy \qquad (16)$$

$$f(x+\Delta x, y+\Delta y) \approx f(x,y) + \frac{\partial}{\partial x} f(x,y)\cdot \Delta x + \frac{\partial}{\partial y} f(x,y)\cdot \Delta y \qquad (17)$$

Therefore,

$$\Delta z \approx \frac{\partial}{\partial x} f(x,y)\cdot \Delta x + \frac{\partial}{\partial y} f(x,y)\cdot \Delta y = y\cdot \Delta x + x \cdot \Delta y \qquad (18)$$

Consider x, y as the value they take in last iteration, so now the new variables turn to $\Delta x, \Delta y, \Delta z$ (Actually Δz can be omitted here, since it is just the linear combination of $\Delta x, \Delta y$)

Similarly, define a parameter α to constrain $\Delta x, \Delta y$:

$$|\Delta x| \le \alpha |x|, \quad |\Delta y| \le \alpha |y| \qquad (19)$$

For example, it can be easily shown by (18) that when α set at 0.5, the error of z introduced by the term $\Delta x \Delta y$ is 9% of the true z when x and y are both positive or negative, while 20% if x and y have different signs. Again value α can be a non-increase function of iteration number as the optimal point is gradually achieved.

New formulation of the problem using Taylor's approximation:

$$\underset{H,P,\Delta Z,\Delta W,\Delta Q}{Min} \| A-H \|_F^2 + \sum_{i=1}^{n}\sum_{j=1}^{K}\max(0,\lambda(\mu - Y_{ij}P_{ij}))$$

Subject to:

$$H_{ij} = H_{ij}^0 + \sum_{k=1}^{l}(Z_{ik}^0 \Delta W_{kj} + W_{kj}^0 \Delta Z_{ik}),\ i=1,2,...,n,\ j=1,2,...,n \quad (20)$$

$$P_{ij} = P_{ij}^0 + (\sum_{k=1}^{l} Z_{ik}^0 \Delta Q_{kj} + Q_{kj}^0 \Delta Z_{ik}),\ i=1,2,...,n,\ j=1,2,...,K \quad (21)$$

$$|\Delta W_{kj}| \le \alpha | W_{kj}^0 |,\ k=1,2,...,l,\ j=1,2,...,n \quad (22)$$

$$|\Delta Z_{ik}| \le \alpha | Z_{ik}^0 |,\ \mathrm{i}=1,2,...,n,\ k=1,2,...,l \quad (23)$$

$$|\Delta Q_{kj}| \le \alpha | Q_{kj}^0 |,\ k=1,2,...,l,\ j=1,2,...,K \quad (24)$$

$$\| W(i,:) \| \le 1,\ i=1,2,...,l \quad (25)$$

Under the approximation, the problem is clearly convex. At each iteration, Z^0, W^0, Q^0 are just the optimizers from the last iteration, and H0, P0 are computed by $H^0 = Z^0 W^0$, $P^0 = Z^0 Q^0$.