

Alternative Ways to Solve Optimization Problem in Support Vector Decomposition Machine

Final Project Report

Yongxin (Taylor) Xi
May 24th, 2007

Overview

- Part I** Introduction of the Problem
- Part II** Construction of SVDM via Alternating Optimization
- Part III** Reduction of Problem Complexity by linear algebra theory
- Part IV** Challenge of Non-convex Bilinear constraints
- Part V** Modified Problem Formulation
- Part VI** Experiments, results and discussions
- Part VII** Future Work and References

Abstract

This report describes the work that has been done in formulation, validation and discussion of SVDM classification methods applied to fMRI data. Part I introduces key concepts, describes the data we will be using, and presents the problem we want to solve. In Part II we use alternating optimization method to solve the problem, which converges to a local minimum. In Part III we successfully utilize linear algebra to largely reduce complexity of the problem. In Part IV we analyze the advantages and drawbacks of alternating optimization, and propose different ways to try obtaining the global optimal by relaxing the bilinear terms into convex terms. But note that by relaxing constraints the problem has changed so that the optimizers are no longer the optimizers in the original problem. In Part V a modified problem formulation will be made, which is more reasonable for the classification problem, and we achieved better classification results with the modified formulation in certain classifications. In Part VI, we show the experiments that have been done, the results of the experiments and some discussions. Lastly, the future work section discusses a new thought about relaxation, and describes the further work to be done.

Part I Introduction of the Problem

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 $k \neq 1$, 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.

$$X_{n \times m} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} = \begin{pmatrix} x_1 \\ \cdots \\ x_n \end{pmatrix} \quad Y_{n \times K} = \begin{pmatrix} Y_{11} & \cdots & Y_{1K} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \cdots & Y_{nK} \end{pmatrix}$$

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 Introduction:

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

The SVDM formulation put forward by Francisco Pereira is:

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

Subject to: $Z_{i,1} = 1$

$$Z_{i,2:end} \leq 1, \quad i = 1, 2, \dots, n$$

$$\|Q_{:,j}\|^2 \leq 1, \quad i = 1, 2, \dots, l; \quad j = 1, 2, \dots, k$$

The two terms in the objective corresponds to SVD and SVM respectively. In this section we will first introduce singular value decomposition (SVD) and support vector machine (SVM), then discuss the formulation and propose new constraints for the problem.

1.2.1 Dimension reduction

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

$$X_{n \times m} \approx Z_{n \times l} W_{l \times m}, \quad |W(i, :)|_2 = 1 \quad (1)$$

is essentially a SVD (Singular Value Decomposition) problem, where $l \ll m$ 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 $X_{n \times m} = U_{n \times n} \Sigma_{n \times n} V_{n \times m}^T$ where $U^T U = I$ $V^T V = I$ and $\Sigma_{n \times n}$ is a diagonal matrix with singular values decreasing along the diagonal. Then reduce X to dimension l : $X_{n \times m} \approx U(:, 1:l) \Sigma(1:l, 1:l) V^T(1:l, :)$, the product of first l columns of U , the shrunk

$l \times l$ diagonal matrix Σ , and the first l rows of transposed V . Finally we can set $Z_{n \times l} = U(:, 1:l) \Sigma(1:l, 1:l)$, and $W_{l \times m} = V^T(1:l, :)$, so that (1) is satisfied.

1.2.2 SVM (support vector machine) Classification

The second term of the objective corresponds to a SVM classification problem, although they are not essentially the same. Here we will first briefly go over the mechanism of SVM.

1.2.2.1 Two category classification

SVM approach aims at achieving the largest possible margin of separation of the data classes. First consider the two category case. Assume that the decision hyperplane can be expressed as $w^T x + b = 0$, where w is the normal to the plane.

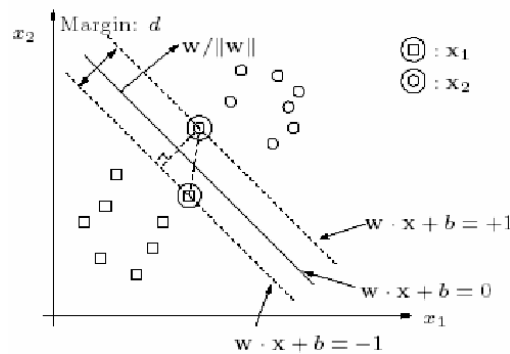


Fig.1

If we assume all the training data to be separable, then there exist w and b such that all data satisfies the following constraints:

$$w^T x_i + b \geq +1, \text{ for } y_i = +1, i = 1, 2, \dots, N$$

$$w^T x_i + b \leq -1, \text{ for } y_i = -1, i = 1, 2, \dots, N$$

If the equality holds for data point x_i , then it is said to be right on the marginal hyperplane. The data points, say x_1 and x_2 , which satisfy

$$w^T x_1 + b = +1$$

$$w^T x_2 + b = -1$$

, will fall on the two hyperplanes that are parallel to the decision plane and orthogonal to w . Subtracting one from the other results in

$$w^T (x_1 - x_2) = 2$$

$$\frac{w^T}{\|w\|} (x_1 - x_2) = \frac{2}{\|w\|}$$

Therefore, the distance between the two hyperplanes is

$$2d = \frac{2}{\|w\|}$$

,where $2d$ can be considered as the width of separation and provides a measure on how separable the two classes of training data are.

Maximizing the distance d is equivalent to minimizing its reciprocal squared, therefore leads to the following formulation:

$$\begin{aligned} \underset{w,b}{\text{Min}} \quad & \frac{1}{2} \|w\|^2 \\ \text{Subject to: } & y_i(x_i^T w + b) \geq 1 \end{aligned}$$

For fuzzily separable data (not clearly separable), the former problem clearly has no solution, because it is impossible to construct a linear hyperplane decision boundary without incurring classification errors. We introduce nonnegative slack variables ξ_i to allow some data to violate the constraint.

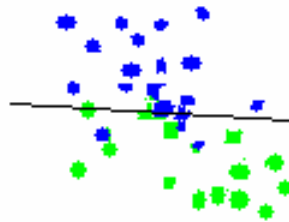


Fig.2

The standard 2-category linear SVM classifier can be constructed through the optimization:

$$\begin{aligned} \underset{w,b,\xi_i}{\text{Min}} \quad & \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\ \text{Subject to: } & y_i(x_i^T w + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \forall i \end{aligned}$$

,again $w \in R^m$ and scalar b define the separating plane, and ξ_i is the slack variable introduced here to tolerate non-separable errors of certain samples.

If $\xi_i = 0$, it means the i th data sample can be classified correctly, if $0 < \xi_i \leq 1$ the i th sample is within the fuzzy region of the classifier, and $\xi_i > 1$ means the i th sample is an outlier. So $\sum_i \xi_i$ gives an upper bound of the number of

classification errors in the dataset, and the parameter C controls the weight of this term in the objective.

1.2.2.2 K category classification

Suppose now there are $K > 2$ classification problems we wish to solve. The classification methods popular in practical use are OVA (one against all), OVO (one against one), which was introduced by Vapnik in 1998 and Krebel in 1999, respectively. In OVA, K binary SVMs will be constructed, each from which is a classifier for one category against all the rest categories. In OVO, we construct $\binom{K}{2}$ binary SVMs for all pairs of different categories. These two methods achieve similar performances (Support Vector Machine for Multi-class Pattern Recognition, J. Weston and C. Watkins), but OVA provides an easier and neater problem formulation.

Now we formulate the K class SVM problem by OVA method:

$$\text{Min } \frac{1}{2} \sum_{j=1}^K \|w_j\|^2 + C \sum_{j=1}^K \sum_{i=1}^N \xi_i^j$$

$$\text{Subject to: } Y_{ij}(w_j^T x_i + b_j) \geq 1 - \xi_i^j, i = 1, 2, \dots, N; j = 1, 2, \dots, K$$

$$\xi_i^j \geq 0, \forall i, j$$

The label matrix $Y_{n \times K}$ is given, and takes value in $\{1, -1\}$.

$$Y_{ij} = +1, \text{ if } x_i \text{ belongs to class } j$$

$$Y_{ij} = -1, \text{ if } x_i \text{ does not belong to class } j$$

w_j and b_j are the norm and intercept to the separating plane of the j th classifier, and ξ_i^j is a measure of classification error brought by the i th sample in the j th classifier.

If $b_j, j = 1, 2, \dots, K$ are set to zero, then an equivalent formulation to this problem is:

$$\text{Min } \frac{1}{2} \|Q\|_F^2 + C \sum_{j=1}^K \sum_{i=1}^N \xi_i^j$$

$$\text{Subject to: } \xi_i^j \geq 1 - Y_{ij}(x_i^T Q(:, j)), i = 1, 2, \dots, N; j = 1, 2, \dots, K$$

$$\xi_i^j \geq 0, \forall i, j$$

In our SVDM, we can use the low-dimensional representation of data $z \in R^l$ (notice that every row of matrix Z now is a lower dimensional representation of the data) instead of the original data $x \in R^m$. So during the

optimization we are looking for a matrix $Q_{l \times K}$ such that $\text{sgn}(ZQ)_{n \times K}$ is a good approximation to $Y_{n \times K}$. Here $\text{sgn}()$ 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.

1.2.3 Optimization Problem:

Our ultimate goal is achieving a) dimension reduction b) K-category classification together. So the two objectives need to be combined in some way.

The problem formulation that Francisco Pereira put forward is to minimize the following objective:

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

$$\text{Subject to: } Z_{i,1} = 1$$

$$Z_{i,2:\text{end}} \leq 1, \quad i = 1, 2, \dots, n$$

$$\|Q_{:,j}\|^2 \leq 1, \quad i = 1, 2, \dots, l; \quad j = 1, 2, \dots, k$$

Variables: $Z \in R^{n \times l}, W \in R^{l \times m}, Q \in R^{l \times K}$

Constants: $X_{n \times m} \in R^{n \times m}, Y_{n \times K} \in \{1, -1\}^{n \times K}$

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

The first term is essentially what has been discussed in part a), but the second term is different. Note that $\|Q\|_F^2$ is omitted and the threshold changed from constant 1 to parameter u . Instead of minimizing $\|Q\|_F^2$ to achieve the largest margin of fuzzy separation, the constraint $\|Q_{:,j}\|^2 \leq 1$ regulates the margin in each SVM to be greater than 2.

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.

As regard to the constraints, the first column of matrix Z are set to be all ones, and the norm of every row of Z except the first entry is less or equal to 1. In this way the first row of W will gradually correspond to the mean of the dataset during the optimization procedure. However, requiring $Z_{i,2:end} \leq 1$ will make entries in other columns much smaller than the first column and might then lose their discriminating power. Also, there are no regulations on the matrix W , whose rows serve as bases for the data after dimension reduction. For example, every column of Z except the first column can be made arbitrarily small if compensate W by multiplying each row of W by the same amount. On the other hand, we might as well subtract the mean from each sample of the dataset first before the optimization procedure, so that the new dataset has zero mean. Also, we would like all the bases of the data (rows in W) to be normalized, i.e. they have norm 1.

We therefore rewrite the constraints to the problem:

$$\text{Subject to: } \|W(i,:)\|_2 = 1, \quad i = 1, 2, \dots, l \quad (3)$$

$$|Q_{i,j}| \leq 1, \quad i = 1, 2, \dots, l; \quad j = 1, 2, \dots, k \quad (4)$$

Now let us look further into the objective. Note that the first term contains matrix variables Z and W , while the second term contains variables Z and Q . The objective 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.

1.2.4 SVDM Testing

After optimization based on training samples, SVDM is constructed. Given a new sample x (in our case, $x \in \mathbf{R}^{1 \times 2048}$), the classifier can predict what category the new sample belongs to (in our case, there are 7 categories) by

$$f(x) = \arg \max_j ((w_j^T x) + b_j), \quad j = 1, 2, \dots, K.$$

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 Alternating Optimization

2.1 Optimization Procedure

We can solve the SVDM optimization problem by solving Q, Z, W alternately 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 alternating 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:

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

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

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:

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

$$\text{Subject to: } |Q_{i,j}| \leq 1, \quad i = 1, 2, \dots, l \quad (8)$$

$$hi \geq 0, \quad i = 1, 2, \dots, n \quad (9)$$

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

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.

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

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

$$\text{Min}_{Z(i,:)} \| X(i,:) - Z(i,:)W \|_F^2 + \sum_{j=1}^K h_j \quad (12)$$

$$\text{Subject to: } h_j \geq 0, j = 1, 2, \dots, K \quad (13)$$

$$h_j \geq \lambda(\mu - Y_{ij}[Z(i,:)Q]_j), j = 1, 2, \dots, K \quad (14)$$

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:

$$\text{Min}_W \| X - ZW \|_F^2 \quad (15)$$

$$\text{Subject to: } \|W(i,:)\|_2 = 1, i = 1, 2, \dots, l \quad (16)$$

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

$$\|W(i,:)\|_2 \leq 1, i = 1, 2, \dots, l \quad (17)$$

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:

$$X(:,j) \approx ZW(:,j), j = 1, 2, \dots, m \quad (18)$$

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:

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

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

This problem is solved by Matlab.

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

$$A = QR$$

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

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, $X = LQ^T = [L \ 0] \begin{bmatrix} Q^T \\ \tilde{Q}^T \end{bmatrix}$, where \tilde{Q}^T forms the other $(m-n)$ orthonormal

bases so that $B = [Q \ \tilde{Q}]$ forms the full orthonormal bases in column for $R^{m \times m}$.

Take following transformations:

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

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.

Part IV Challenge of Non-convex Bilinear constraints

4.1 Advantages and Drawbacks of Alternating Optimization

The alternating optimization introduced in Part II is a very efficient algorithm and will converge to local optimum quickly (generally within 10 iterations). However, because during each step we constrain two out of three matrix variables to be constant, it only achieves local optimum. 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):

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

Subject to:

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

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

$$\|W(i, :)\| = 1, \quad i = 1, 2, \dots, l \quad (3)$$

where A is a constant n by n matrix, Y is a constant n by k matrix, and 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.

4.2 Methods of relaxation:

4.2.1 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:

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

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

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

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

$$\|W(i, :)\| \leq 1, \quad i = 1, 2, \dots, l \quad (8)$$

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:

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

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

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

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

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

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.

4.2.2 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:

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

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

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

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

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 $\|W(i,:)\| \leq 1, i=1,2,\dots,l$, 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:

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

If we take $\alpha = 0.5$, it can be shown from (11), (12) that $0.75xy \leq z \leq 1.25xy$. 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.

4.2.3 Taylor's Expansion Approximation

Convex enclosure relaxation method introduced in 4.2.2 increases the number of constraints to four folds. Another way of relaxation which does not add the problem complexity is to take first order Taylor's expansion to approximate $z = xy$:

Write z as a function of x and y , and expand this function at (x, y) :

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

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

Therefore,

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

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 $\Delta x, \Delta y$:

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

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 both 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 would be:

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

Subject to:

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

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

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

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

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

$$\|W(i, :)\| \leq 1, \quad i = 1, 2, \dots, l \quad (25)$$

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 H^0, P^0 are computed by $H^0 = Z^0 W^0, P^0 = Z^0 Q^0$.

Part V Modified Problem Formulation

5.1 Introducing $\|Q\|_F^2$ term in the objective

The original problem is:

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

$$\text{Subject to: } \|W(i,:)\|_2 = 1, \quad i = 1, 2, \dots, l$$

$$|Q_{i,j}| \leq 1, \quad i = 1, 2, \dots, l; \quad j = 1, 2, \dots, k$$

As discussed in section 1.2.3, the second term of the objective is no longer a standard SVM problem, because 1) the threshold is changed from constant 1 to parameter μ ; 2) $\|Q\|_F^2$ is omitted so instead of minimizing $\|Q\|_F^2$ to achieve the largest margin of fuzzy separation, the constraint $|Q_{i,j}| \leq 1$ regulates the margin.

However, it is possible to make the problem working as a standard SVM, by merely introducing the $\|Q\|_F^2$ term in the objective, deleting the constraint of Q , and setting μ at 1:

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

$$\text{Subject to: } \|W(i,:)\|_2 = 1, \quad i = 1, 2, \dots, l$$

5.2 Introducing b term in the constraints

Note that from the beginning till now the intercept b of the classifier has always been omitted. In preprocessing of the data, we subtract the mean off the data to make it centralized around the origin. However, the separating planes of the K classifiers do not necessarily go cross the origin, although the mass point is there. In fact, for OVA classifiers, the separating planes of centralized dataset will generally not go through the origin, illustrated in the figure below.

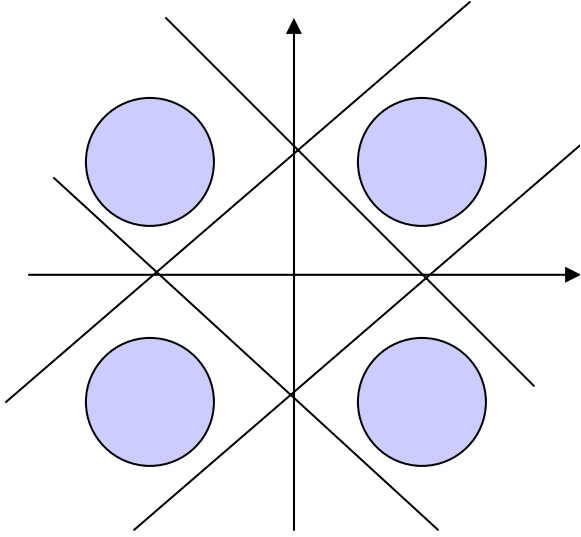


Fig. 3

Fig. 3 illustrates the classification for four category ideal two dimensional data. The data has been centralized, and none of the four separating planes passes the origin. Actually, if any of them is required to go through origin, the classification result will become worse.

The added term b then takes the problem into:

$$\text{Min}_{Z,W,Q} \|X - ZW\|_F^2 + \lambda \left(\frac{1}{2} \|Q\|_F^2 + C \sum_{i=1}^n \sum_{j=1}^K \max(0, 1 - Y_{ij}([ZQ]_{ij} + b_j)) \right)$$

$$\text{Subject to: } \|W(i,:)\|_2 = 1, i = 1, 2, \dots, l$$

Part VI Experiments, results and discussions

Testing of the original problem:

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

$$\text{Subject to: } \|W(i,:)\|_2 = 1, i = 1, 2, \dots, l$$

$$\|Q_{i,j}\| \leq 1, i = 1, 2, \dots, l; j = 1, 2, \dots, k$$

6.1 Tests on Alternating Optimization Method

Each minimization problem is solved either by Matlab implementation or the Mosek optimization package (<http://www.mosek.com>). We stopped when an iteration (a set of three minimizations, with respect to W , Z and Q) yielded less than 0.01% decrease in the objective.

A typical convergence of the optimization problem is shown in Fig. 4. The green line illustrates the convergence of the whole objective. The red line

corresponds with the second term of the objective, and the blue line means the first term. Usually, the optimization takes between 3 to 10 iterations to converge. Note that we start with a good initiation matrices W_0 and Z_0 from SVD, i.e. the procedure described in section 1.2.1., and initiation matrix Q given W and Z following the procedure in section 2.1.2.

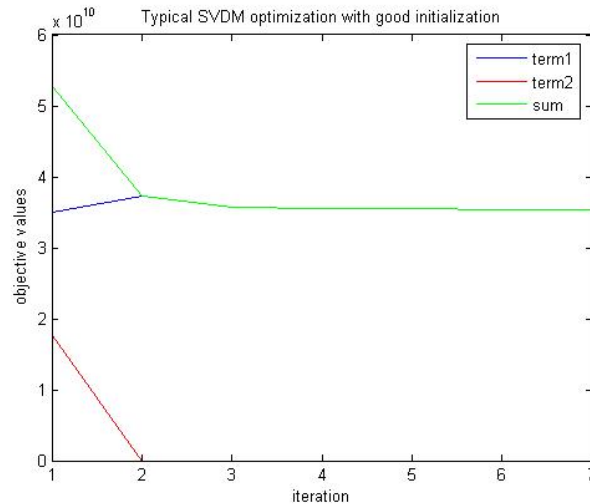


Fig. 4

Since the first term achieves minimum when Z_0 and W_0 are the SVD components of matrix X , the overall objective has the lower bound $\|X - Z_0 W_0\|$, where the blue line intersects with the Y axis. Another remarkable observance is that usually after 2 iterations, the classification error will remain at zero, which says that after a linear transformation of the data in the reduced dimension, the data can be linearly classified nicely.

6.2 Experiments of Classification

There are two main types of validation methods to obtain the classification accuracy of the classifier. The first is called cross validation, or leave-one-out. To test the i th sample, first leave this sample out of the training set, train the classifier based on the rest $n-1$ samples, and then test on the i th sample. The second method is independent testing. To use this testing method in our dataset, we can draw certain amount of data from the data set as training set, while having the rest as testing data.

6.2.1 Cross Validation (leave one out)

- a. 7 category test
- b. 2 category test (face versus object)

6.2.2 Independent Testing

6.2.2.1 Independent among subjects

I. Generalizing on new subjects

a. 7 category test

b. 2 category test (face versus object)

II. Generalizing on new runs

a. 7 category test

b. 2 category test (face versus object)

6.2.2.2 Independent within subject

a. 7 category test

b. 2 category test (face versus object)

6.2.3 Pair-wise Classifier Testing

Construct $\binom{K}{2}$ classifiers for each pair of different stimuli, and test on $z = xy$ their performances.

6.2.3 SVDM Testing Results

Given a new data $x \in R^{1 \times 2048}$, 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:

$$f(x) = \arg \max_j ((w_j^T x) + b_j), j = 1, 2, \dots, K$$

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 subjects	30%	69%
	Generalization on new runs	48%	93.9%
Independent within subjects		53%	92%
Cross-Validation		50%	80%

Table 1

Explanations of table 1:

Independent among subjects task: We input samples of different subjects together into the SVDM training phase, and test on various subjects as well. In the “generalization on new subjects task”, we train the classifier by the first 70% of subjects, (i.e. 5 subjects out of 7) and test on the remaining two subjects. The “generalization on new runs” task draws the front 70% of samples of each subject so that the machine can learn from each subject, and then test on the remaining samples of each subject. The “independent within subjects” task is a task for one person only. The machine learns the front 75% of samples from one person, and is tested on the remaining 25%. We conducted around 20 7-category leave-one-out experiments and around 50 2-category leave-one-out experiments. The SVDM predicts correctly on 10 out of 20, and 40 out of 50, respectively. This task is most time consuming, in future work more experiments need to be done for this task. To provide a comparing standard, independent testing tasks used to measure the performance of other modified SVDM use the same rules of selecting training samples and testing samples as here.

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. From the table, it is obvious that the construction of 7-category classifier is much more demanding than the 2-category classifier. Since there are four face stimuli among the seven categories, they are difficult to tell from each other, because all of them will activate the FFA area (Fusiform Face Area, Kanwisher 2000) in the IT cortex, so that they share a much more similar pattern than the objects.

To see more clearly how difficult it is to tell one stimulus from another, we conduct the pairwise classification of different stimuli. The classification accuracy of each pairwise classification is shown in Table 2. For each pair, 75% samples from each stimulus are used in training phase, and the remaining 25% samples will be tested.

Independent among subjects

accuracy	F face	M face	Monkey	Dog	House	Chair	Shoe
F face	0	0.46429	0.60714	0.64286	0.85714	0.89286	0.67857
M face	0.46429	0	0.64286	0.60714	0.89286	0.96429	0.82143
Monkey	0.60714	0.64286	0	0.67857	0.75	0.75	0.78571
Dog	0.64286	0.60714	0.67857	0	0.96429	0.96429	0.92857
House	0.85714	0.89286	0.75	0.96429	0	0.75	0.75
Chair	0.89286	0.96429	0.75	0.96429	0.75	0	0.64286
Shoe	0.67857	0.82143	0.78571	0.92857	0.75	0.64286	0

Table 2

The table accords with our expectation: classifiers for face versus non-face stimuli achieve a much better score than face versus face cases. Object versus object classifiers are better than face versus face but poorer than face versus objects. It indicates that face information is processed in a very unique way in the neural cortex.

6.3 Implementation of QR factorization

Complexity of solving W reduces from n by m (392 by 2048) to n by n (392 by 392). We implemented the newly formulated optimization problem, and as expected, get the same results while achieves approximately 4 times faster in speed.

6.4 Experiments of relaxation into convex problem

Relax the bilinear terms into four linear inequality constraints, or into a linear equality constraint by using convex enclosure relaxation or Taylor Expansion Method. We selected the good initialization point to start with, and adjusted the parameter α to different values. To our disappointment, however, the method is not promising. The optimizers of the problem do not decrease the objective in the primal problem. It might be due to the inaccuracies accumulated by the sum of bilinear terms in the objective and constraints. To address this particular problem, another proposal will be discussed in the Future Work section.

6.4 Tests on the Reformulated Problem

Modify the objective and constraints of the problem so that it becomes the combination of SVD and standard SVM problem. The pair-wise classifier test results in Table 3 shows an increase of classification accuracy for most of the classifiers when compared with Table 2. Among the 21 classifiers, 12 of them

have better performance, 6 out of them remain the same, and 3 decreased (they are Woman vs. dog; man vs. dog; House vs. woman)

accuracy	F face	M face	Monkey	Dog	House	Chair	Shoe
F face	0	0.57143	0.67857	0.57143	0.82143	0.92857	0.67857
M face	0.57143	0	0.75	0.53571	0.96429	0.96429	0.85714
Monkey	0.67857	0.75	0	0.78571	0.85714	0.78571	0.82143
Dog	0.57143	0.53571	0.78571	0	0.96429	0.96429	0.96429
House	0.82143	0.96429	0.85714	0.96429	0	0.75	0.85714
Chair	0.92857	0.96429	0.78571	0.96429	0.75	0	0.64286
Shoe	0.67857	0.85714	0.82143	0.96429	0.85714	0.64286	0

Table 3

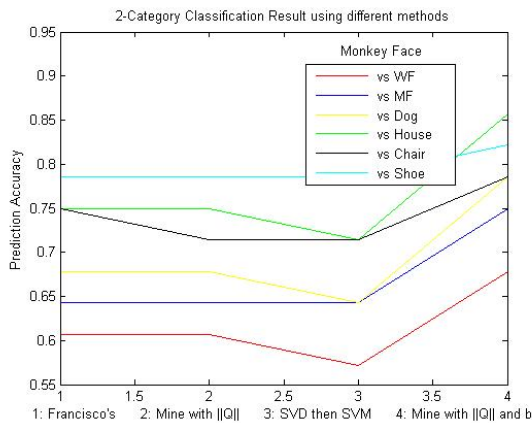


Fig. 5

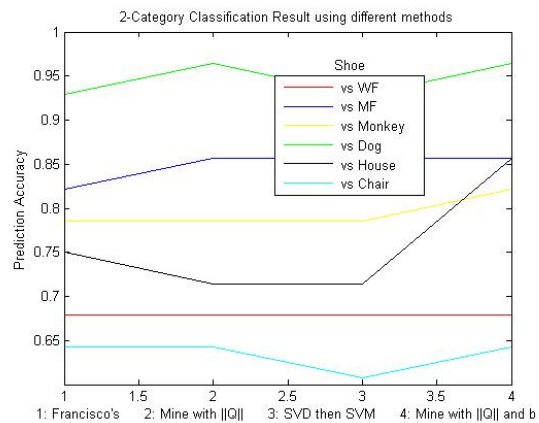


Fig. 6

Fig. 5 and Fig. 6 illustrate the change of some certain classifier performance with respect to different formulation of the problem. For example, for the monkey versus other stimulus task, the new formulation for the SVDM construction achieves best performance.

Part VII Future Work and References

7.1 Future work 1

- More tests to be done to test the improvements under the new formulated SVDM problem, for example, independent testing and leave-one-out testing.
- Use random generalization of training sets and testing sets and compare the averaged results for different formulations of SVDM classifiers.

7.2 Future work 2

-Relaxation of the problem:

Revisit Convex Enclosure Relaxation: the essence of the idea is to box the non-convex curved surface with four pieces of linear planes. Therefore, the points can wander wherever within the box—no longer on the curved surface $z = xy$ now. That is why so much error is introduced. If after we get the optimizers we project them again onto the surface $z = xy$, then the error will be dramatically reduced while it still serves as optimizer candidates for the primal problem.

References

1. F. Pereira and G. Gordon, “The Support Vector Decomposition Machine,” Proceedings of the 23rd International Conference on Machine Learning, 2006
2. Boyd, S. & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press.
3. S.Y. Kung and M.W. Mak, (2004). Biometric Authentication, A machine learning approach, Pearson Education
4. David W. Walker, Eigenvalues and Singular Values (Chapter 10)
5. Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition