Project 3: Movement Decoding for Brain Computer Interfaces

<Jin Gu>
<jingu@andrew.cmu.edu>

Abstract

In this project, a classifier is implemented to detect the movement communication between brain and computers. There are two kinds of movement to be classified, which are left movement and right movement. Therefore, it is a two-class supervised learning problem and SVM is a good method to solve it.

1. Overview

The goal of this project is train an SVM classifier for brain signals. Therefore, Interior point method, Newton method and linear search is implemented to solve the optimization problem for an SVM solver. What is more, two-level cross validation is used to avoid over-fitting. The first level is six-fold and the second level is five-fold. Test accuracy, mean accuracy and standard deviation of all folds should be estimated. At last, the result can be shown with an applied function.

2. Mathematical Formulation

2.1 Logarithmic barrier and Interior point method

An SVM classifier is

$$f(X) = W^T X + C \begin{cases} \geq 0 & Class\ A \\ < 0 & Class\ B \end{cases}$$

And an SVM solver tries to maximize the margin between two classes, which can be changed to the follow problem

$$\begin{aligned} & \min_{W,C,\xi} \sum_{i=1}^{N} \xi_i + \lambda \cdot W^T W \\ S.T.\quad & y_i \cdot (W^T X_i + C) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & (i = 1,2,\cdots,N) \end{aligned}$$

And with Logarithmic barrier, this constrained optimization can be converted to an unconstrained problem.

$$\min_{W,C,\xi} \quad \sum \xi_i + \lambda \cdot W^T W - \frac{1}{t}\sum_{i=1}^{N} \log\left(W^T X_i \cdot y_i + C \cdot y_i + \xi_i - 1\right) - \frac{1}{t}\sum_{i=1}^{N} \log(\xi_i)$$
$$t \to \infty$$

Make

$$Z = [W, C, \xi]$$

With Interior point method, this unconstrained problem can be solved.

The method starts with an initial guess with $Z^{(0)} = [W^{(0)}, C^{(0)}, \xi^{(0)}]$. Here, $W^{(0)}$ and $C^{(0)}$ can be any value and I set $W^{(0)}$ to be all one and $C^{(0)}$ zero. $\xi^{(0)}$ should be calculated with the formulation:

$$\xi_i^{(0)} = \max\{1 - y_i \cdot [W^{(0)T} X_i + C^{(0)}], 0\} + 0.001$$

Then the project iteratively solves the unconstrained nonlinear optimization problem to find the optimal solution for $[W^*, C^*, \xi^*]$ until $t \geq T_{max}$. Here $T_{max} = 1000000$ and $t = \beta t\ (\beta = 15)$.

2.2 Newton method

In Interior point method algorithm, Newton method is used to solve the unconstrained nonlinear optimization problem.
Newton method starts with an initial

$$Z^{(0)} = [W^{(0)}, C^{(0)}, \xi^{(0)}]$$

Then it iteratively computes the Newton step and decrement

$$\Delta Z = -\nabla^2 f(Z)^{-1} \cdot \nabla f(Z)$$
$$\sigma = -\nabla f(Z)^T \cdot \Delta Z$$

To check if

$$\sigma/2 \leq tolerance (tolerance = 0.000001)$$

If not, it use backtracking linear search to choose step size s to update

$$Z = Z + s \cdot \Delta Z$$

2.3 Backtracking linear search

Backtracking linear search is used to choose step size s to update Z.
It starts at s = 1, then if check if

$$\begin{cases} W^{(k)T} X_i \cdot y_i + C^{(k)} \cdot y_i + \xi_i^{(k)} - 1 > 0 \\ \xi_i^{(k)} > 0 \end{cases} (i = 1,2,\cdots,N)$$

If no, it updates s and Z with

$$s = 0.5s.$$
$$[W^{(k)}, C^{(k)}, \xi^{(k)}] = Z + s \cdot \Delta Z$$

Otherwise it quits.

2.4 Calculate gradient and hessian matrix in Newton method

In Newton method, gradient and hessian matrix of cost function are needed to check a stop condition.
From the cost function,

$$\min_{W,C,\xi} \sum \xi_i + \lambda \cdot W^T W - \frac{1}{t}\sum_{i=1}^{N} \log\left(W^T X_i \cdot y_i + C \cdot y_i + \xi_i - 1\right) - \frac{1}{t}\sum_{i=1}^{N} \log(\xi_i)$$
$$t \to \infty$$

For convenience, make

$$T_i = W^T X_i \cdot y_i + C \cdot y_i + \xi_i - 1$$

Gradient matrix should be

$$G = \left[\frac{\partial f}{\partial W}, \frac{\partial f}{\partial C}, \frac{\partial f}{\partial \xi}\right]$$

Which is

$$\frac{\partial f}{\partial w_j} = 2 \cdot \lambda \cdot w_j - \frac{1}{t} \cdot \sum_{i=1}^{N} \frac{x_{ij} y_i}{T_i}$$
$$\frac{\partial f}{\partial C} = -\frac{1}{t} \cdot \sum_{i=1}^{N} \frac{y_i}{T_i}$$
$$\frac{\partial f}{\partial \xi_i} = 1 - \frac{1}{t} \cdot \frac{1}{T_i} - \frac{1}{t} \cdot \frac{1}{\xi_i}$$

And hessian matrix should be

$$\begin{bmatrix}
\frac{\partial^2 f}{\partial w_1^2} & \frac{\partial^2 f}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_1 \partial w_M} & \frac{\partial^2 f}{\partial w_1 \partial C} & \frac{\partial^2 f}{\partial w_1 \partial \xi_1} & \cdots & \frac{\partial^2 f}{\partial w_1 \partial \xi_N} \\
\frac{\partial^2 f}{\partial w_2 \partial w_1} & \frac{\partial^2 f}{\partial w_2^2} & \cdots & \frac{\partial^2 f}{\partial w_2 \partial w_M} & \frac{\partial^2 f}{\partial w_2 \partial C} & \frac{\partial^2 f}{\partial w_2 \partial \xi_1} & \cdots & \frac{\partial^2 f}{\partial w_2 \partial \xi_N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\
\frac{\partial^2 f}{\partial w_M \partial w_1} & \frac{\partial^2 f}{\partial w_M \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_M^2} & \frac{\partial^2 f}{\partial w_M \partial C} & \frac{\partial^2 f}{\partial w_M \partial \xi_1} & \cdots & \frac{\partial^2 f}{\partial w_M \partial \xi_N} \\
\frac{\partial^2 f}{\partial C \partial w_1} & \frac{\partial^2 f}{\partial C \partial w_2} & \cdots & \frac{\partial^2 f}{\partial C \partial w_M} & \frac{\partial^2 f}{\partial C^2} & \frac{\partial^2 f}{\partial C \partial \xi_1} & \cdots & \frac{\partial^2 f}{\partial C \partial \xi_N} \\
\frac{\partial^2 f}{\partial \xi_1 \partial w_1} & \frac{\partial^2 f}{\partial \xi_1 \partial w_2} & \cdots & \frac{\partial^2 f}{\partial \xi_1 \partial w_M} & \frac{\partial^2 f}{\partial w_2 \partial C} & \frac{\partial^2 f}{\partial \xi_1^2} & \cdots & \frac{\partial^2 f}{\partial \xi_1 \partial \xi_N} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial \xi_N \partial w_1} & \frac{\partial^2 f}{\partial \xi_N \partial w_2} & \cdots & \frac{\partial^2 f}{\partial \xi_N \partial w_M} & \frac{\partial^2 f}{\partial \xi_N \partial C} & \frac{\partial^2 f}{\partial \xi_N \partial \xi_1} & \cdots & \frac{\partial^2 f}{\partial \xi_N^2}
\end{bmatrix}$$

Which is

$$\frac{\partial^2 f}{\partial w_j^2} = 2 \cdot \lambda + \frac{1}{t} \cdot \sum_{i=1}^{N} \frac{(x_{ij} y_i)^2}{T_i^2}$$
$$\frac{\partial^2 f}{\partial w_j \partial w_k} = \frac{1}{t} \cdot \sum_{i=1}^{N} \frac{x_{ij} x_{ik} y_i^2}{T_i^2}$$
$$\frac{\partial^2 f}{\partial w_j \partial C} = \frac{1}{t} \cdot \sum_{i=1}^{N} \frac{x_{ij} y_i^2}{T_i^2}$$
$$\frac{\partial^2 f}{\partial w_j \partial \xi_i} = \frac{1}{t} \cdot \frac{x_{ij} y_i}{T_i^2}$$
$$\frac{\partial^2 f}{\partial C^2} = \frac{1}{t} \cdot \sum_{i=1}^{N} \frac{y_i^2}{T_i^2}$$
$$\frac{\partial^2 f}{\partial C \partial \xi_i} = \frac{1}{t} \cdot \frac{y_i}{T_i^2}$$
$$\frac{\partial^2 f}{\partial \xi_i^2} = \frac{1}{t} \cdot \frac{1}{T_i^2} + \frac{1}{t} \cdot \frac{1}{\xi_i^2}$$
$$\frac{\partial^2 f}{\partial \xi_i \partial \xi_j} = 0$$

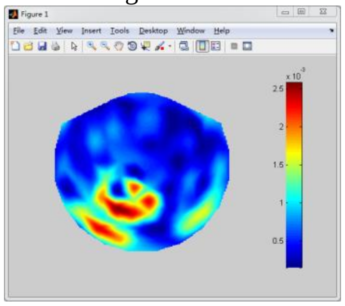
3. Experimental Results

For the first training fold, show the values of *W* and *c* in a table.

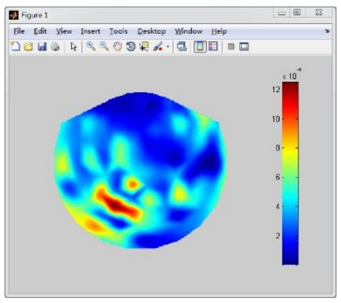
W	feaSubEImg	feaSubEOvert
1	0.0026	0.00096
2	0.0023	0.00093
3	0.0023	0.00069
4	0.0023	0.00063
5	0.0021	0.00062
C	-2.6832	-0.6344

For the first training fold, show the channel weights plot as introduced in the lecture slide.

feaSubEImg



feaSubEOvert



Show the test accuracy mean accuracy and standard deviation of all folds.

	feaSubEImg	feaSubEOvert
Fold_1	0.9250	0.8750
Fold_2	0.7750	1
Fold_3	0.8750	0.9750
Fold_4	1.0000	1
Fold_5	0.8750	0.8750
Fold_6	0.9000	0.9250
Mean	0.8917	0.9417
Std	0.0736	0.0585

Comparison between two kinds of accuracy

In this project, two-level cross validation is applied to avoid over-fitting. In the second level, train data from first level are divided into two dataset. One is training dataset, another is test dataset. With this two dataset, optimal solution and lambda can be found. The first accuracy can be obtained. After this, I use the best lambda and all the training data of first level to train the classifier again and get the second accuracy. Here is the result.

feaSubEImg	First accuracy	Second accuracy
Fold_1	0.9250	0.8500
Fold_2	0.7750	0.8000
Fold_3	0.8250	0.8250
Fold_4	1.0000	0.9750
Fold_5	0.8750	0.9000
Fold_6	0.9000	0.9250
Mean	0.8917	0.8792
Std	0.0736	0.0660

The results show that with more data, the standard deviation decreases, which means that over-fitting problem can be fixed by adding more data.

4. Discussion

4.1 Factors that may impact classification accuracy

In this project, features of data, value of lambda and initial parameters can impact the accuracy greatly. The goal of SVM is to find a linear decision boundary between two classes. With better features, the classifier can separate the test data easier. In SVM, lambda is use to solve the unconstrained optimization problem. With a good lambda, the problem can converge more accurately. What is more, the value of initial parameters, like beta, tolerance and Tmax is very important, they decide the iterator times and convergence threshold.

4.2 Anything unique you have done to improve/validate your program's accuracy/efficiency

In order to improve the efficiency, when calculating the gradient and hessian matrix, it is better to use matrix form operation because it is much faster than using for loop. Though in this project, the dataset in small, therefore there is no big difference, all calculation have been converted into matrix form operation as much as possible.

4.3 Possible improvements can be done.

There are still many ways to improvement the project performance. For example, PCA (principal component analysis) can be implemented to reduce the dimensions of data features. With PCA, the classifier can be faster and avoid considering some useless features. What is more, more lambda should be tested to find the best one. Finally, data is always the important one, more data should be collected to get a more accurate classify. And other methods to solve the unconstrained optimization problem can be used in project, like QP solver. All in all, it will be a very interesting thing to do the project optimization.

References.