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.

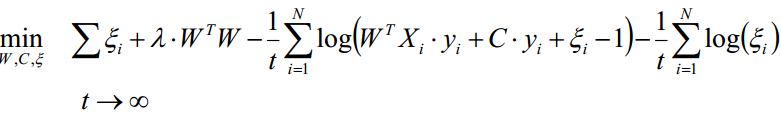
# 2. Mathematical Formulation

**2.1 Logarithmic barrier and Interior point method**

An SVM classifier is

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

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



Make

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

The method starts with an initial guess with. Here, and can be any value and I set to be all one and zero. should be calculated with the formulation:

Then the project iteratively solves the unconstrained nonlinear optimization problem to find the optimal solution for until. Here and.

**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

Then it iteratively computes the Newton step and decrement

To check if

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

**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

If no, it updates s and Z with

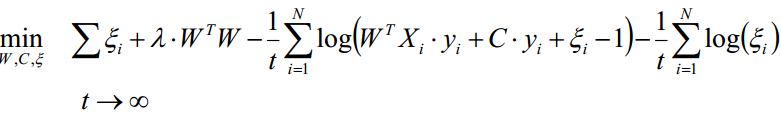
.

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,



For convenience, make

Gradient matrix should be

Which is

And hessian matrix should be

Which is

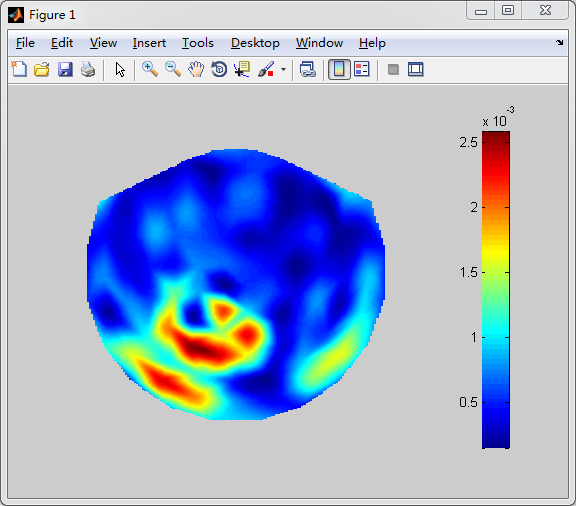
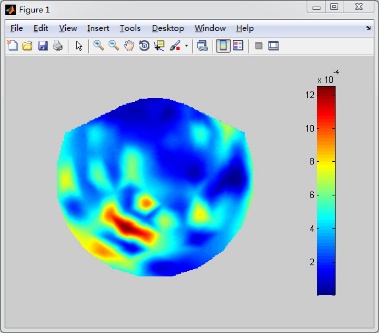
# 3. Experimental Results

**For the first training fold, show the values of and 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.