# Big data and high dimensional data analysis, computer exercise 2

Xijia Liu

November 20, 2017

## Introduction

In this computer exercise, we practice on the knowledge what we have learned from lecture 4 to 9. The studying points will be covered are linear classifiers (logistic regression, linear discriminant analysis and Perceptron), nonlinear classifier (K Nearest Neighbor), feature extraction, variables selection (shrinkage method), model evaluation and validation. There are 2 main exercises based on two different data sets which are Breast Cancer and Gene Expression. By solving task 1 and 2, we can learn how to implement the training and validation methods on real data. There is a pure coding exercise as well. We will try to implement perceptron algorithm by our own coding. This is also an pedagogical exercise which will help us to introduce the important concepts next step.

**The last day to hand in the report: 2017-12-07.**

Each task is graded: Very Good (VG), Good (G) and Not good (NG). A grading can be given conditionally, e.g. you will get the grade VG if you address the comments and remarks made by the teacher.

The grading on computer exercise 1 is as follows

Grade 5: All tasks are solved, handed in in time and graded VG.

Grade 4: Task 1 and 2 are solved, handed in in time and all are graded VG.

Grade 3: Task 1 and 2 are solved, handed in in time and all are graded G at least.

# Task 1

In this task, we work on 'BreastCancerDataTrain.txt' data which you can find on Cambro. This data set contains breast cancer diagnostic results and 10 features created from medical image of breast mass from 469 patients. For more information about the data, please check the help.txt document.

## Task 1.1 Visualize data

First, we should visualize our data and to feel the complexity of this problem. You can apply the methods from computer exercise 1. My suggestion is that plot the pairwise scatter plot of the first 5 principle components of the feature variables.

## Task 1.2 Linear Discriminant Analysis (LDA)

In this sub-task, you need to build a LDA classifier based on the whole data. Then apply your LDA classifier on the whole data, and calculate the accuracy. Hint: You can find the build in function 'lda' from package 'MASS'. The build function 'predict' can help you to do the prediction. Please the help documents in R when you get question about the build in functions.

## Task 1.3 Logistic Regression (LR)

Now, you need to build a LR classifier on whole data. Then apply your LR classifier on the whole data, and calculate the accuracy. Hint: The build in function of logistic regression in R is glm. You need to choose the right family of the distribution of the target variable. A little bit different from 'lda' function, you can not simply get the prediction results by build in function

predict, since it will return the linear predictor. You need further calculate the probability using logit function

$$\phi(s) = \frac{1}{1 + e^{-s}}$$

### Task 1.4 K-Nearest Neighbor (KNN)

Please fit a KNN model on the data. You can choose the number of neighbor as 5. Then apply your KNN classifier on the whole data, and calculate the accuracy. The build in function of KNN is 'knn' in R.

### Task 1.5 Validation and model selection

Here, you need to perform a validation to select the best model from previous sub-tasks. In other words, you need to tuning the hyper parameter in KNN and select the best model from LDA, LR and KNN. You can freely chose your validation method, e.g. leave one out, k-fold or even simply divide your sample into training and validation set. Once you find the best model, please try to find another data set which is called 'BreastCancerDataTest.txt' from Cambro. In this data set, only feature variables are available. You need to apply your best model to predict the diagnostic results for them. Then calculate the accuracy of your optimal classifier.

## Task 2

In this task, we continuously work on 'gene expression' data which has been used in computer exercise 1. For more information about the data, please check the document of the previous computer exercise.

## Task 2.1 Feature extraction

Here, we do PCA on original gene data, then train classifier about 'subtype' by using principle components as extracted feature variables. Cross validation is applied to select the optimal model. You may follow the suggested procedure below:

1. Randomly (use random seed 2017) split the gene data and meta data into two parts, training data ($80\%$) and testing data ($20\%$).

2. Train a logistic regression with the first $k$ principle components. Treat $k$ as a hyperparameter. The potential range of $k$ is $1$ to $20$. Do a 10 fold cross-validation to select the optimal $k$.

3. Apply the optimal model on the testing data, and calculate the accuracy of your classifier.

## Task 2.2

Now, instead of chosing the optimal number of principle components, we apply penalized logistic regression to select the optimal subset of the first 20 principle components.

1. Apply building function 'glmnet' in package 'glmnet' to train a logistic regression with $L_1$ penalty to classify the 'subtype' given principle components.

2. Use build function 'predict' with parameter $s = 0.1$ to predict the 'subtype' in testing data set.

3. Apply building function 'cv.glmnet' to chose the optimal shrinkage parameter $\lambda$

4. Apply the optimal model on testing data and calculate the accuracy.

### Task 2.3 Variables selection

Finally, we give up principle components and directly apply penalized method on gene data. We simply apply penalized method on raw gene features and subtype. Do cross validation. Then choose the best model and find out the "best" subset of genes in a sense of accuracy of classification of subtypes.

## Task 3

In this task, we try to implement perceptron algorithm by our own coding.

### Perceptron Algorithm

- **Input**: sequence of $N$ labeled examples $< (y_1, \mathbf{x}_1), ..., (y_N, \mathbf{x}_N) >$, where $y_i = 1$ or $-1$ for $i = 1, ..., N$.
- **Initialization**: the coefficients of perceptron classifier $\mathbf{w}^{(0)}$.
- **For** $t = 0, 1, ...$

  **Stop** if no mistake by using $\mathbf{w}^{(t)}$ as coefficients of linear decision boundary.

  1. find the next mistake example w.r.t $\mathbf{w}^{(t)}$, say $(y_k^{(t)}, \mathbf{x}_k^{(t)})$ i.e.

  $$sign\left(\mathbf{w}^{(t)'}\mathbf{x}_k^{(t)}\right) y_k^{(t)} \neq 1$$

  2. Update **w** as

  $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_k^{(t)}\mathbf{x}_k^{(t)}$$

  3. Jump to next round $t + 1$

### Task 3.1

Implement this algorithm in R.

## Task 3.2

Find the R code 'DataGeneration.R' from Cambro. Please generate the data given the code, then apply your function of perceptron on the data and plot the decision boundary.