Machine Learning (Homework #2)

Due date: 12/2

1. Dimension Reduction

(a) For a K-class problem, the between-class scatter matrix is defined by: (Eq. (4.46))

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$$

where $\mathbf{m}_k = \frac{1}{N_k}\sum_{n\in\mathcal{C}_k}\mathbf{x}_n$ and $\mathbf{m} = \frac{1}{N}\sum_{i=1}^{K} N_i\mathbf{m}_i$, show that the maximum rank of $\mathbf{S}_B$ is $K-1$.

(b) Please write a program to implement dimension reduction and tackle a simple classification problem with continuous inputs by using probabilistic generative model. You are given a data file **Irisdat.xls** where each data sample contains 4 attributes:

Length of sepal, Width of sepal, Length of petal, Width of petal

This data set contains three classes of iris: **Setosa** (SET), **Virginic** (VIR), **Versicol** (VER). The number of data samples is 150. Please use the first 120 data samples as your training set, and the last 30 samples as the test set. In the training phase, you may apply the maximum likelihood theory to find model parameters:

$$\pi = \frac{N_1}{N_1 + N_2}$$

$$\boldsymbol{\mu_1} = \frac{1}{N_1}\sum_{n=1}^{N} t_n\mathbf{x}_n,\ \boldsymbol{\mu_2} = \frac{1}{N_2}\sum_{n=1}^{N}(1-t_n)\mathbf{x}_n$$

$$\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2$$

(Note that you should generalize the result to the multiclass case.)

(1) P

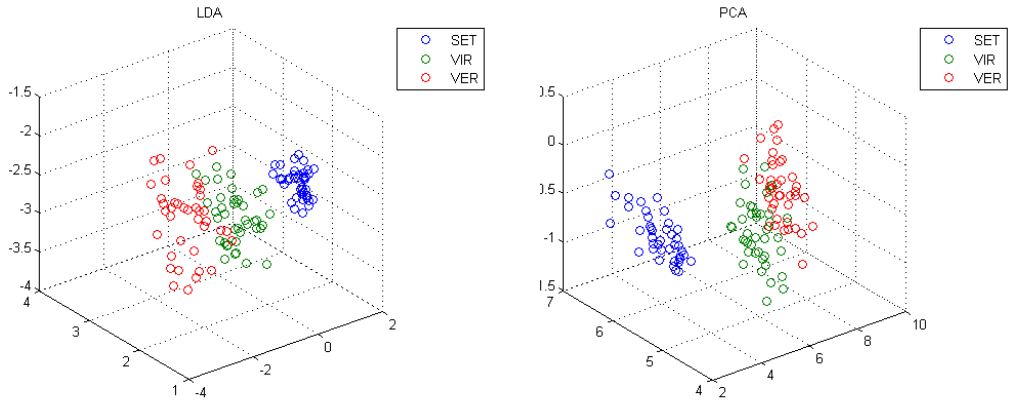
Truth\Predict	SET	VIR	VER	Total
SET	40	0	0	40
VIR	0	38	0	38
VER	0	2	40	42
Total	40	40	40	120

e show the classification chart and classification accuracy on both training set and test set in your report.

(2) Please apply the principal component analysis (PCA) for 120 training samples first. You shall reduce the dimension of training and test set to 3, 2, and 1 dimensions respectively by the bases obtained from PCA. In each case, show the classification chart and classification accuracy on both sets.

(3) Please apply the linear discriminant analysis (LDA) to 120 training samples first. You shall reduce the dimension of training and test set to 3, 2, and 1 dimensions respectively by the bases obtained from LDA. In each case, show the classification chart and classification accuracy on both sets.

(4) After reducing the dimension of data samples to 3 by using PCA and LDA, plot the data points of both training set and test set, and do some discussions.



Hint:

You may refer the following links for implementation of PCA.

http://sebastianraschka.com/Articles/2014_pca_step_by_step.html

https://plot.ly/ipython-notebooks/principal-component-analysis/#Preparing-the-Iris-Dataset

http://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579

2. Logistic Regression

You are given an KDD99 data set. This data set contains 5 classes. In this exercise, you will implement two different algorithms to construct a **multiclass logistic regression model** with the softmax transformation. The error function is formed by using the cross-entropy error function.

Note: cross-entropy error function $E(\boldsymbol{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$

Softmax function: $p(C_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$

Dataset description

KDD99 data set is used to investigate the system of network intrusion detector which is seen as a predictive model to distinguish between the "bad" connections, called intrusions or attacks, and the "good" normal connections. This database contains a standard set of data which includes a wide variety of intrusions simulated in a military network environment. You can see the details in the web: (https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html).

In this homework, we provide the file which has been pre-processed and simplified. The attribute of pre-processing data sample is described as

- Dst_bytes: number of data bytes from destination to source
- Flag: normal or error status of the connection
- Root_shell: 1 if root shell is obtained; 0 otherwise
- Su_attempted: 1 if"su root"command attempted; 0 otherwise
- Dst_host_count: number of connections to destination host
- Dst_host_srv_count: number of connections to destination host service
- Dst_host_same_srv_rate: rate of connections to same destination host service
- Dst_host_diff_srv_rate: rate of connections to different destination host service
- Dst_host_same_src_port_rate: rate of syn error in the same destination host service
- Dst_host_srv_diff_host_rate: rate of syn error in the different destination host service
- Class: Normal=0, Dos=1, Probe=2, U2R=3, R2L=4

(a) Please implement the **gradient descent algorithm** to update your parameter, and show the results in your report which should include the **learning curve** and the **test miss classification rate**. Updating rule of gradient descent:

$$\boldsymbol{w}_k^{new} = \boldsymbol{w}_k^{old} - \eta * \nabla E(\boldsymbol{w}_k)$$

(b) Please implement the **Newton-Raphson algorithm** to update your parameter, and show the results in your report which should include the learning curve and the test miss classification rate. Updating rule of Newton-Raphson algorithm:

$$\boldsymbol{w}_k^{new} = \boldsymbol{w}_k^{old} - \boldsymbol{H}_k^{-1} * \nabla E(\boldsymbol{w}_k)$$

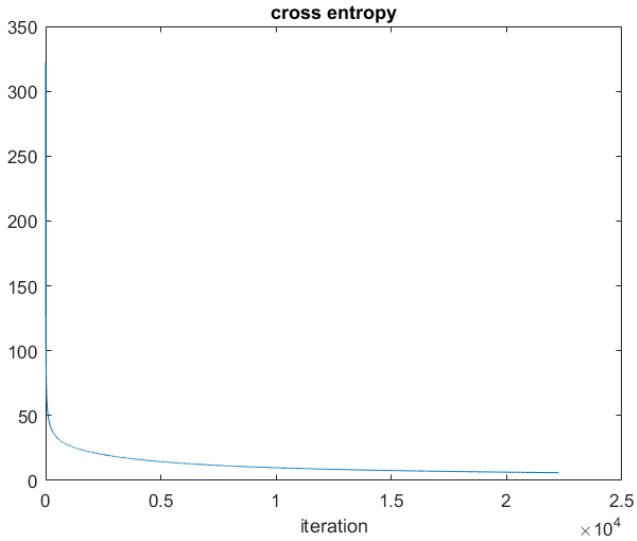
Hint:

i. You may use softmax function

$$p(C_k|\phi) = y_k(\phi) = \frac{1}{\sum_j \exp(a_j - a_k)}$$

to avoid the overflowing issue.

ii. The example of learning curve is provided as



iii. In (a), you can implement the gradient descent algorithm with a fixed iteration number or keeping the updating until the cross-entropy error function less than a threshold (it is better no less than 6). And, you need to tune the learning rate and learning epoch to make your miss classification rate the lower the better.

iv. In (b), if you encounter a NaN in cross-entropy function, just multiply a small scalar λ for the Hessian matrix .

$$\boldsymbol{w}_k^{new} = \boldsymbol{w}_k^{old} - \lambda * \boldsymbol{H}_k^{-1} * \nabla E(\boldsymbol{w}_k)$$

v. You have to implement the details including how to calculate the gradient of the error function and how to calculate the Hessian matrix by your own, and do not use toolbox.

vi. You have to deliver your source code with your report.