Assignment 2

Machine Learning, Summer term 2014, Ulrike von Luxburg

To be discussed in exercise groups on April 28-30

You can download the Matlab functions `mixGaussian1d, mixGaussian2d, genLinData` and the dataset `20Newsgroup.mat` from the course web page.

Exercise 1 (Text classification, 3+1 points) In this exercise, we apply the kNN classifier for automatically classifying user posts in a newsgroup. The 20 Newsgroups dataset is a collection of approximately 19,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups (see `20NgClasses.txt`). To simplify the work, we use a bag of words representation of documents. In this representation, the order of words are ignored and we only count the appearance of a word in a document. Your training data is a 11269 x 53975 matrix, where each row corresponds to a document and element (i, j) shows how often the word j occurs in the document i. You do not need to know the corresponding word for each feature, but they are available in `vocabulary.txt`. We want to classify documents in class 6 (`comp.windows.x`) and 8 (`rec.autos`).

(a) Classify classes 6 and 8

- Load the dataset and prepare the training set for classes 6 and 8. For simplicity, select $n = 40$ random training example from each class :

```
load('20Newsgroup.mat');
n = 40;
trIdx6 = find(y_train==6);
rand_train_6 = randperm(length(trIdx6));
% Do the same for class 8 to get trIdx8 and  rand_train_8
trList = [trIdx6(rand_train_6(1:n)) ; trIdx8(rand_train_8(1:n))];
x_train_6_8 = x_train(trList,:);
y_train_6_8 = y_train(trList);
```

- Do the same for the test data (on 40 random samples).
- Note that the train and the test data have different feature sizes. The reason is the existence of words in the test set which never appeared in the training set. You can ignore those features by simply cropping them: `x_test_cropped = x_test(:,1:size(x_train,2));`.
- Plot the training and the test error for `k=1,3,5,7,9`. Does it change in different runs?

(b) Can you train your classifier using all training points? If you use `for` loops in your classifier, the training on the whole dataset would take several hours. Try to rewrite you code using only matrix operations.

Exercise 2 (Bayes classifier, 4+4 points) In this exercise you will do the maximum likelihood and the Bayes classification by hand. First generate a simple synthetic dataset

```
[X,Y] = mixGaussian1d(1000,0.5,0.5,0,6,1,2);
```

(a) Marginals, priors and class conditional distribution

- Read the Matlab code in `mixGaussian1d.m` and describe its functionality in words.
- Plot the points `[X,Y]` in 2d (this reflects the joint distribution $P(X, Y)$).
- As your data is 1-dimensional, it is hard to interpret this plot. A better way to display the data is to plot the marginal $P(X)$. Estimate the density by its histogram:

```
figure (1); clf; hold all;
[countC,binsX] = hist(X,30);
PX = countC/size(X,1);
plot(binsX,PX,'.-');
```

- Plot the class conditional distributions $P(X|Y=1)$ and $P(X|Y=2)$ with different colors in the same figure. To make histogram bins compatible with each other, pass `binsX` as an argument to `hist`.
- Estimate priors $P(Y=1)$ and $P(Y=2)$.

(b) Maximum likelihood and Bayesian maximum a posteriori

- Based only on looking your plots, predict the labels for $X' = \{0, 1, 2, 2.5, 3, 4, 5\}$ using maximum likelihood and Bayes methods. For the Bayes method, you need to draw a new plot.
- Repeat the exercise with `[X,Y] = mixGaussian1d(1000,0.1,0.6,0,6,1,2);`.

Exercise 3 (Letter classification, 1+1 points) Assume you are going to write a two-class classifier which distinguishes the letter j from the letter k. The frequency of the letter j in a typical English text is 0.015, and for the letter k it is 0.045.

(a) Assume your input data consists only of letters j and k. Will you buy a classifier with probability of error 0.3? What would be the lowest probability of error if you do the classification without even looking at your input data?

(b) Assume your input data are all English letters and you want to separate the letter j from others. Will you buy a classifier with probability of error 0.02?

Exercise 4 (Decision boundary, 1+3 points) Generate a dataset by calling
`[X Y] = mixGaussian2d(100,0.4,0.6);`
X is a set of 2d points belonging to two different classes with labels in Y. As a prior knowledge, you know that the density of each class is a Gaussian.

(a) Compute the sample mean and sample variance for each class.

(b) Using the estimated mean and variance from part (a), find the analytic decision boundary for the Bayes classifier (see Section 2.6.3 from the pattern recognition book by Duda et al. available in the course web page).

Exercise 5 (Types of errors, 1+1+1 point) You are asked to write a spam filtering algorithm. Two types of errors can occur during your classification: A false-positive occurs when spam filtering wrongly classifies a legitimate email message as spam. While most anti-spam tactics can block or filter a high percentage of unwanted emails, doing so without creating significant false-positive results is a much more demanding task.
A false-negative occurs when a spam email is not detected as spam, but is classified as non-spam. A low number of false negatives is an indicator of the efficiency of spam filtering.

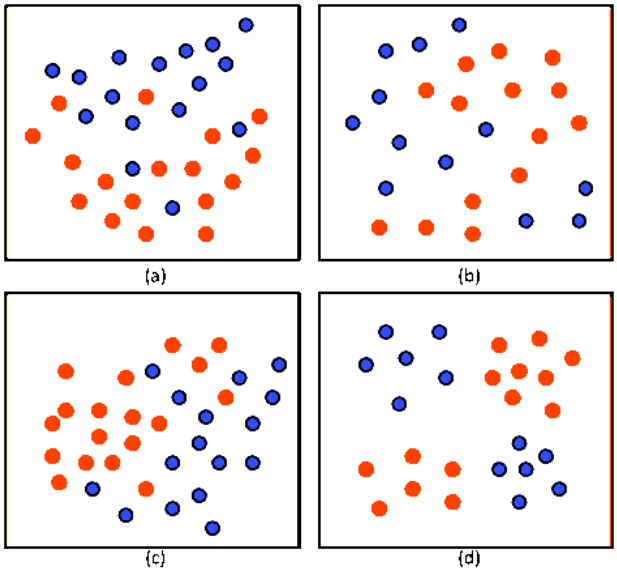
(a) Your algorithm finds 85% of spams, and miss-classifies 5% of legitimate emails. Additionally you know that about 60% of your incoming emails are spam. What are false positive, false negative and average error of you classifier?

(b) Find a classification algorithm with false positive rate 0. Find a classification algorithm with false negative rate 0.

(c) In Part 4 of Exercise 2-a (class conditional distributions), sketch the approximate Bayesian decision boundary by hand with respect to the following loss functions

- 0-1 loss
- Unsymmetric lost: $l(spam, non-spam) = 1, l(non-spam, spam) = 100$.

Exercise 6 (Decision boundary, 1+2 point) Consider the following four samples, where colors indicate class labels. Each sample is typical of an underlying distribution.



(a) Plot the decision boundaries of the 1NN classifier for the four samples.

(b) Suggest a method to solve these classification problems using regression. If you use the least square regression, how would the decision boundaries look like?

Exercise 7 (Least square regression, 1+3+2 point) In this exercise you will implement the linear least square method for regression. First generate a simple synthetic dataset

```
sigma = 0.1;
[X_train,Y_train] = genLinData(50,sigma);
[X_test,Y_test] = genLinData(30,sigma);
```

(a) Preprocessing

- Plot the input data and describe it by reading `genLinData.m`
- Preprocess the training data by concatenating 1 (for the bias term) to each training point.

(b) Classification and prediction

- Write a function `w = LLS(X,Y)` that given the training data X and the training labels Y, returns a linear classifier (vector of weights) w. X is a $n \times d$ matrix consisting n points in $\mathbb{R}^d$. Y is a vector of size n and w is a vector of size d.
- In the same figure from Step 1, plot the fitted line.
- Predict labels for `X_test` (preprocess `X_test` as you did it for the training data). Plot it in the figure from Step 1 with a different color.

(c) Find the test error, sensitivity to outliers

- Write a function `err = lossL2(Y,Y_pred)` which returns the empirical squared loss of predicting `Y_pred` instead of `Y`.
- Find the average test error for 10 different runs of your code. Plot the average test error for

  ```
  sigma = [0.01 0.1 0.4 0.9 1];
  ```

- Add an extreme outlier to your training data

  ```
  X_train = [X_train;10];
  Y_train = [Y_train;10];
  ```

  and run your code to see its effect in linear least square regression.