Homework 2, Machine Learning

K Nearest Neighbours and K-means

1 Data

In this task we use again the wine dataset from `http://archive.ics.uci.edu/ml/datasets/Wine`. This dataset has 13 features and 3 classes. Class labels are in the first column in each row.

Try using data both directly and after standardization. The data standardization can be done like this:

$$x_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \tag{1}$$

where we extract from each feature of each point the mean value of this feature and scale it with the standard deviation of the feature.

2 K Nearest Neighbours

The first task is to implement KNN algorithm. You can either:

1. Implement the model and learning algorithm by your self using your favorite programming language or
2. Use some toolkit or library and implement only the code to execute experiments.

You can do both of course if you like to.

2.1 Toolkits

For python the recommended toolkit is `scikit-learn`: `http://scikit-learn.org`. Look for examples in `http://scikit-learn.org/stable/modules/neighbors.html` and reference in `http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`.

You can also use any other toolkit that is available for your favorite programming language.

2.1.1 Experiments

Experiment with different options provided with the toolkit function. Record the cross-validation accuracies of different values of parameters or options.

2.2 Implementation

When making your own implementation the minimal work to do is to implement euclidean distance function (in Python there is the respective function in the module `scipy.spatial.distance`) and the possibility to set hyperparameter K.

2.3 Evaluation

Evaluate the results of different experiments using k-fold cross-validation. Choose either 5 or 10 folds (or try with both). In `scikit-learn` cross-validation is implemented in `http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cross_validation`. You can also make your own implementation.

3 K-means

The second task is to implement K-means clustering algorithm. You can either:

1. Implement the model and learning algorithm by your self using your favorite programming language or
2. Use some toolkit or library and implement only the code to execute experiments.

You can do both of course if you like to.

3.1 Toolkits

For python the recommended toolkit is `scikit-learn`: `http://scikit-learn.org`. Look for information in `http://scikit-learn.org/stable/modules/clustering.html#k-means` and reference in `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`.

You can also use any other toolkit that is available for your favorite programming language.

3.2 Experiments

Try with the correct cluster number $K = 3$. You can try also some other number of clusters and verify whether the data consist of three distinct Gaussian-resembling clusters or not.

3.3 Evaluation

As K-means in an unsupervised clustering algorithm then then there is no straightforward method to evaluate its results. As in our case the number of classes is small we can get a quite good overview by looking at the **confusion matrix**. The confusion matrix should contain as many rows as there are clusters and as many columns as there are true labels. In each cell should be the number telling how many time the points having a label identified by the column were put in the cluster identified by the row. When the confusion matrix is reasonable small (and this is the case now) then it gives information about how pure are the clusters, whether some true cluster is split between several induced clusters or whether some induced cluster contains several true clusters.

4 Report

The homeword submission must include the code of the programs and a short write-up (preferably in LaTeX). The write-up should include a self-contained description of the both tasks and the solution:

- What is the problem;
- Short description of the used method;
- Details of how to use your program;
- Descriptions of the experiments;
- Results of the experiments (for K-means it should include the confusion matrices).

Preferably include figures that plot the cross-validation accuracy describing the dependence on hyperparameter values. In python the module `matplotlib` can be used to generate plots. These can be saved as pdf and imported easily into LaTeX.