Assignment 9

Machine Learning: Algorithms and Theory

Prof. Ulrike von Luxburg / Diego Fioravanti / Moritz Haas
Tobias Frangen / Siavash Haghiri

Summer term 2018 — due to **June 26**

1. When you hand in your assignment you need to hand in the notebook too. Please do not write down a report with just results and/or figures. Ideally you should email it, it is easier to correct and more eco friendly, but we accept printed versions too. From now on not handing in the notebook will result in 0 points for the programming part.

2. Before the end of the course you need to present at least one of your solution in the tutorial. If you do not do that you cannot take the exam! If for any reason you cannot attend let us know, it is possible to change group or find another solution.

3. Join the class on ILIAS otherwise we cannot contact you if we need to.

Exercise 1 (Grid search vs random search, points 1 + 1 points)
In this assignment we will investigate the difference between running cross validation via grid search and via a random search. Since parts this exercise can be time consuming please do not use a 10-fold CV, just use the default for sklearn (which is 3). In `X_test`, `y_test` and `X_train`, `y_train` you can find the "breast cancer wisconsin dataset" already loaded. In X each row corresponds to a 30 different features and in Y you will find if the cancer is benign or malignant. For more info look `http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html`

(a) Using a SVM with RBF kernel, run a cross validation with grid search (use `GridSearchCV`) with the parameters you find in the variable `parameters`. Which C works best?

(b) Using a SVM with RBF kernel, run a cross validation with random search (use `RandomizedSearchCV`) with the parameters you find in the variable `param_dist` using `n_iter` = 100 and `n_iter`=1000. Which C works best? Does it work better than a grid search ? Why do you think an exponential distribution is chosen for generating the parameters instead of a normal one?

You can read more about random search cross validation here `http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf`, if you are interested.

Exercise 2 (Implement Multi-dimensional scaling (MDS), 3+2+2 points)

(a) Define a function `MDS` which takes a symmetric $n \times n$ distance matrix `D` and the number of dimensions to embed into `d` as input and returns the embedding `X_embedded` which is a $n \times d$ dimensional matrix. Try to avoid loops in your implementation and use matrix operations instead.

(b) Load the symmetric $n \times n$ distance matrix from `eurodist.csv`. This gives you the road distances between 21 European cities where the matrix entry in the i-th row and j-th column corresponds to the distance between city i and j, i.e. the distance between $i = 1$, Athens, and $j = 2$, Barcelona, is 3313km along the shortest road. Apply `MDS` with $d = 2$ to embed the cities into $\mathbb{R}^2$ and plot it. For every point add the names of the cities.

(c) In this task we will come back to the USPS data set again. In `X` and `y` you can find the some of the USPS dataset digits already loaded. They correspond to the test image for the digits $0, 3, 6, 9$. For this subsample of the USPS data set perform MDS with $d = 2$. You have to compute the distance matrix first. For this you can use `euclidean_distances`. Make a scatter plot to see how the digits distribute in the embedding in $\mathbb{R}^2$. Use different colors to plot different digits.

Exercise 3 (Implement Isomap, 5+2 points)

(a) Define a function `Isomap` which takes a symmetric $n \times n$ distance matrix `D`, parameter `k` for the KNN-graph construction and the number of dimensions to embed into `d` as input. Return the embedding `X_embedded` in $\mathbb{R}^d$ via Isomap.

Hint: `https://networkx.github.io/documentation/networkx-1.10/tutorial/tutorial.html` is a good place to start for python and graphs. Remember that you want the path lenght not only the path, so `all_pairs_dijkstra_path` does not do what you want.

(b) Apply Isomap to the subsamples USPS data set from exercise 1 task (c) with $d = 2$. Vary $k \in \{5, 20, 50\}$ and plot the resulting embeddings. Comment on which embedding seems to be the most reasonable to use for clustering algorithms. What happens for $d = 1$?

Exercise 4 (k-means and kernel k-means, 2 + 2 points)

(a) Show that the cluster centers are the best representatives of a cluster in the sense that for any cluster the cluster center m satisfies equation (1). (See the slides for understanding notation).

$$\sum_{i \in C_k} \|X_i - m_k\|^2 \leq \sum_{i \in C_k} \|X_i - X_j\|^2, \forall j \in C_k \tag{1}$$

(b) Prove that on any given data set of n points in $\mathbb{R}^{\mathsf{d}}$, the k-means algorithm terminates after a finite number of steps (hint: as an intermediate step, show that in each iteration of the while loop, the objective function cannot increase; you may use the result from part a.)