Data Mining: Practical Assignment #5

Due on Thu & Fri, June 05-06 2014,

Task 1

In the MIN-CommSy you find several files with 2D data (`triangle.dat`, `clusters.dat`, `capitals.dat`) and 3D data (`box.dat`, `sphere.dat`). Also available is the program `data_create.py` used to create the data. Explore how the given SOM represents these data. Try different network topologies (1D and 2D) and sizes. In particular, perform the following experiments:

1. Observe how a 1D network – a long "chain" of units – folds itself into the data `triangle.dat`.

2. Observe how a 2D network – a "grid" of units – folds itself into the data `box.dat`.

3. Use the implemented k-means (given in `KMEANS.py`) and compare its behaviour to a SOM on the data `clusters.dat`. Set $k = 4$ for both, SOM and k-means.
   Alternatively implement an online-k-means (by using the SOM implementation and setting the neighbourhood interaction width to zero throughout learning) and compare its behaviour to a SOM on the data `clusters.dat`. Set $k = 4$ for both, SOM and k-means.

4. Solve the travelling salesman problem (TSP) on the data `capitals.dat`, i.e. our travelling salesman should visit all EU capitals along the shortest path. To this end, use a 1D network with as many units as there are capitals given (= 44). Will learning correctly lead to a solution where each city is covered by one unit?

   To explain the network behaviour, consider that capitals in the east of Europe are denser together while in the west they are more dispersed. In theory (assuming e.g. infinitely many data) the network behaviour can be characterized as follows. Let ρ_{data} be the density of the data and ρ_w the density of the units' representations (centroids) in data space after learning. Theory predicts the relation:

   $$\rho_w \approx \rho_{data}^{d/(d+2)}$$

   where d is the dimensionality of the data space and the "2" comes from the L2 norm (Euclidean distance / squared error) that is minimized in the SOM "cost function".

Comments about `gnuplot`. All given data files can be plotted with gnuplot. The SOM program also writes the learnt weights (centroids) like data into a file `out.dat`. Furthermore, the links between the centroids are drawn by gnuplot commands produced in the file `out.gnu`.
In the shell of gnuplot, the most important commands are `plot`, which is for 2D, and `splot`, which is for 3D. A command may look like this:

```
splot "sphere.dat" with dots, "out.dat" with points
```

Gnuplot commands, which are written by the SOM program into a the file `out.gnu` can be loaded as:

```
load "out.gnu"
```

This plots the connecting lines between neighboring SOM units.
For the TSP problem, visualisation of the data with labels can be done via:

```
unset key
plot "capitalsnames.dat" with labels
```

Gnuplot under the operating system Windows: The homepage of gnuplot is `www.gnuplot.info`. From there you can get via SourceForge the file `gp460win32.zip`. Unzip this file, and in the sub-directory `gnuplot\binary` you find the executable `wgnuplot.exe`. Execute this and you have it running. In gnuplot you can print the current working directory via `pwd`, and you can change to the directory of your data like in the following example:

```
cd "C:/Users/weber/Desktop/DAMI/SOM_data/"
```

Task 2

Now solve the TSP with the genetic algorithm given in `GA.py`. Look into the code to recapitulate how this method works.

1. Describe the genotype (representation of a problem solution).
2. Describe the following steps of the algorithm:
   (a) Cost function. How does this differ from the objective of the SOM?
   (b) Parent selection.
   (c) Crossover operator.
   (d) Mutation operator.
   (e) Survivor selection.
3. In which of these steps and for which purpose is randomness being used?
4. Vary the parameters in the program to optimize the solution. What is your experience?
5. Can the costs increase during learning (can we escape from local minima if not allowing increasing costs)?
6. Which of these steps need to get changed if the GA should be applied to a different problem?