DATA.STAT.770 Dimensionality Reduction and Visualization, Spring 2021, Exercise set 8

Problem E6: Self-Organizing Map

Part 1: Self-organizing map versus vector quantization. Prove that the update rule of the self-organizing map (SOM; lecture 8) becomes the same as the update rule of vector quantization when the number ot iterations t approaches infinity.

Part 2: Self-organizing map of kidney disease patients. If you have Matlab available, familiarize yourself with the free self-organizing map toolbox for Matlab (SOM Toolbox) available at `http://www.cis.hut.fi/somtoolbox/` or with the self-organizing map functionality of Matlab's commercial Neural Network Toolbox.

If you have R available, you can alternatively familiarize yourself with the "kohonen" package for R that implements self-organizing maps.

If you would like to use some other software than R or Matlab, familiarize yourself with the older command-line tool SOM-Pak available at `http://www.cis.hut.fi/research/som_lvq_pak.shtml`.

With the software that you chose, produce a map of the Chronic Kidney Disease data set (a data set of 400 patients from a hospital in India) available at `http://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease`; use the "chronic_kidney_disease_full.arff" file, the data are on rows 146 onwards.

- This data set has both 14 categorical (nominal) attributes and 11 real-valued (numeric) attributes; use the 11 real-valued attributes as input features to the SOM and use the class attribute ("ckd" or "notckd") as a target label.
- Some patients do not have all of the real-valued attributes (marked as "?"), exclude those patients before running the SOM.
- Before running the SOM, normalize each feature except the class attribute by dividing it with its standard deviation.
- Create a 5×8 hexagonal grid for the SOM and train the SOM using the data; do not use the class labels in training.
- Plot how the "blood pressure" and "red blood cell count" attributes vary over the prototype vectors of the resulting SOM.
- Plot also what is the majority class of data at each prototype vector ("ckd" or "notckd"). Use the prediction function of the SOM package to do this.

Problem E7: Laplacian Eigenmap, Isomap, and Precision and Recall

Part 1: Laplacian Eigenmap and Isomap for swiss roll data. Using the dimensionality reduction toolbox (discussed in the previous Exercise Set 7), load the swill roll data set (provided in Exercise Set 7) and reduce it to two dimensions using three methods: Laplacian Eigenmap, Isomap, and Locally Linear Embedding. Plot the resulting pictures and discuss the differences.

Part 2: Precision and recall. Suppose for each data point i, $i = 1, \ldots, N$, a true neighborhood $N(i)$ is defined, which is a set of other data points that are neighbors of i. Then, if a retrieval method retrieves a different set $R(i)$ from the visualization for each i, representing data points that visually look like neighbors of i, the quality of the retrieval can be measured by *average precision* and *average recall*, defined as

$$AveragePrecision = \frac{1}{N} \sum_{i=1}^{N} \frac{|N(i) \cap R(i)|}{|R(i)|}$$

$$AverageRecall = \frac{1}{N} \sum_{i=1}^{N} \frac{|N(i) \cap R(i)|}{|N(i)|}$$

The value of average precision and average recall depend a) on the data set and its visualization, b) on how the true neighborhoods $N(i)$ are constructed, and c) on how the retrieval constructs the retrieved sets $R(i)$.

Evaluate the precision and recall for Laplacian Eigenmap and Isomap on the swiss roll dataset. We will make the following choices.

a) We will use the swiss roll dataset, and the Laplacian Eigenmap and Isomap visualizations.

b) We will define the true neighborhood $N(i)$ as the 5 closest data points to i in the original feature space.

c) We will consider a simple scenario where $R(i)$ contains all points whose distance to i is smaller than a threshold T.

Use several thresholds T, and list the resulting values of average precision and average recall as a table over T or as plots over T.