Assignment 9

Machine Learning, Summer term 2013, Ulrike von Luxburg

Solutions due by June 17

Exercise 1 (Clustering using k-means, 5 points) In this exercise, you will try the k-means clustering algorithm to cluster the handwritten digits data from USPS dataset. Select 200 samples from each of digits $\{1, 3, 4, 6\}$.

- Use the matlab command `kmeans` to cluster the selected digits. In the options for the command, set `'replicates'` to 5 and `'Display'` to 'final'. This would show the value of local optimum in 5 different random initialization of centers.
- How would you evaluate the result of clustering? Discuss the difficulties. Evaluate the result of k-means with your suggested method.
- In real world, you do not have access to the ground-truth cluster of your data. How would you then assess the quality of the clustering?
- Read the documentation of command `kmeans` and describe what the parameter `'distance'` does. Try the clustering with two other distances and evaluate their results.
- Describe a scenario, in which the distance `'cityblock'` can perform better than the usual Euclidean distance `'sqEuclidean'`.

Exercise 2 (Implementing spectral clustering, 7 points) In this exercise you implement the unnormalized and normalized spectral clustering. This only needs few lines of matlab code. To evaluate your code, use the data from Exercise 1.

- Build a k nearest neighbor graph using the code provided in Assignment 8 (`buildKnnGraph`). A proper value for k would be in the range $k \in \{5, \cdots, 50\}$. Set the name of the graph adjacency matrix to W. The output W is the adjacency matrix with edge weights $W(i,j) = \|x_i - x_j\|$. You need to convert it to a similarity matrix. It is common to use Gaussian weights (also called Gaussian kernel) $S(i,j) = exp(-\|x_i - x_j\|^2/2\sigma^2)$, where σ is the width of the Gaussian. You can convert W to a similarity matrix by

  ```
  W(W~=0) = exp(-W(W~=0).^2/(2*sigma^2));
  ```

  You should set the kernel width σ in a range corresponding to the average edge weights in W (around 1000 in this data set).
- Form the unnormalized ($L_u = D - W$) graph Laplacian matrix. Useful commands are `diag, sum`.
- Find the eigenvalues and eigenvectors of the unnormalized Laplacian matrices using `[V,D] = eig(full(L))` [1]. Plot the points using the second, third and forth eigenvectors of the Laplacian by `scatter3(V(:,2),V(:,3),V(:,4),...)`. This is called the spectral embedding.
- Cluster the first four eigenvectors of the Laplacian (first four column of V) using `kmeans`.
- Evaluate the result of clustering.

[1] An alternative for large sparse matrices is to use `[V,D] = eigs(L,4,'sm');`, which returns the 4 smallest eigenvalues and corresponding eigenvectors.

- Repeat the procedure using normalized Laplacian matrix ($L_{norm} = D^{-0.5} L_u D^{-0.5}$). A useful command is `sqrt`. Evaluate the result of clustering.

Exercise 3 (Spectral clustering demo, 2 points) Download the spectral clustering demo from the course webpage. Run `DemoSpectralClustering`. Play with different data sets and parameters, and try to understand plots. In this demo, the similarity scores between vertices x and y is defined as $s(x_i, x_j) = exp(-\|x_i - x_j\|^2/2\sigma^2)$, where σ is the kernel width.
Select a data set. Vary the number of neighbors from low to high. For which range of k do the clusters in the embedding look "well separated"?
Describe the interplay between the kernel width σ and the number of neighbors k. What would happen if we choose a large k, but small σ?

Exercise 4 (Your own exam questions, 6 points) In this exercise, everybody is supposed to come up with suggestions for exam questions. This is a good way to recap the material and to study for the exam itself.
Put yourself in our place! We don't want to ask stupid questions but "nice questions". In general, written exams contain three types of questions:

- Questions which are just about **reproducing** knowledge.
- Questions for testing whether the person **understands** the concepts and can apply them to simple situations.
- Questions that requires to **transfer** knowledge to new situations.

Your task is now to design at least three exam questions, one of each type. The more questions you come up with, the better. Please enter your questions to the following webpage:

`http://www2.informatik.uni-hamburg.de/ML/ML-AL-2013/questions.php`

Check the questions already published in the website to avoid posting repeated questions!
At the end of the course, these questions can help everybody to prepare for the exam. So take your time to invent good questions! We promise that we are going to use at least one of the questions in the real exam.