Machine Learning	Prof. Dr. Bernt Schiele
	Dr. Paul Swoboda & Dr. Gerard Pons-Moll
Exercise Sheet 10 - 07.01.2019	due: 14.01.2019, 14:15

Exercise 18 - Clustering

Please implement the clustering algorithms discussed in the lecture.

- **(4 Points)** Complete the function `kmeans_diy` for the Llyod's algorithm for k-means clustering.
- **(4 Points)** Complete the function `spectralclustering_diy` for the spectral clustering algorithm with unnormalized graph Laplacian:

  $$L = D - W$$

  where D is the `degree matrix` and W is the `weighted adjacency matrix` of the graph. The similarity graph here is set to be the fully connected type, i.e., we construct all points with positive similarity with each other and we weight all edges by

  $$s(x_i, x_j) \ = exp(-\|x_i - x_j\|^2/\gamma)$$

  where x_i, x_j denote the i-th and j-th data points in the dataset, γ is fixed to be 1 in this exercise.
- **(3 Points)** Now apply your clustering code (k-means and spectral clustering) to the provided dataset `ex9_data` and visualize your results by running the script `ex9_main.m`. The data file contains two datasets `data1` and `data2`, where each includes two groups of data points X_1, X_2in R^2, you need to apply the clustering on `data1` and `data2` separately. What's your observation and conclusion? (Written solution on paper.)

Hints:

- For spectral clustering, you can use *eig* function to compute the eigen vector and eigen value of a matrix.
- You are recommended to implement a function `compute_pairwise_distance` to compute pairwise distance between two datasets which can be shared by both of your clustering algorithms' code.

Solution:

a. **(3 Points)** Example code for k-means clustering

```
function [ cur_group ] = kmeans_diy( input_data, num_cluster )
% your implementation of kmeans clustering Lloyd's algorithm
%   input_data: data to be classified, size:NxD
%   num_cluster: specified number of cluster
%   cur_group: obtained label from clustering, size: Nx1
%   where N is the number of data and D is the dimension of each data point

num_data = size(input_data,1);

% initialize value of centroid
p = randperm(num_data);
center = input_data(p(1:num_cluster),:);
```

```
    % iterative loop to update assignment
    stop = 0;
    cur_group = zeros(num_data,1);
    num_itr = 0;
    max_itr = 5000;
    while stop == 0
        % compute new grouping
        dist = compute_pairwise_distance(input_data,center);
        [¬,new_group] = min(dist,[],2);

        % compute new center
        for i = 1:num_cluster
            center(i,:) = mean(input_data(cur_group==i,:),1);
        end

        % stopping criteria
        if new_group == cur_group | num_itr > max_itr
            stop = 1;
        else
            cur_group = new_group;
        end

        num_itr = num_itr + 1;
    end

    end
```

b. **(3 Points)** Example code for spectral clustering

```
    function [ cur_group ] = spectralclustering_diy( input_data, num_cluster )
    %your implementation of (unnormalized spectral clustering)
    %   input_data: data to be classified, size:NxD
    %   num_cluster: specified number of cluster
    %   cur_group: obtained label from clustering, size: Nx1
    %   where N is the number of data and D is the dimension of each data point

    % compute adjacency matrix
    dist = compute_pairwise_distance(input_data,input_data);
    gamma = 1;
    W = exp(-1/gamma*dist);
    % compute unnormalized laplacian
    D = diag(sum(W,2));
    L = D - W;
    % compute eigenvectors
    [U,¬] = eig(L);
    U = U(:,1:num_cluster);
    % perform k-means clustering
    cur_group = kmeans_diy(U,num_cluster);

    end
```

c. **(2 Points)** You can see the plots and explanations for the two clustering methods in Figure 1.

Exercise 19 - Distances in High Dimensions

Let X and Y be two independent $\mathbb{R}^d$-valued random variables with expectations

$$\mathbb{E}[X] = \mu_X = (\mu_{X,1} \dots \mu_{X,d})^T, \qquad \mathbb{E}[Y] = \mu_Y = (\mu_{Y,1} \dots \mu_{Y,d})^T,$$

and covariances

$$\mathbb{E}[(X - \mu_X)(X - \mu_X)^T] = \sigma_X^2 \mathbb{1}, \qquad \mathbb{E}[(Y - \mu_Y)(Y - \mu_Y)^T] = \sigma_Y^2 \mathbb{1}, \qquad \sigma_X^2, \sigma_Y^2 > 0,$$

with $\mathbb{1}$ denoting the identity matrix in $\mathbb{R}^d$.

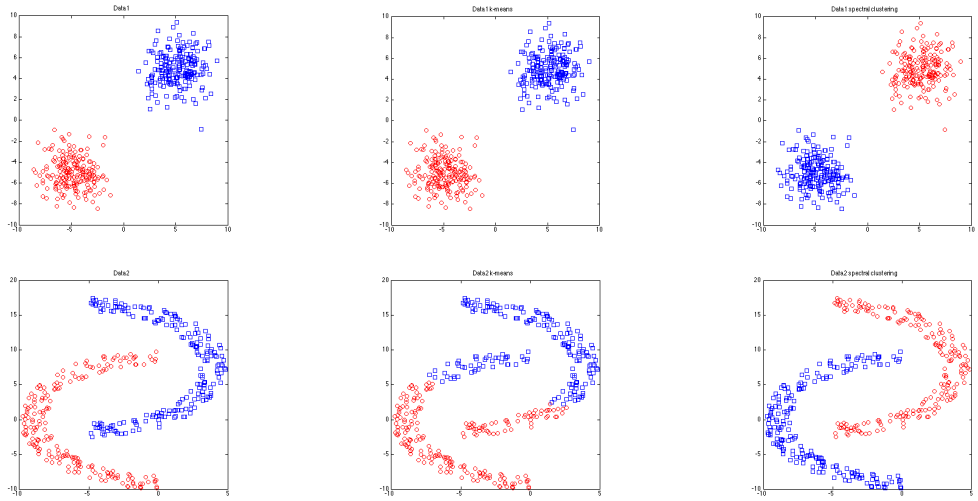

Figure 1: For the first case, both clustering algorithms perform relative well while in the latter case, spectral clustering outperforms the k-means clustering. These two settings is an example to show k-means algorithm tends to find sphere-like clusters in the data and has trouble to find non-sphere clusters whereas the spectral clustering does not have such limitation.

a. **(3 Points)** Show that the expected squared Euclidean distance of X and Y is given by

$$\mathbb{E}\left[\|X-Y\|^2\right] = d\sigma_X^2 + d\sigma_Y^2 + \|\mu_X - \mu_Y\|^2.$$

b. **(1 Point)** Suppose $\sigma_X^2 > \sigma_Y^2$ and consider the limit $d \to \infty$, assuming that $\|\mu_X - \mu_Y\|^2 = C$, where C is a positive constant independent of d. Let Z be a random variable with the same distribution as X which is independent of X. Using the result in b., compare $\mathbb{E}\left[\|X-Y\|^2\right]$ to $\mathbb{E}\left[\|X-Z\|^2\right]$.

Hints:

a. Use that $\|X-Y\|^2 = \|(X-\mu_X)-(Y-\mu_Y)+(\mu_X-\mu_Y)\|^2$.

b. Exploit the linearity of $\mathbb{E}$.

Solution:

a.

$$\begin{aligned}\mathbb{E}[\|X-Y\|^2] &= \mathbb{E}\left[\|(X-\mu_X)-(Y-\mu_Y)+(\mu_X-\mu_Y)\|^2\right] \\ &= \mathbb{E}\left[\left\|\widetilde{X}-\widetilde{Y}\right\|^2\right] + \mathbb{E}[\|\mu_X-\mu_Y\|^2] + 2\mathbb{E}\left[\left\langle \widetilde{X}-\widetilde{Y}, \mu_X-\mu_Y\right\rangle\right],\end{aligned}$$

where $\widetilde{X} = X-\mu_X$, $\widetilde{Y} = Y-\mu_Y$. Since $\|\mu_X-\mu_Y\|^2$ is non-random, we may drop $\mathbb{E}$. Next, using linearity of $\mathbb{E}$ as indicated in the hint,

$$\begin{aligned}\mathbb{E}[\|X-Y\|^2] &= \mathbb{E}\left[\left\|\widetilde{X}-\widetilde{Y}\right\|^2\right] + \|\mu_X-\mu_Y\|^2 + 2\left\langle \mu_X-\mu_Y, \mathbb{E}[\widetilde{X}]-\mathbb{E}[\widetilde{Y}]\right\rangle \\ &= \mathbb{E}\left[\left\|\widetilde{X}-\widetilde{Y}\right\|^2\right] + \|\mu_X-\mu_Y\|^2 \qquad (*),\end{aligned}$$

since $\mathbb{E}[\widetilde{X}] = \mathbb{E}[\widetilde{Y}] = 0$. We now expand

$$\begin{aligned}\mathbb{E}\left[\left\|\widetilde{X}-\widetilde{Y}\right\|^2\right] &= \mathbb{E}\left[\left\|\widetilde{X}\right\|^2\right] + \mathbb{E}\left[\left\|\widetilde{Y}\right\|^2\right] - 2\mathbb{E}\left[\left\langle \widetilde{X}, \widetilde{Y}\right\rangle\right], \\ &= \mathbb{E}\left[\sum_{i=1}^d \widetilde{X}_i^2\right] + \mathbb{E}\left[\sum_{i=1}^d \widetilde{Y}_i^2\right] - \mathbb{E}\left[\langle X-\mu_X, Y-\mu_Y\rangle\right].\end{aligned}$$

Independence of X and Y implies that X and Y are uncorrelated, hence the last term equals zero. Using that for $i = 1, \ldots, d$,

$$\begin{aligned} \mathbb{E}\left[\widetilde{X}_i^2\right] &= \mathrm{Var}[\widetilde{X}_i] &= \mathrm{Var}[X_i] &= \sigma_X^2, \\ \mathbb{E}\left[\widetilde{Y}_i^2\right] &= \mathrm{Var}[\widetilde{Y}_i] &= \mathrm{Var}[Y_i] &= \sigma_Y^2, \end{aligned}$$

and applying the last hint, it follows that

$$\begin{aligned} \mathbb{E}\left[\textstyle\sum_{i=1}^d \widetilde{X}_i^2\right] &= \textstyle\sum_{i=1}^d \mathbb{E}\left[\widetilde{X}_i^2\right] &= d\sigma_X^2, \\ \mathbb{E}\left[\textstyle\sum_{i=1}^d \widetilde{Y}_i^2\right] &= \textstyle\sum_{i=1}^d \mathbb{E}\left[\widetilde{Y}_i^2\right] &= d\sigma_Y^2. \end{aligned}$$

Combining this result with $(*)$ then proves the statement.

b. According to b., we have $\mathbb{E}\left[\|X - Z\|^2\right] = 2d\sigma_X^2$. One computes

$$\mathbb{E}\left[\|X - Z\|^2\right] - \mathbb{E}\left[\|X - Y\|^2\right] = d(\sigma_X^2 - \sigma_Y^2) - C.$$

Now as $d \to \infty$, the expected squared distance of X and Z dominates that of X and Y, even though X and Z have the same distribution.