Machine Learning Undergrad Course@SUTD
Assignment 3

K-means and some simple math

Due on Tuesday, October 20, 2015

upload all files for all tasks in one zip-(or .tar.gz)-file. for the images provide a description which image is for what, e.g. give the descriptions in one file which contains all other task results (equations, textual explanations) and descriptions for all images. best is one long pdf (or word) file with embedded images, equations, textual explanations. Also provide a short textual explanations what you did and why.

1. generate a function which draws N samples from 3 gaussians in 2-dimensional feature space with respective probabilities p_1, p_2, $p_3 = 1 - p_1 - p_2$ such that the gaussians are restricted to the space $(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, x_2 > 0$ with more details given below (3P)

   - The variances are assumed to be diagonal matrices, that means we have $\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$. Therefore each gaussian has 5 parameters: p_i, $\mu_i \in \mathbb{R}^2$, Σ_i with its diagonal shape.
   - for each sample in order to draw it, draw first a random choice in order to decide from which gaussian it does come, then draw it from the respective gaussian
   - How to achieve the restriction to $(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, x_2 > 0$? The simplest (and sometimes wasteful) way to achieve that, is rejection sampling. Rejection sampling is: draw a data point, and throw it away if it is not in $(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, x_2 > 0$, until you have the desired number of data points
   - suggestion for generation interface is:

   ```
   def gensomemixturedata(numdata, probs,means,sigmadiags)
       ...
       return x
   ```

   with probs being a 1×3 numpy matrix (i-th entry $probs[i]$ is p_i), means being a 2×3 numpy matrix with $means[:, i]$ being the mean for gaussian $\#i$, sigmadiags being a 2×3 numpy matrix with $sigmadiags[k, i]$ being the σ_k^2 for gaussian $\#i$.

2. k-means (10P)

   (a) Implement k-means, such that it terminates, when the objective does not change by more than $1e-4$. (3P)

   (b) draw for the three gaussian model with $p_1 = 0.15, p_2 = 0.3$ 2000 samples in two dimensions. Further parameters: $\mu_1 = (3, 3)^\top$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\mu_2 = (6, 3.6)^\top$, $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$, $\mu_3 = (5.1, 9)^\top$, $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1.5 \end{pmatrix}$
   Save these points for a later re-usage. (0.5P)

   (c) run k-means algorithm on them with $k = 2, 3, 4, 8, 16$ centers, plot and save results showing the points and the cluster centers for $k = 2, 3, 4, 8$ (with different colors for points and centers). Hand over the graphics to us. (1.5P)

(d) now iterate 10-times **for** $k = 2, 3, 4, 8, 16$: take the saved samples from above (do not draw them randomly), in each run initialize the points randomly by the k-means++ initializer, run k-means, save the positions of cluster centers, and the value of the objective function when minimization stopped (2P)

(e) In 2d one has computed a repetition of 10 times of clusterings for $k = 2, 3, 4, 8, 16$ (so 50 clusterings).

For each of the 10 repeated clusterings (and for fixed k) one can compute an average of the pairwise distances of cluster centers (over all pairs of cluster centers):

$$d_1(k) = \frac{2}{k(k-1)} \sum_{j=1}^{k} \sum_{l=1}^{j-1} \|z_j - z_l\|^2$$

Now one pls compute for each $k = 2, 3, 4, 8, 16$ separately an average over the 10 repetitions of $d_1(k)$ - this gives a graph as a function of k. This gives one averaged distance. Lets compute for each $k = 2, 3, 4, 8, 16$ a second average, namely the average over the 10 repetitions of the objective functions at the time when k-means stopped. (2P)

(f) print these two average scores in two separate graphics as a function of $k = 2, 3, 4, 8, 16$, hand over the graphics to us. Note the variation depending on the random choice of initialized centers. This is due to the algorithm being non-convex (1P)

(g) hand over the code to us

3. k-medoids (9P)

(a) Implement k-medoids (3P)

(b) take your saved old data from the very first draw. Let be $x^{(1)}$ the first dimension of your datapoint x, and $x^{(2)}$ the second dimension of your datapoint x, compute $mx^{(1)} = \max_{x \text{ is a point in your data}} x^{(1)}$, $mx^{(2)} = \max_{x \text{ is a point in your data}} x^{(2)}$ (1P)

(c) turn your data points into three dimensional histogram data by mapping each datapoint as

$(x^{(1)}, x^{(2)}) \mapsto (\frac{x^{(1)}}{mx^{(1)}+0.01}, \frac{x^{(2)}}{mx^{(2)}+0.01}, 1 - \mathbf{0.5}\frac{x^{(1)}}{mx^{(1)}+0.01} - \mathbf{0.5}\frac{x^{(2)}}{mx^{(2)}+0.01})$ (1P)

for each of $k = 2, 3, 4, 8, 16$ choose randomly initializing cluster centers from your data. save these initializers (0P)

(d) run k-medoids algorithm on your three dimensional upgraded data with $k = 2, 3, 4, 8, 16$ centers with the same distances function as for k-means - squared euclidean distance, and the saved center initializers

Plot the results for the data as a function of the first two dimensions of your upgraded data (note that for each data point the third dimension is a fixed function of the first two according to above mapping, so it is not a degree of freedom) (2P)

(e) run k-medoids algorithm on your three dimensional upgraded data with $k = 2, 3, 4, 8, 16$ centers with the distances function being χ^2-distance: $d(x_1, x_2) = \sum_{d=1}^{D} \frac{(x_1^{(d)} - x_2^{(d)})^2}{(x_1^{(d)} + x_2^{(d)})}$, and the saved center initializers

treat in the χ^2-distance the case $\frac{0}{0}$ as zero.

Plot the results for the data as a function of the first two dimensions of your upgraded data. (2P)

4. derive the solution for $\min_z \frac{1}{n}\|x_i - z\|_2^2$ by setting the derivative to zero. (1P)

5. compute some derivatives with respect to vector x: either compute the gradient $\nabla f(x)$ or the linear mapping $Df(x)[h]$ which maps a direction vector h onto the directional derivative $Df(x)[h]$ o function f at point x (6P)

(a) $f(x) = x^\top A z z^\top B x$, $x \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d}, B \in \mathbb{R}^{d\times d}, z \in \mathbb{R}^{d\times 1}$

(b) $f(x) = x^\top A x x^\top v$, $x \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d}, v \in \mathbb{R}^{d\times 1}$

(c) $f(x) = (x^\top A x)^3$, $x \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d}$

(d) $f(x) = \|Ax\|_2$, $x \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d}$

(e) $f(x) = \|Ax\|_1$, $x \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d}$

(f) $f(x) = x^\top v\, cos(x^{(1)})$,where $x^{(1)}$ is the first dimension and $v \in \mathbb{R}^{d\times 1}$