EE6132: Advanced Topics in Signal Processing

Programming Assignment - 2 : K-means, EM, and regression algorithms

Due date: April 11th 2016, 11:55pm IST

Note:

1. For any questions, please schedule a time with TAs before deadline according to their convenience. Please use moodle dicussion threads for posting your doubts and also check it before mailing to TAs, if the same question has been asked.
   (a) Problem1, Vijay vijay.ap@ee.iitm.ac.in
   (b) Problem2, Prakruti prakruti.gogia@gmail.com
   (c) Problem3, Sharath ee15s050@ee.iitm.ac.in, Vishnu ee12d038@ee.iitm.ac.in
   (d) Problem4, Sapana ee15s300@ee.iitm.ac.in
   (e) Problem5, Anil ee15s055@ee.iitm.ac.in
2. Submit a single zip file in the moodle named as PA2_Rollno.zip containing the report and the folders containing corresponding codes.
3. Read the problem fully to understand the whole procedure.
4. Late submissions will be evaluated for reduced marks and for each day after the deadline we will reduce the weightage by 10%.

Problem-1: **[20 marks]** In this problem, we will study the K-means clustering algorithm. Given a set of data points $\{\mathbf{x}_j\}_{j=1}^{P}$, K-means clustering aims to assign every point to one of the clusters $\mathcal{C}_i$ so as to minimize the sum of distance of each point to its cluster centre $\boldsymbol{\mu}_i$. We will take the 3D RGB colour value as our data point based on which we segment an image by assigning each of its pixels to different clusters. In our case, $\{\mathbf{x}_j\}_{j=1}^{P}$ is the set of all pixel colour values.

Steps:

0. Fix the desired number of clusters K.

1. Initialize random values (picked from the input set) to the mean of the K clusters, $\{\boldsymbol{\mu}_i^{(1)}\}_{i=1}^{K}$, where $\boldsymbol{\mu}_i^{(1)} \in [0 - 255]^3$ and the superscript (t) denotes tth iteration.

2. Assign every pixel $\mathbf{x}_j$ of the image to the cluster which has its mean located closer to this point compared to the means of other clusters, i.e. each cluster assignment is as follows:

$$\mathcal{C}_i^{(t)} = \{\mathbf{x}_j : \|\mathbf{x}_j - \boldsymbol{\mu}_i^{(t)}\|_2 \leq \|\mathbf{x}_j - \boldsymbol{\mu}_k^{(t)}\|_2, \forall k, k \neq i\} \tag{1}$$

   The cost at this iteration is given by

$$d^{(t)} = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{C}_i^{(t)}} \|\mathbf{x}_j - \boldsymbol{\mu}_i^{(t)}\|_2. \tag{2}$$

3. Generate and store a segmented image by assigning the colour value of each pixel as the mean value of the cluster it is assigned to. Store the number of changed assignments (i.e. the number of pixels which are assigned to one cluster in the previous iteration and a different cluster in this iteration) in the variable $s^{(t)}$. (For $t = 1$, let $s^{(t)} =$ number of image pixels.) If $s^{(t)} == 0$ or $t == 100$, stop.

4. Update the mean of all clusters using the newly assigned colour values.

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{1}{|\mathcal{C}_i^{(t)}|} \sum_{\mathbf{x}_j \in \mathcal{C}_i^{(t)}} \mathbf{x}_j \tag{3}$$

5. Repeat 2 to 4.

Questions:

(a) [**4 marks**] We will segment an image of a colour spectrum `spectrum.png` in which the colours vary in the horizontal direction from red to violet. Fix $K = 7$. Pick the initial mean values from pixels equally spaced in the horizontal direction in the centre row of the image. Show the segmented image.

(b) [**5 marks**] Repeat (a) with the initial mean values randomly picked from the input image itself. Show the segmented images with two different random initializations. Compare with the segmented image from (a). Comment on the results.

(c) [**10 marks**] Run K-means on `rio.png` with $K = 2, 4, 8$ and with random initial means. Show the final segmented image, and plot $d^{(t)}$ vs. t and $s^{(t)}$ vs. t for each of these Ks. Show the segmented images for iterations $t < 11$ for the cases $K = 2$ and 4.

(d) [**1 mark**] Choose one: K-means algorithm [guarantees global minimum/ guarantees only a local minimum/ does not guarantee a minimum].

Problem-2: [**7 marks**] In this problem, we will study the Expectation Maximization algorithm with the aim to segment an image with it. Given a set of data points $\{\mathbf{x}_j\}_{j=1}^{P}$, k clusters with k underlying distributions, EM algorithm assumes that each pixel value is generated from these k clusters associating with it probability of the point belonging to each cluster. In this question, we shall assume that the clusters distributions are gaussian.
Questions:

(a) [**4 marks**] We will segment the same image spectrum `spectrum.png` Fix $K = 7$. Run the EM algorithm from `http://in.mathworks.com/matlabcentral/fileexchange/34164-image-segmentation-with-em-algorithm/content/kGaussian_color_EM.m` Run the algorithm twice with random initial means. Display the outputs.

(b) [**2 marks**] Run the EM algorithm on `rio.png` with $k = 8$ twice, with random initial means. Display the outputs.

(c) [**1 mark**] Mention two differences between the K-means and EM algorithms.

Problem-3: [**12 marks**] Linear Regression: In this problem, we will look at Linear regression using Bayesian and Non Bayesian approaches. You have been given N points of (x, y) as the observed data (in data_problem3.mat file), in which y represents the world state for the observation x. Usually the world state data y will be a random variable, i.e it contains additive noise. We consider this additive noise is Gaussian with probability density $\mathcal{N}(0, \sigma^2)$. In linear regression, we are assuming that y is linearly related to x as $y = \phi^T x' + n$ where $\phi = [\phi_0 \quad \phi_1]^T$ and $x' = [1 \quad x]^T$ and n is the Gaussian noise. So the probability distribution for $y_i|x_i$ will be $\mathcal{N}(\phi^T x_i', \sigma^2)$. You have also given the ground truth value g (i.e. $g = \phi^T x'$) for computing the prediction error.

1. **[5 marks]** We perform ML estimation using training data subsets (of the whole training dataset (x, y)) of varying length k and compute prediction error on whole prediction dataset (x, g). In order to get consistent results for a k value, we need to take average of prediction errors using different training sets of length k. So Divide the training data (x, y) in to 50 different (disjoint) training sets of equal length $N/50$. Vary the number of data points k for training as $k = 10, 20, 40, 100, 150, 200, 250, 300$ and do the following

   (a) From each of these training sets (of length $N/50$), take only the first k data points to get 50 new training sets containing k data points.

   (b) Compute maximum likelihood estimate ϕ_{ML} using each of these new training data sets (separately)

   (c) Compute the prediction error on all data points (N points) of (x, g) (i.e. mean square error between $\phi_{ML}^T x'$ and g) separately for each training set and take the average. i.e.
   Let E_{kl} be the prediction error of l'th training set containing k number of points. Then compute average error E_k as

   $$E_k = \frac{1}{50} \sum_{l=1}^{50} E_{kl}$$

   (d) Plot E_k versus k

2. **[5 marks]** Now we perform Bayesian estimation for each k as above. Here we assume a prior probability distribution for ϕ as $\mathcal{N}(0, \sigma_p^2 I)$. Assume σ^2 (variance of $y_i|x_i$ distribution) to be known.

   (a) Estimate posterior distribution (separately for each of the 50 training sets containing k data points as used for the ML estimation). Let $(x^{(kl)}, y^{(kl)})$ be the

l'th training set containing k data points. Then estimate posterior distribution $Pr(\phi|x^{(kl)}, y^{(kl)})$ as $\mathcal{N}(\mu_{post}, \sigma^2_{post}I)$

(b) Compute predictive distribution for each data points x_i ($i = 1, 2 \ldots, N$) of x as

$$Pr(g_i|x_i, x^{(kl)}, y^{(kl)}) = \mathcal{N}_{pred}(\mu_i, \sigma_i^2)$$

(c) For each x_i ($i = 1, 2, \ldots, N$), take μ_i as the estimate of g_i and compute prediction error (mean square error E_{kl}) between estimated μ_i and original g_i.

(d) Average the error of 50 training set containing k points (i.e compute E_k similar to ML estimation)

(e) Plot E_k versus k for Bayesian estimation

3. [**2 marks**] Compare the performance of Bayesian estimation with ML estimation for each k.

Use 'data_problem3.mat' for this question.
Details of 'data_problem3.mat' file

- x - x-cordinate of data points (N points with N=20000) for training and prediction
- y - y-cordinate of data points (N points with N=20000) for training
- g - ground truth values (N points with N=20000) for computing prediction error
- sigma2 - σ^2
- sigma_p2 - σ_p^2

Problem-4: Non-Linear Regression

In this problem, we will look at non-linear regression using different kernels. You are required to submit all the plots.

1. [**5 marks**] Data generation : Generate synthetic data using the following Matlab code snippet. We are generating a noisy sinc function with six clearly distinguishable peaks and a total of 701 sample data points.
$x = -3 : 0.01 : 4;$
$y = sinc(x);$

$z = awgn(y, 25);$

Let the x-axis represent the data(x) and y-axis the world state(w). Plot the data. Explain why are we going for non-linear regression with respect to x and w.

2. [**5 marks**] Next, we will perform non-linear regression on the data using the following kernels:

   (a) Radial basis function(Here: Gaussian)

   (b) Sigmoid function (Here: Arc tangent)

   The transformed vector z is computed by evaluating the data x under a set of kernel functions with the following parameters:

   (a) shifting factor: α_i - centers of the RBFs, horizontal ofsets of the arc tangent functions

   (b) scaling factor: λ - determines RBFs' width, controls the speed with which the arc tangent functions change

   RBF: The choice of above mentioned parameters is critical. For RBF kernel, start with six kernel functions and a constant function to evaluate the data x. α_is for these have to be choosen at uniformly sampled intervals, such that each RBF accounts for a part of the original data space. λ has to be obtained looking at the variance around the peaks in the original data. Refer Fig. 8.6 from the textbook.

   Arc Tangent: For this case, consider seven arc tangent functions. Again uniformly position them and estimate the weights. For proper parameter selection refer Fig. 8.7 from the textbook.

   Estimate the weights by MLE of linear regression model using the nonlinearly transformed data z instead of the original data x. Using obtained weights plot predicted curve and find prediction error.

3. [**10 marks**] Now we will perform error analysis on the problem. Take the original sinc as the ground truth data.

   (a) For each of the kernels, vary λ and plot both fitting and prediction errors. Explain the plot.

(b) For, RBF, for the optimal lambda obtained in the previous step, vary the number of kernel functions in the range(2,4,6...50). Plot prediction error corresponding to these. Explain the plot. Report the optimal number of kernel functions.

Hints: Refer the following links to read more about errors. The blog is especially simple and useful.

`https://en.wikipedia.org/wiki/Errors_and_residuals`

`http://scott.fortmann-roe.com/docs/MeasuringError.html`

Problem-5: Relevence Vector Regression

In this problem we will look at the relevance vector regression. For this regression we will use the existing toolbox(**SB2**[1]) present in Problem 5 folder. We will use the following **SparseBayesDemo.m** code tailored for this problem from the toolbox.

1. Here you are provided with 1-dimensional data for regression in **data.mat** file, which has train data(trainX.mat is input and trainY.mat is the corresponding output) and test data. In this case we will use Gaussian basis centered around training data for non linear fitting. For this we need to select the basis width(λ), vary λ between [0,1](5 cases) and select the one giving minimum MSE on test data. Plot the λ vs MSE curve.

   Using this best λ do the regression and submit plots showing the given test data and the fit obtained by the model. Also plot the weights obtained by relevance vector machine as stem plot(matlab) and submit it.

   Run the demo code once, which runs on a sample data, please observe the obtained plots to get an idea of what kind of plots you need to submit.

2. RVM gives the most relevant data points for the fit, to see this we will randomly select 10 data points and use them as centers for Gaussian basis fuctions and do nonlinear regression. You can use the above obtained λ for this case. Repeat this random selection 5 times and report the average of MSE on test data. Also submit the plot showing the model fit and test data for all the 5 random selections. Please highlight the 10 data points that you have selected in these plots.

–end–

[1] http://www.miketipping.com/downloads.htm