EE5177: Machine Learning for Computer Vision

Programming Assignment - 1 : Fitting probability distributions to data

Due date: Feb 4th 2017, 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 earlier.
   (a) Problem1: Sharath, email: ee15s050@ee.iitm.ac.in
   (b) Problem2: Anil, email: ee15s055@ee.iitm.ac.in
2. Submit a single zip file in the moodle named as PA1_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**]

You are given a set X of D-dimensional data points generated using a Gaussian distribution of mean μ and covariance matrix Σ. Consider the distribution of prior on the mean to be Gaussian with $P(\mu) = \mathcal{N}_\mu(\mu_0, \Sigma_0)$. Given the covariance Σ, we want to estimate the following for the underlying distribution of the data points $P(x|X)$.

1. Find the maximum likelihood ML estimate of the mean: μ_{ML}. [**2 marks**]
2. Find the maximum aposteriori (MAP) estimate of the mean: μ_{MAP}. [**3 marks**]
3. Find the posterior distribution $P(\mu|X)$ as $\mathcal{N}(\mu_B, \Sigma_n)$ using the Bayesian method. Also, find $P(x|X)$ as $\mathcal{N}(\mu_B, \Sigma_B)$ using this posterior. [**5 marks**]

The estimates can be quantitatively analysed using a distance measure between the ground-truth and estimated values. We employ the Bhattacharya distance for this purpose. Calculate the Bhattacharya distance between the ground truth PDF $\mathcal{N}(\mu, \Sigma)$ and the estimated PDF $\mathcal{N}(\mu_*, \Sigma_*)$, where $* \in \{ML, MAP, B\}$, as

$$d(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(\mu_*, \Sigma_*)) = \frac{1}{8}(\mu - \mu_*)^T \widetilde{\Sigma}^{-1}(\mu - \mu_*) + \frac{1}{2} \ln \left(\frac{\det \widetilde{\Sigma}}{\sqrt{\det \Sigma \det \Sigma_*}} \right) \tag{1}$$

where $\widetilde{\Sigma} = \dfrac{\Sigma + \Sigma_*}{2}$.

Contents of `data.mat` for $D = 10$:
`data` - 50000 points of 10-dimensional data
`mu` - ground-truth mean
`Sigma` - ground-truth covariance
`mu01` - mean of prior-1
`Sigma01` - covariance of prior-1
`mu02` - mean of prior-2
`Sigma02` - covariance of prior-2

Steps to plot d vs. n: [**8 marks**]

1. Divide `data` into 100 disjoint sets each containing 500 points.

2. Take one of the 100 sets containing 500 points. Take the first n data points from this set and compute the Bhattacharya distance d for ML, MAP, and Bayesian. Vary $n = 2$ to 500, and calculate d for each n.

3. Repeat this d vs. n calculation for each of the 100 sets, and average them to obtain a final d for each n (for ML, MAP, and Bayesian separately).

4. Plot the final d vs. n for ML, MAP, and Bayesian methods for prior-1 (`mu01`,`Sigma01`) and prior-2 (`mu02`,`Sigma02`) using the above procedure.
   Plot-1: ML and MAP for prior-1
   Plot-2: MAP and Bayesian for prior-1
   Plot-3: ML and MAP for prior-2
   Plot-4: MAP and Bayesian for prior-2

Questions: [**2 marks**]

- Compare ML and MAP estimates from the d vs. n plots. Which one will you prefer?
- Compare MAP and Bayesian estimates from the d vs. n plots. When does the Bayesian estimate perform better than the MAP estimate?

Note: The report should contain derivations, plots, and inferences.

Problem-2 [25 marks]: You have been provided with a dataset consising of face(face_train.zip, face_test.zip) and non-face(nonface_train.zip, nonface_test.zip) images. For this problem, use train data to fit Multivariate Normal distribution for face and non-face models using Maximum Likelihood Estimation. Then evaluate the loglikelihood score(logarithm of probability) for test data from both the models built for face and non-face. Now for a given test data whichever model gives higher loglikelihood score classify it as belonging to that model.

Instructions for preparing the data :
The dataset consists of grayscale images of faces(660 train and 500 test) of size 28x28, cropped exactly to face. The non-face dataset has grayscale images of natural and indoor scenes(400 train and 395 test) of the same dimension.
Step 1 : Convert each of the images into a vector of dimension 784 x 1 and stack up the

images to form the faces data matrix of dimension 784 x 660. Do the same for the non-faces dataset, 784 x 400. Also reduce the scale of intensities from [0,255] to [0,1].
Step 2 : Going about classification with all the 784 dimensions would be computationally infeasible. So, we will use a dimensionality reduction method called Principal Component Analysis (PCA, `https://en.wikipedia.org/wiki/Principal_component_analysis`), to bring down the dimensions from 784 to a certain small number(say 100).

For this you need to do the eigen decomposition of covariance matrix of train data and then order the eigen values from highest to lowest. Now select the eigen vectors corresponding to top 100 eigen values. Project the data along these selected eigen vectors which gives you the final dimension reduced data. For test data also use the eigen vectors of correspoding train data for dimension reduction.

Make an analysis of the image classification accuracy on the following accounts:

1. Fix the pricipal components to 140 and vary the size of the train data used for training the models as follows:

Number of faces	Number of nonfaces
150	150
150	200
200	200
250	200
300	200
300	300
400	300
400	400
600	400

   In all these cases choose first n data for training the models i.e. for the 1st case choose first 150 faces from face_train data and first 150 from nonface_train data. While testing the models do it on the entire test data. Analyze the classification results by looking at the confusion matrix(`https://en.wikipedia.org/wiki/Confusion_matrix`). Also plot the classification accuracy for all these cases.

2. Now consider only the best face model(in terms of no. of face and nonface images) that you have trained. Try to visualize the loglikelihood score given by this model to

the test set of faces and nonfaces, if the model is good enough it should give high score to face images and low score to nonface images. In order to use this face model alone for classification we need to have a threshold score above which we can classify as face and below as nonface. Now we will look at a method for arriving at this threshold by plotting the Receiver Operating Characteristic(ROC) curve.

ROC curve is obtained by plotting True Positive Rate(TPR) vs False Positive Rate(FPR) by varying the threshold. For more details look at wiki reference, (ROC, `https://en.wikipedia.org/wiki/Receiver_operating_characteristic`).

(a) Plot ROC curve using the face model as a predictor on test data. Comment on its shape. Now from this ROC suggest a threshold which gives the maximum TPR under the constraint that FPR ≤ 0.2.

(b) Now plot the ROC curve using the non-face model as predictor on test data. Comment on its shape and predictive ability in comparison with face model. Also give a threshold which gives the maximum TPR under the constraint that FPR ≤ 0.2.

Look at the loglikelihood score given by the models to decide the range and steps over which you might want to vary the threshold.

Based on this experiment, which model would you like to use for the classification task?

–end–