CS5691 : Pattern Recognition and Machine Learning

Programming Assignment 1 **Deadline: 15^{st} March 2019, 23:55 hrs**

Instructions:

1. You have to turn in the well-documented code along with a detailed report of the results of the experiments.
2. Any kind of plagiarism will be dealt with severely. Acknowledge any and every resource used.
3. Plot your data and analyze before proceeding.
4. Be precise with your explanations. Avoid verbosity. Put only relevant and best/worst results plot. Report size should be **10-12 pages** (single column, 11pt). You will get a heavy penalty if you make a longer report than this.
5. You can use any language for this assignment. Using Python or MATLAB would be easier.
6. Create a folder named "TeamNumber_TeamMember1RollNo_TeamMember2RollNo' (for e.g. "1_CS17S016_CS17S011). In this folder, you should have your report and a sub-folder "codes" which should have all your codes. Upload this folder(.zip) on Moodle. Please follow the naming convention strictly.
7. Please make only one submission for the team. No emailed reports will be accepted.

Bayesian Classifier

In this assignment, you are supposed to build the Bayesian classifiers for the datasets assigned to your team. The purpose of this assignment is the analysis of classification techniques and getting used to handling data in Machine Learning. Dataset for each team can be found here. It also has the sample plots required. Divide the points randomly for training(70%), validation(15%) and testing(15%).

Algorithm 1: Bayes Classifier

1. Compute prior probabilities for each class from the dataset.
2. Estimate class conditional densities using the maximum likelihood estimation.
3. Use Bayes rule to estimate the posterior probability $q_i(X), i = 0, 1, ...M - 1$.
4. The Bayes classifier, h_B for the M-class case is:

$$h_B(X) = \alpha_i \quad \text{if} \quad \sum_{j=0}^{M-1} L(\alpha_i, C_j)\, q_j(X) \le \sum_{j=0}^{M-1} L(\alpha_k, C_j)\, q_j(X), \forall k \quad \text{(break ties arbitrarily)}$$

where, $C_0, C_1, \cdots, C_{M-1}$ are the class labels, $L(\alpha_j, C_k)$ is the loss when classifier says α_j and true class is C_k, and $q_i(X)$ is the posterior probability.

---

1. Consider the following loss function L : $\begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$ and build a Bayesian model on datasets 1 and 2 for the following cases (You can refer Section 2.6 of "Pattern classification" book by [Duda et al. 2001] for theory): (10 marks)
   (a) Model 1: Naive Bayes classifier with covariance $= \mathcal{I}$.
   (b) Model 2: Naive Bayes classifier with covariance same for all classes.
   (c) Model 3: Naive Bayes classifier with covariance different for all classes.
   (d) Model 4: Bayes classifier with covariance same for all classes.
   (e) Model 5: Bayes classifier with covariance different for all classes.

   The report should include the following for both datasets (You can refer to sample plots here and Figures 2.6 and 2.9 of "Pattern classification" book by [Duda et al. 2001]):

   1. Table of classification accuracies for all the models on training data and validation data.
   2. Confusion matrix on test set for the best model among those in parts (a)-(e).
   3. Decision boundary and decision surface of the best model. Superpose the training data on this plot.
   4. Contour curves and eigenvectors of the covariance matrix for the best model.

2. Use your best Bayesian model on dataset 2 and find classification accuracy on test set, by considering different sizes of the training set, say 100, 500, 1000, 2000, and 4000. Plot classification accuracy on test set by averaging over 20 replications and add standard

error bars. Also, find the number of training samples you need to get 85% classification accuracy on test set. (3 marks)

3. Build a Bayes classifier on datasets 3 and 4 assuming that the underlying distributions are normal. (4 marks)
   (a) Determine the training error by considering only one feature.
   (b) Repeat the same by considering any two features.
   (c) Repeat the same by considering all three features.
   (d) Discuss your results. In particular, is it possible for a finite set of data that the error might increase as the number of features increase?

4. Consider Bayesian estimation of mean on one-dimensional Gaussian dataset 5. Suppose prior of the mean is $\mathbf{P}(\mu) \sim \mathcal{N}(\mu_0, \sigma_0)$. Estimate σ by assuming that $\mu_0 = -1$. Plot your estimated densities $\mathbf{P}(x/\mathcal{D})$ for $n = 10, 100, 1000$ with different $\sigma^2/\sigma_0^2 = 0.1, 1, 10, 100$. (3 marks)

Regression and Bias Variance

Download the required data from here.

5. Generate 100 sample points in the domain (0,1). For each sample point, generate a target value using the function assigned to your team, with additive Gaussian Noise having Zero mean and variance 0.2. Divide the data set as 70:20:10 (train:test:validation). Perform polynomial regression. In particular, do the following: (*Plot the data, the target function and the regression output. Also tabulate the coefficients obtained for each model.*)
   - Choose 10 points from the training data-set and perform regression for degrees: {1,3,6,9}.
   - For each of the models above, analyze over-fitting by:
     (a) **Ridge Regression**: Try different values for the regularization parameter, ranging from very low to very high.
     (b) Varying the data-set size.
   - Show the scatter plot with target output t_n on x-axis and model output $y(x_n, w)$ on y-axis for the best performing model, for training data and test data.
   - Plot the root-mean-square(RMS) error. (*See 1.3 in Bishop's book for definition.*)

For sample plots, see Figures 1.4, 1.5, 1.6, 1.7 and Tables 1.1, 1.2 of Bishop's book

6. Let $X \sim unif(-1, 1)$ and $Y = e^{(tanh 2\pi x)} - x$. Let the noise be $N(0, \sigma^2)$, with $\sigma^2 = 0.2$. Analyze bias-variance trade-off for the three models corresponding to the polynomials of degrees 1, 5, 9 respectively. In particular do the following for each model:

   1. Sample 10 points and do Ridge regression for λ={0.001, 0.01, 0.1}.
   2. Store the empirical risk observed for this data-set.

   Repeat the steps 1,2 above for 1000 times. Plot the histogram of the empirical risk for each of the models.
   Interpret the results, and relate the observed behaviour with bias-variance trade-off.
   Refer to Figure 9.4 of Duda's book.

7. **Linear model for regression using Gaussian basis functions**.
   For the bivariate data-set assigned to your group build a linear model of regression using Gaussian basis functions. Train the model on data-set "train100.txt" to show over-fitting. Control the over-fitting using ridge regularization and by increasing the training data size. Try out different values of regularization parameter λ, and present the results as following:

   - Plot of the approximated functions obtained using training datasets.
   - Scatter plot with target output t_n x-axis and model output $y(x_n, w)$ on x-axis for the best performing model, for training data and test data.
   - A table showing the E_{RMS} on the training data and the test data.