STAD68 Advanced Machine Learning and Data Mining

Fall 2016

Homework 1

Due: 22 September 2016

Homework submission: Homework should be submitted electronically via Blackboard by 11.59pm on the due date.

Allowed resources: You may use the course notes, the books listed on the syllabus, and Wikipedia. Do not Google for the answer.

Problem 1 (Sample exam-level difficulty problem: Naive Bayes)

Consider a classification problem with training data $\{(\tilde{\mathbf{x}}_1, \tilde{y}_1), \ldots, (\tilde{\mathbf{x}}_n, \tilde{y}_n)\}$ and three classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. The sample space is $\mathbb{R}^5$, so each data point is of the form $\mathbf{x} = (x^{(1)}, \ldots, x^{(5)})$. Suppose we have reason to believe that the distribution of each class is reasonably well-approximated by a multivariate Gaussian such that each coordinate has variance 1.

1. How is the Gaussian assumption translated into a naive Bayes classifier? Write $g(\mathbf{x}|\mu, \sigma)$ to denote the Gaussian density function. Write out the full formula for the estimated class label $\hat{y}_{\text{new}} = f(\mathbf{x}_{\text{new}})$ under zero–one loss for a newly observed data point $\mathbf{x}_{\text{new}}$, in terms of the estimated parameters of the model.
   Hint: This equation should not contain the training data, only parameters estimated from the training data.
2. How do you estimate the parameters of the model? Give the estimation equations for (a) the parameters of the class-conditional distributions and (b) the class prior $P(y = k)$ for each class $\mathcal{C}_k$.
3. When would the naive Bayes assumption be exact? Give your answer in terms of assumptions on the parameters of the multivariate Gaussian class conditional distributions. When would expect the naive Bayes classifier to perform well?

Problem 2 (Sample exam-level difficulty problem: Linear Classification)

Consider a perceptron classifier in $\mathbb{R}^2$, given by a hyperplane with orthogonal vector v_{H} and offset c, and two points $\mathbf{x}_1$ and $\mathbf{x}_2$. Suppose that

$$v_{\text{H}} := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad c := \frac{1}{2\sqrt{2}} \qquad \mathbf{x}_1 := \begin{pmatrix} -3 \\ 0 \end{pmatrix} \qquad \mathbf{x}_2 := \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} .$$

1. Compute the classification result for $\mathbf{x}_1$ and $\mathbf{x}_2$.
2. If a perceptron and SVM are trained on the same linearly separable classification problem, in what ways could the two classifiers disagree on the training data? On test data? Answer the second problem with an 2D illustration of trained classifiers, training data, and test data. Make sure both classifiers, as drawn, could actually be those learned by the perceptron and SVM, respectively.

Problem 3 (Perceptron)

You will implement the Perceptron and visualize its behavior.

Homework questions:

1. Write code to create a synthetic ("made up") data set that is linearly separable. In particular, write a procedure that accepts two argument: d the number of dimensions and n the number of examples. Your code should then output the labeled data. You can choose the format, but a natural one would be a $n \times d$-matrix for the d-dimensional data $\mathbf{x}$ and a n-vector y for the label.

2. Visualize the output of your procedure in 2 dimensions and $n \in \{10, 100\}$ data points using a scatter plot with + symbols for positively labeled examples and − symbols otherwise. Show two random data sets for each setting.

3. Write code that takes as input 1) a linear classifier $\mathbf{v}_{\text{H}}$ and 2) n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (e.g., in the form of a matrix), and returns the vector of classifications.

4. Visualize the output of your procedure again on a randomly generated 2D data set and the classifier $\mathbf{v}_{\text{H}} = (0\ 0)^T$. Make classifications (+/−) that are correct be BLUE and those that are incorrect, RED.

5. Write code to take a dataset and produce a random training–test data split. In particular, write code that takes as input 1) a number k and 2) a labelled data set with at least $n > k$ observations, and outputs a random split of the dataset into two halves: k labeled training and $n - k$ labeled test data.

6. Visualize a full data set of $n = 100$ examples and a random size $n' = 10$ training data set. You can make the visualization side by side, or plot them in the sample plot using color to distinguish those points that are in the training data set.

7. Write a procedure that takes 1) an initial classifier $\mathbf{v}_{\text{H}}$ and 2) a labeled data set, and implements the perceptron algorithm with the step size rule $\alpha(k) = \frac{1}{k}$, where k is the number of the current iteration. The procedure should return the learned classifier $\mathbf{v}_{\text{H}}$ if the data are linearly separable, or should return an error message otherwise, stating the data are not linearly separable.

8. Generate a random linearly separable data set of $n = 10$ data points, run the perceptron classifier, and visualize the linear boundary. Repeat the same procedure on a data set that is not linearly separable (you can create such a data set by hand, if you wish). Demonstrate that your program recognizes that the data are not linearly separable.

9. (Bonus) Repeat the above steps again, but for affine classifiers.