Homework 4
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Instructor: Hasan Kurban

November 9, 2017

Directions

Please follow the syllabus guidelines in turning in your homework. I am providing the LaTeX of this document too. This homework is due Monday November 20, 2017 11:59p.m. **OBSERVE THE TIME**. Absolutely no homework will be accepted after that time. Bring a hardcopy to Tuesdays class on the 21th. If you do not bring a hardcopy with the statement of your own work, the homework will not be accepted. All the work should be your own. Within a week, AIs can contact students to examine code; students must meet within three days. The session will last no longer than 5 minutes. If the code does not work, the grade for the program may be reduced. Lastly, source code cannot be modified post due date.

K-Fold Cross Validation for Model Selection

ALGORITHM `k-fold cross validatiaon`
INPUT

- training data $\Delta = (\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)$
- set of parameter values Θ
- learning algorithm $\mathcal{H}$
- integer k

OUTPUT

- $\theta^* = \text{argmin}_\theta[error(\theta)]$
- $h_{\theta^*} = \mathcal{H}(\Delta; \theta^*)$

Randomly partition Δ into $\Delta_1, \ldots, \Delta_k$
*** $\Delta_1 \cup \Delta_2 \ldots \cup \Delta_k = \Delta$ and $\Delta_i \cap \Delta_j = \varnothing$ for $i \neq j \in [1, 2, \ldots, k]$
for $\theta \in \Theta$ **do**
    for $i = 1 \ldots k$ **do**
        *** Train a model for each training set
        $h_{i,\theta} = \mathcal{H}(\Delta \setminus \Delta_i; \theta)$
    end for
    *** Use the trained models over Δ_i (test data sets) to evaluate the models for each parameter
    $\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}_{\Delta_i}(h_{i,\theta})$
end for

K-Nearest Neighbors (KNN) Algorithm in Theory

ALGORITHM `K-nearest neighbors`
INPUT

- training data Δ
- test data Δ'
- distance metric d, i.e., $d : \Delta^2 \to \mathbb{R}_{\geq 0}$
- integer k: nearest neighbors number

OUTPUT

- class label of each $z \in \Delta'$

for $z = (\mathbf{x}', y') \in \Delta'$ **do**
    Compute $d(\mathbf{x}, \mathbf{x}')$, the distance between z and every example $(\mathbf{x}, y) \in \Delta$
    Select $\Delta_z \subseteq \Delta$, the set of closest k training examples to z
    Voting:

- majority voting: $y' = \text{argmax}_v \sum_{(\mathbf{x_i}, y_i) \in \Delta_z} I(v = y_i)$
- distance-weighted voting: $y' = \text{argmax}_v \sum_{(\mathbf{x_i}, y_i) \in \Delta_z} w_i \times I(v = y_i)$ where $w_i = \frac{1}{d(\mathbf{x}', \mathbf{x_i})^2}$

end for

Naive Bayes Classifier

ALGORITHM `Training of naive bayes classifier (continuous attributes)`
*** training set: $\Delta = \{(\mathbf{x_j}, y_j)\}_{j=1}^m$
for $i = 1, \ldots, k$ **do**
    *** class-specific subsets
    $\Delta_i \leftarrow \{\mathbf{x_j} | y_j = c_i, j = 1, \ldots, m\}$
    *** size of the subsets
    $m_i \leftarrow |\Delta_i|$
    *** prior probability
    $\hat{P}(c_i) \leftarrow m_i/m$
    *** mean
    $\hat{\mu}_i \leftarrow \frac{1}{m_i} \sum_{\mathbf{x_j} \in \Delta_i} \mathbf{x_j}$
    *** centered data
    $\mathcal{Z}_i \leftarrow \Delta_i - \mathbb{I}_{m_i} \hat{\mu}_i^T$
    *** variance
    $\hat{\sigma}_i \leftarrow \frac{1}{m_i} \mathcal{Z}_i^T \mathcal{Z}_i$
end for
return $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$ for all $i = 1, \ldots k$

TESTING($\mathbf{x}$ and $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$, for all $i \in [1, k]$):
$\hat{y} \leftarrow \text{argmax}_{c_i} \{f(\mathbf{x} | \hat{\mu}_i, \hat{\sigma}_i)\ \hat{P}(c_i)\}$
return $\hat{y}$

M-estimate of Conditional Probability

If the class-conditional probability for one of the attributes is zero, then overall posterior probability for the class vanishes. This problem can be addressed by using the m-estimate approach for estimating the conditional probability:

$$P(x_i|y_j) = \frac{n_c + m \times p}{n + m}$$

- x_i: training example x_i, y_j**:** class y_j
- n_c : number of training examples from class y_j that take on the value x_i
- n : total number of instances from class y_j
- m : equivalent sample size. m determines the trade-off between the prior probability p and the observed probability n_c/n
- p : user-specified parameter. p can be regarded as the prior probability of observing the attribute value x_i among records with class y_j

In this homework, you are asked to implement k-nearest neighbors (KNN), naive bayes classifier and k-fold cross validation for model selection. You will test/compare them over Ionosphere, car evaluation and credit approval data sets. Click on the links below to obtain the data sets.

- Ionosphere Data Set
- Car Evaluation Data Set
- Credit Approval Data Set

Problem 1: K-Fold Cross Validation [25 points]

Implement k- fold cross validation and select $k = 5$ to create 5 training and 5 test data sets from each data set and save these 30 files. You will use these data sets for model comparison and parameter selection.

Problem 2: K-Nearest Neighbors (KNN)[55 points]

2.1 Implement KNN algorithm with two different distance functions. You can either use existing distance functions, i.e., Euclidean or design your own.

2.2 Use the data sets obtained in problem 1 to determine the optimal k over each data set for KNN algorithm. For 5 different k values, plot the test error for each data set. Total number of figures = 3 (data set number) $\times$ 2 (distance function number) = 6. Report the best k and distance function for each data set.

2.3 Use the KNN package in R for validation.

Problem 3: Naive Bayes Classifier [55 points]

3.1 Implement Naive Bayes classifier. The Pseudo-code for naive bayes algorithm is provided above. You may need to modify it for categorical variables. To handle unseen feature values, you may need to make use of m-estimate of conditional probability method. There are also other techniques, i.e., Laplace smoothing.

3.2 Train Naive Bayes classifiers over training data sets and test each classifier against corresponding test data. Make a plot that shows the error over each test data. Report the average error rate for 5-fold cross validation for each data sets.

3.3 Use Naive Bayes package in R for validation.

Problem 4: Naive Bayes Classifier vs. K-Nearest Neighbors [30 points]

In this question, you are asked to compare Naive Bayes classifier with k-nn algorithm. First, determine the best KNN model for each data set. Then, Make a plot that reveals comparison of two algorithms using test error for each data set. (Total number of figures = 3)

Problem 5 [15 points]

From textbook, Chapter 4 exercise 10.g and 13 (only for k-nn and logistic regression)

Extra credit (optional) [40 points]

1. You are asked to evaluate the performance of two classifier models, M_1 and M_2. The test set you have chosen contains 26 binary attributes, labeled as A through Z. The table below shows the posterior probabilities obtained by applying the models to the test set. (Only the posterior probabilities for the positive class are shown). As this is a two-class problem, P(-) = 1 - P(+) and P(-|A,...,Z) = 1 - P(+|A,...,Z). Assume that we are mostly interested in detecting instances from the positive class.

Instance	True Class	$P(+\|A,\ldots,Z,M_1)$	$P(+\|A,\ldots,Z,M_2)$
1	+	0.73	0.61
2	+	0.69	0.03
3	−	0.44	0.68
4	−	0.55	0.31
5	+	0.67	0.45
6	+	0.47	0.09
7	−	0.08	0.38
8	−	0.15	0.05
9	+	0.45	0.01
10	−	0.35	0.04

   (a) Plot the ROC curve for both M_1 and M_2. Which model do you think is better. Explain your reasons?

   (b) For model M_1, suppose you choose the cutoff threshold to be $t = 0.5$. In other words, any test instances whose posterior probability is greater than t will be classified as a positive example. Compute the precision, recall, and F-measure for the model at this threshold value.

   (c) Repeat the analysis for part (c) using the same cutoff threshold on model M_2. Compare the F-measure results for both models. Which model is better? Are the results consistent with what you expect from the ROC curve?

   (d) Repeat part (c) for model M_1 using the threshold $t = 0.1$. Which threshold do you prefer, $t = 0.5$ or $t = 0.1$? Are the results consistent with what you expect from ROC curve?

2. Student/s who design/s the best either Naives Bayes classifier or KNN algorithm for the given data sets will receive 20 points.

What to Turn-in (Submission Instructions)

Put the below files in a zipped folder for your submission. The zipped folder should be named as "usename-section number", i.e., hakurban-P556

1. The *tex and *pdf of the written answers to this document.
2. Code and Data
   (a) Question 1: crossValidation.R, output of cross validation: training and test data sets
   (b) Question 2.1: knn.R, Question 2.3: knnValidation.R
   (c) Question 3.1: naiveBayes.R, Question 3.3: naiveBayes-Validation.R
3. A README file that explains how to run your code and other files in the folder

References

[1] Bater Makhabel. *Learning Data Mining with R.* Packt Publishing Ltd, 2015.

[2] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

[3] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to data mining. 1st, 2005.