IDS575: Statistical Models and Methods *Updated: 03/03/2020*

Homework #3

Instructor: Moontae Lee *Total Points: 150+10*

Policy

1. HW3 is due by 04/12/2020 11:59PM in Central Time. One submission per each group.
2. You are allowed to work individually or as a group of up to three students.
3. Having wider discussions is not prohibited. Put all the names of students beyond your group members. However individual students/groups must write their own solutions.
4. Put your write-up and results from the coding questions in a single pdf file. Compress R source codes into one zip file. Each student/group must submit only two files. (You will lose the points if the answers for coding questions are not included the pdf report)
5. If you would include some graphs, be sure to include the source codes together that were used to generate those figures. Every result must be easily reproducible.
6. Maximally leverage Piazza to benefit other students by your questions and answers. Try to be updated by checking notifications in both Piazza and the class webpage.
7. Late submissions will be penalized 20% per each late day. No assignment will be accepted more than 3 days after its due date. For HW3, it will be 04/15/2020 11:59pm.

Problem 1: New Kernels from Old Kernels [30 points]

The dual problem of SVM evaluates an inner product between every pair of two feature vectors. Since the inner product $\langle x^{(i)}, x^{(j)} \rangle$ is the only term that depends on the input features of the training data $D = \{(x^{(i)}, y^{(i)}) | 1 \leq i \leq m\}$, we can structurally incorporate higher-order interaction terms such as $(x_k^{(i)})^2$, $(x_l^{(j)})^2$, or $x_k^{(i)} x_l^{(j)}$ by introducing the feature mapping function ϕ and the kernel $K(x, z) = \langle \phi(x), \phi(z) \rangle$. Kernalization is important and useful not only for SVM but also for many other machine learning algorithms.

(a) Given a uni-variance radial basis kernel $K(x, z) = \exp\{-\frac{\|x-z\|^2}{2}\}$, prove the feature mapping of x and z are distanced at most $\sqrt{2}$. (Hint: Think about $\|\phi(x) - \phi(z)\|^2$)

While we learned how to verify the validity of a kernel, we have not much learned how to construct various kernels. For the rest of the problems, we are interested in building new kernels based on existing valid kernels. For each subproblem, prove the validity of the new kernel K if you think so. Otherwise, provide a counter-example. For your convenience, assume K_1 and K_2 are kernels over $\mathbb{R}^n \times \mathbb{R}^n$ and $\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^d$ be a feature mapping function.

(b) $K(x,z) := K_1(x,z) + K_2(x,z)$

(c) $K(x,z) := K_1(x,z) - K_2(x,z)$

(d) $K(x,z) := aK_1(x,z) \quad (a > 0)$

(e) $K(x,z) := bK_1(x,z) \quad (b < 0)$

(f) $K(x,z) := K_1(x,z)K_2(x,z)$ (Hint: Could be longer to justify than other subproblems)

(g) $K(x,z) := f(x)f(z) \quad (f : \mathbb{R}^n \longrightarrow \mathbb{R}$, a real valued function)

(h) $K(x,z) := K_3(\phi(x), \phi(z)) \quad (K_3$: another valid kernel over $\mathbb{R}^d \times \mathbb{R}^d)$

(i) $K(x,z) := p(K_1(x,z)) \quad (p(x)$: a polynomial function with positive coefficients)

Problem 2: Naïve-Bayes Text Categorization [50 points]

In this problem, you are to work on a text classification problem by using the same data provided for Problem 5 in Homework 2. Your goal is to classify each text article into one of the 4 categories by using a multi-class Naïve-Bayes model rather than using multi-class SVMs. If you use a Bernoulli Naïve-Bayes model, the class label for a document that consists of n different word types is decided by the following formula:

$$p(y = k|x_1, ..., x_n) = \frac{p(x_1, ..., x_n|y = k)p(y = k)}{p(x_1, ..., x_n)} \tag{1}$$

$$\propto p(x_1, ..., x_n|y = k) = p(y = k) \prod_{j=1}^{n} p(x_j|y = k). \tag{2}$$

It means that if a document consists of j-th word x_j by l_j times (among its n unique words out of $|V|$ vocabulary), we can classify its class label y as k by finding the

$$argmax_{k \in \{1,2,...,4\}} \; p(y = k|x_1, ..., x_n).$$

That is the most probable class label given your word observations is based only on the appearance of each word ignoring how many times they appear. Recall that the denominator of Equation (1) does not affect on the class prediction because it does not consists of $y = k$ term. So, it is removed in Equation (2). The last equality in Equation (2) is a result of Naïve-Bayes assumption rather than holding in general. On the other hand, if you use a multinomial Naïve-Bayes model, the class label is decided by

$$p(y = k|x_1, ..., x_n) = \frac{p(x_1, ..., x_n|y = k)p(y = k)}{p(x_1, ..., x_n)} \tag{3}$$

$$\propto p(x_1, ..., x_n|y = k) = p(y = k) \prod_{j=1}^{n} p(x_j|y = k)^{l_j}. \tag{4}$$

It means that your prediction takes into account not only of the existence of each word, but also of its actual frequency. Due to Naïve-Bayes assumption, your conditional probability $p(x_j|y = k)$ for j-th word is now multiplied l_j times as every occurrence is conditionally independent given the class label $y = k$. Note that the fact that a word x_j is appeared l_j times is recorded in our dataset as a sequence of $j : l_j$ terms. If you use Bernoulli Naïve-Bayes, you are treating all appearing l_j equally as 1.

Try to load the training and test data. While each article can be seen as a vector in $\mathbb{R}^{|V|}$, you should only store each of the existing words and their counts like the SparseMatrix in multi-class SVMs.

(a) The category of each text article must depend on the meaning of its content. Explain why Naïve-Bayes assumption is not too unrealistic for text categorization problem. (Hint: In Equation (2) or (4), $\prod$ part is invariant even if you multipliy $p(X_j|Y = k)$ in various different orders)

(b) [Coding] Train the model based on the training data by MLE. For each class k among 4 different classes, you should learn the parameter $\phi_{j|y=k}$, which is the conditional probability $p(x_j|y = k)$. You should also learn the parameter $\phi_{y=k}$, which is the prior probability of each class $p(y = k)$. Report the confusion matrix and training accuracy by predicting the class labels of the training set by your trained Bernoulli Naïve-Bayes model.

(c) [Coding] Learn the model parameters again by performing Laplace smoothing (in Lecture Notes #09a). Report the new confusion matrix and training accuracy when predicting on training data. Report another confusion matrix and test accuracy when predicting on test data. (Note: You cannot report test statistics without Laplace smoothing because there are unseen words in the test data as we experienced at Problem 5 in Homework 2)

(d) [Coding] Report part (c) with multinomial Naïve-Bayes model. Report correspondingly to part (c).

(f) (+5 pts) Compare and contrast the results from part (c) and (d). Justify why one works better than the other in our dataset. Explain, more in general, the weakness of Naïve-Bayes models by comparing Bernoulli event model and multinomial event model. (Hint: Think about what happen if the same word occurs multiple times in an article)

Problem 3: Hidden Markov Models [30 points]

Suppose we have Hidden Markov Model (HMM) with 3 hidden states (S_1, S_2, S_3) and the corresponding observations (O_1, O_2, O_3). Each state can take k different values, and there are total m observations across all states.

(a) Count the number of parameters to define the initial distribution, the transition distribution, and the emission distribution.

(b) Does the number of parameters depend on the number of states? Briefly justify your answer.

(c) Enumerate all conditional independence holding in this HMM.

The specific initial, transition, and emission distributions are given for the following problems. Answer the following questions by using these probability values. **These are not coding questions!**

State	$p(S_1)$
A	0.99
B	0.01

S_t	S_{t+1}	$p(S_{t+1}\|S_t)$
A	A	0.99
A	B	0.01
B	A	0.01
B	B	0.99

S_t	O_t	$p(O_t\|S_t)$
A	0	0.8
A	1	0.2
B	0	0.1
B	1	0.9

(d) Compute the probability to observe the sequence $(O_1, O_2, O_3) = (0, 1, 0)$ when using the forward algorithm. Try to include both the intermediate equations of values and the final value after computation. (Feel free to use a calculator for evaluating the final value)

(e) Compute the probability to observe the sequence $(O_1, O_2, O_3) = (0, 1, 0)$ when using the backward algorithm. Include the intermediate evaluations similarly to the part (d).

(f) Are the two results from the forward and backward algorithms same? What is the most likely sequence of values for these three states? Note that each state takes a value of either A or B. Include all intermediate evaluations that lead you to your conclusion. (Hint: You can reuse your work in part (d) and (e)).

(g) If you use Viterbi algorithm, what is the most likely sequence of values for these three states? Include all intermediate evaluations that lead you to your conclusion.

(h) Try to find the most likely sequence of values for considering each state separately (i.e., ignoring between state dependency). Compare whether the resulting sequence of values is equivalent to what you have observed in part (f) or (g). Justify why and whether this is hold in general.

Problem 4: K-means Clustering [15 points]

Given the following 6 data points, simulate K-means clustering manually with $K = 2$. Each example consists of two features and initially assigned to the cluster k.

(a) Plot the data points, and compute the centroid of each cluster.

(b) Assign each data point to its closest centroid, reporting the new cluster label k.

i	$x_1^{(i)}$	$x_2^{(i)}$	k
1	1	4	1
2	1	3	2
3	0	4	1
4	5	1	1
5	6	2	1
6	4	0	2

(c) Repeat (a) and (b) until convergence. Once the centroids and the cluster labels stop changing, report the cluster label k for each data point.

(d) (+ 5pts) Check whether the final clustering matches your initial visual clustering in part (a). Is the clustering result always same regardeless of initial cluster assignment? (Hint: You can try and report by coding on R)

Problem 5: Principal Component Analysis $+K$-means [25 points]

For this problem, first generate a synthetic data set with 20 data points for each of the three class. Each data point must consist of 50 features, so that we have total 60×50 data matrix. Feel free to use $rnorm()$ or $runif()$. Make sure that you should add a mean shift to the data points in order to make them three distinctive classes. Then you are going to play with two unsupervised algorithms: PCA and K-means clustering.

(a) Run PCA on these 60 data points. Plot the first two principle axes. Try to use different colors to contrast the data points that belong to different classes. If three classes are distinctive enough, continue to the next part (b). Otherwise keep synthesizing a new dataset until you reach at some degree of separations across three classes in terms of two principal component axes.

(b) Run K-means clustering with $K = 3$. Compare the obtained clusters to the true class labels. (Hint: Feel free to use $table()$ function, but note that K-means clustering only separates without assigning particular label values. Be careful in comparison)

(c) Run K-means clustering with $K = 2$ and 4. Explain your results in contrast to part (b).

(d) Now run K-means clustering with $K = 3$ only on the two principal axes that you discovered in part (a). In other words, run K-means clustering on 60×2 matrix. Explain your result comparatively to the previous results.

(e) Given the original 60×50 data that you worked on part (b), run K-means clustering with $K = 3$ after standardizing each feature. Explain your result in contrast to part (b) (Hint: Feel free to use $scale()$ function to make each feature have standard deviation 1. You should not remove the existing mean shift in each feature.)