CIS 700-006: Advanced Machine Learning for NLP **Fall 2017**

Problem Set 0

Handed Out: August 31, 2017 *Due: Not Mandatory*

- The goal of this assignment is to make sure you have the basic terminology, ideas and techniques needed for CIS700. The questions below do not test this exhaustively but provide a sample that will allow you to evaluate whether you are prepared to take this advanced machine learning class.
- Please write down a brief and clear solution. We will not grade it, but rather give you a solution to study and compare with you own work.
- I highly recommend that you spend time on it. My expectation is that you will spend no more than 3 hours on this and solve correctly at least 5 of the 7 problems. Please record the time you spend.
- It is fine if you need to use additional material to refresh you memory or your understanding of the material. My goal is to make sure that are familiar with a lot of the material, and have sufficient understanding of the area so that you can navigate your way and solve these problems.
- There are plenty of resources you can consult in case you need help. In particular, you can use the notes/videos from my Machine Learning class: http://L2R.cs.illinois.edu/ danr/Teaching/CS446-17/schedule.html
- Please make sure you complete this assignment before the next lecture. We will provide a solution then.

1. **[Perceptron - 14 points]**

   In this question, we will be asking you about Perceptrons and their variants.

   Let $D = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$, where the j-th example $\mathbf{x}^{(j)}$ is associated with the label $y^{(j)} \in \{-1, +1\}$. Each example $\mathbf{x}^{(j)}$ is a bit-vector of length n, i.e. $\mathbf{x}^{(j)} \in \{0,1\}^n$, with the interpretation that the i-th bit of the vector $(x_i^{(j)})$ is 1 if the element described by $\mathbf{x}^{(j)}$ has the i-th attribute on.

   (a) Let us first consider a Perceptron where the positive example $\mathbf{x}$ satisfies $\mathbf{w} \cdot \mathbf{x} \geq \theta$, where $\mathbf{w} \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ and $\mathbf{x}$ is some example $\mathbf{x}^{(j)}$ from D.

      1. Suggest an equivalent representation of this Perceptron in the form of $\underline{\mathbf{w}' \cdot \mathbf{x}' \geq 0}$ given an example $\mathbf{x}^{(j)}$, where $\mathbf{x}' \in \{0,1\}^{n'}$ for some suitable integer n'. (That is, define n', w', x'.)

2. In the following table, we describe a specific data set S. Using an initialization of $\mathbf{w}' = \mathbf{0}$, i.e. the zero vector, and a learning rate of $R = 1$, complete the columns under **(a)** of the table using the Perceptron learning algorithm.

	S			(a)		(b)	
j	$\mathbf{x}_1^{(j)}$	$\mathbf{x}_2^{(j)}$	$y^{(j)}$	Mistake? Y/N	Updated $\mathbf{w}'$	Mistake? Y/N	Updated $\mathbf{w}'$
Initialization				———	$\mathbf{0}$	———	$\mathbf{0}$
1	1	1	+1				
2	1	0	−1				
3	0	1	+1				

(b) We now consider the *Perceptron with margin* algorithm, for margin $\gamma > 0$. We will represent that decision of the algorithm as before, $\mathbf{w}' \cdot \mathbf{x}' \geq 0$, but will need to use a different update rule for the weights.

1. Let the margin be $\gamma > 0$ and the learning rate $R > 0$. For a given $(\mathbf{x}^{(j)}, y^{(j)})$, write down the update rule for the Perceptron with margin.

2. Using the same data set S used above, use an initialization of $\mathbf{w}' = \mathbf{0}$ (the zero vector), a learning rate of $R = 1$ and margin $\gamma = 1.5$, and complete the columns under **(b)** of the table using the *Perceptron with margin* learning algorithm.

(c) Now we would like to learn a linear separator of the form $\mathbf{w}' \cdot \mathbf{x}' \geq 0$, the canonical representation for any separating hyperplane. This time however, we would like to learn the weights $\mathbf{w}'$ by *minimizing* the error made by the linear separator over S.

We define the error made by $\mathbf{w}'$ over S using the *hinge loss* function. Complete the definition of this loss:

$$L(y^{(j)}, \mathbf{x}^{(j)}, \mathbf{w}') = \_\_\_\_\_$$

(use $\mathbf{x}^{(j)'}$, the representation of example $\mathbf{x}^{(j)}$ in the form of $\mathbf{x}'$ in the canonical representation.)

Thus the goal of learning is to minimize the following error:
(Complete the error definition)

$$\text{Error}(\mathbf{w}', D) = \sum_{j=1}^{m} L(y^{(j)}, \mathbf{x}^{(j)}, \mathbf{w}') = \_\_\_\_\_$$

One way to do this is to make use of Stochastic Gradient Descent.

1. Write the pseudocode for Stochastic Gradient Descent using this hinge loss function with a fixed learning rate of $R > 0$.
   Remember to:
   i. Specify how to initialize $\mathbf{w}'$.
   ii. Specify how to transform $\mathbf{x}^{(j)}$ to $\mathbf{x}^{(j)'}$.
   iii. Specify how to update $\mathbf{w}'$.
2. Suggest a condition on the problem definition that will make the Stochastic Gradient Descent algorithm identical to the Perceptron with Margin algorithm.

2. [**Kernels - 16 points**]

In this question we will develop a learning algorithm that will take as input a URL and classify it according to whether it is relevant to the topic "Machine Learning" or not. The classifier will only depend on the string of the URL, and not on the web page itself.

In the following we will develop a kernel that will be used to learn how to map a URL string to "relevant" and "irrelevant".

We are given a collection of m URLs $u_1, u_2, \ldots, u_m$. Each URL consists of characters taken from a vocabulary V of n characters $c_1, ..., c_n$. We can assume that V includes *all* ASCII characters.

The *basic* feature vector for each URL u is $F(u)$. $F(u)$ is a binary vector, $F(u) \in \{0,1\}^n$, where the jth component in $F(u)$ indicates whether the character c_j appears in URL u ($F(u)[j] = 1$) or not ($F(u)[j] = 0$). For example, for the URL u = *www.cnn.com*, the set of active features in $F(u)$ is $A = \{$***w***, ***c***, ***n***, **.**, ***o***, ***m***$\}$, i.e. components in $F(u)$ that correspond to the indices of the characters of A will be 1, all others will be 0.

Each u is also labeled as relevant ($l = 1$) or irrelevant ($l = 0$).

(a) The presence of some *set* of characters can be indicative of "machine learning", e.g. *ml, sv*. So in addition to the basic features in $F(u)$, we want to include features that indicate if a pair of *different* characters c_i and c_j appear anywhere in the URL u. Let us call this new feature space $\phi(u)$.

1. What is the total number of features in the expanded feature space of $\phi(u)$ (as a function of the vocabulary size n)? Explain briefly.

2. Assume a URL u = *www.a.sg*
   Write down the active features in the expanded feature vector $\phi(u)$.

3. For a URL u of length s, where all the s characters are different, what is the number of active features in $\phi(u)$ (as a function of s)? Explain briefly.

4. We want to use the Kernel Perceptron algorithm to learn the function above. Design a kernel $K(u_1, u_2)$ that allows us to compute the value of the dot product $\phi(u_1)\phi(u_2)$ in time linear in n by directly computing it in the $F(u)$ space without expanding to the $\phi(u)$ space (assume that length of u_1 and u_2 is at most n). Give the formula for $K(u_1, u_2)$ and explain why it is true.

(b) Now we are going to define a new feature space $\psi(u)$. $\psi(u)$ will include all features from $F(u)$ and also include features that indicate whether a pair of characters appear consecutively in URL u. For example, for the URL $u =$ *www.cnn.com*, the set of active features in $\psi(u)$ will be $A = \{$***w***, ***c***, ***n***, ***.***, ***o***, ***m***, ***ww***, ***w.***, ***.c***, ***cn***, ***nn***, ***n.*** , ***co***, ***om***$\}$.

1. What is the size of the expanded feature space $\psi(u)$ (as a function of the vocabulary size n)?

2. For a URL u of length s where all the s characters are different, what is the number of active features of $\psi(u)$ (as a function of s)?

3. Write a kernel $K(u_1, u_2)$ that can compute the dot product $\psi(u_1)\psi(u_2)$ in time linear in n (assume that length of u_1 and u_2 is at most n).

3. [**Multiclass - 15 points**]

You are a newspaper editor and you notice that your journalism interns repeatedly make mistakes in using appropriate prepositions in the essays they write. You collect 100 example sentences where interns made mistakes and label them with the correct preposition from set $Y = \{\texttt{with}, \texttt{on}, \texttt{for}, \texttt{at}\}$. There are 25 example sentences for each.

You want to build a classifier to predict which preposition should appear in a given sentence. Let us denote each example sentence as a n-dimensional boolean feature vector $\langle x_1, \ldots, x_n \rangle, x_i \in \{0, 1\}$ that has a label $y \in Y$.

(a) You decide to use the **One vs. All** scheme, but are not sure whether to use the **Perceptron** algorithm or **naïve Bayes**. Fill in the table below, using the options from Table 1.

<table>
<tr><th>Perceptron</th><th>Naïve Bayes</th></tr>
<tr><td colspan="2"><u>(i) Training protocol (training the model)</u></td></tr>
<tr><td>What classifiers do you need to learn?
Fill the table below using values from Table 3. You don't have to use all the rows below if you don't need them.
<table>
<tr><th rowspan="2">Classifier #</th><th colspan="2">Positive examples</th><th colspan="2">Negative examples</th></tr>
<tr><th>label(s) of examples</th><th># of examples</th><th>label(s) of examples</th><th># of examples</th></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td><td></td></tr>
</table></td><td>What parameters do you need to estimate from the data?
Choose appropriate options from Table 3 to answer the question.</td></tr>
</table>

(1) {with}	(5) {with, on}	(11) {with, on, for}	(16) $\Pr(x_i)$	$i \in \{1, \ldots, n\}$
(2) {on}	(6) {with, for}	(12) {with, on, at}	(17) $\Pr(x_i \mid y)$	
(3) {for}	(7) {with, at}	(13) {with, for, at}	(18) $\Pr(y \mid x_i)$	$y \in Y$
(4) {at}	(8) {on, for}	(14) {on, for, at}	(19) $\Pr(y)$	
	(9) {on, at}		(20) 25	(22) 75
	(10) {for, at}	(15) {with, on, for, at}	(21) 50	(23) 100

Table 1: Options to choose from. You may choose an option multiple times. In the table, the notation {} means all examples belonging to the labels given inside the curly brackets.

Perceptron	Naïve Bayes
(ii) Test protocol (labeling unseen examples)	
How will you label an unseen example based on the classifiers you learned above?	How will you label an unseen example based on the parameters you defined above?

(b) After you trained the model, you notice that the interns make another common mistake, while using the preposition `in`. You collect 25 more examples where the correct preposition is `in`, and want to learn a classifier that distinguishes *five* classes instead of four. Fill in the table below.

Perceptron	Naïve Bayes
(iii) New training protocol	
Do you have to change the Perceptron classifiers you learned in part (i)? Justify your answer.	What parameters do you need in order to learn the new classifier?
	What parameters from part (i) can you reuse as-is?
How will the test protocol (part (ii)) change?	What are the new values of the parameters from part (i) that you have to re-estimate?
	Which parameters will you have to estimate afresh (i.e. the parameters that you did not have in part (i))?

4. **[Support Vector Machines - 15 points]**

We are given the following set of training examples $D = \{(x_1^{(i)}, x_2^{(i)}, y^{(i)})\}, i = 1, \ldots, m$, where $x_j^{(i)}$ are integer-valued features, and $y^{(i)}$ are binary labels.

x_1	x_2	y
-2	-4	+
-2	0	+
0	2	+
2	2	-
2	-2	-
0	-4	-

Our objective is to learn a hyperplane $w_1 x_1 + w_2 x_2 + b = 0$ using the hard-SVM objective. Write down the objective for this two dimensional case:

minimize_____

subject to_____

You will need to create a plot to finish the following questions. (If you find that creating the plot is too time consuming it's sufficient to answers the questions below by providing the coordinates of the appropriate and the equation of the separator.)

(a) Finding the hard-SVM hyperplane:

1. Plot the training examples, and indicate the support vectors.

2. Draw the hyperplane produced by the hard-SVM..

3. Find the values of $w_1, w_2, b \in \mathbb{R}$ that optimize the hard-SVM objective.

(b) Experimental evaluation:

1. Provide the classification rule used to classify an example with features x_1, x_2, using the hyperplane produced by hard-SVM.

2. What will the error of your classifier be on the training examples D (expressed as the fraction of training examples misclassified)?

3. Draw the hyperplane that will be produced by hard-SVM when you use all training examples except $a = (0, 2, +)$. Using this hyperplane, will you classify a correctly?

4. Draw the hyperplane that will be produced by hard-SVM when you use all training examples except $b = (2, 2, -)$. Using this hyperplane, will you classify b correctly?

5. What will be the average error if you use 6-fold cross validation on the training set D?

(c) Soft-SVM formulation:

1. Write the soft-SVM objective below. Circle either min or max.

2. For what range of positive C values, will the hyperplane produced by this soft-SVM objective be most similar to the hyperplane produced by hard-SVM. Circle one of the following.

very small moderate very large

5. [**Naive Bayes - 14 points**]

You would like to study the effects of *irrigation*, *fertilization* and *pesticide* use with respect to the **yield** of a farm. Suppose you are provided with a collection $D = \{D_1, \ldots, D_m\}$ of m data points corresponding to m different farms. Each farm has three binary attributes *IsIrrigated* (X_1), *IsFertilized* (X_2) and *UsesPesticide* (X_3), and each has either a high yield ($V = 1$) or a low yield ($V = 0$). The label is **Yield**. A natural model for this is the **multi-variate Bernoulli model**.

Below is a table representing a *specific collection* S of data points for 8 farms to illustrate how a collection might look like.

#	*IsIrrigated* (X_1)	*IsFertilized* (X_2)	*UsesPesticide* (X_3)	**Yield** (V)
1	No (0)	Yes (1)	No (0)	High (1)
2	Yes (1)	Yes (1)	No (0)	High (1)
3	No (0)	Yes (1)	No (0)	Low (0)
4	No (0)	Yes (1)	No (0)	High (1)
5	No (0)	No (0)	Yes (1)	Low (0)
6	Yes (1)	No (0)	Yes (1)	Low (0)
7	No (0)	No (0)	No (0)	Low (0)
8	No (0)	Yes (1)	No (0)	High (1)

(a) Circle *all* the parameters from the table below that you will need to estimate in order to completely define the model. You may assume that $i \in \{1, 2, 3\}$ for all entries in the table.

(1) $\alpha_i = \Pr(X_i = 1)$	(7) $\beta = \Pr(V = 1)$
(2) $\gamma_i = \Pr(X_i = 0)$	(8) $\varphi = \Pr(V = 0)$
(3) $p_i = \Pr(X_i = 1 \mid V = 1)$	(9) $q_i = \Pr(V = 1 \mid X_i = 1)$
(4) $r_i = \Pr(X_i = 0 \mid V = 1)$	(10) $s_i = \Pr(V = 0 \mid X_i = 1)$
(5) $t_i = \Pr(X_i = 1 \mid V = 0)$	(11) $u_i = \Pr(V = 1 \mid X_i = 0)$
(6) $w_i = \Pr(X_i = 0 \mid V = 0)$	(12) $y_i = \Pr(V = 0 \mid X_i = 0)$

(b) How many **independent** parameters do you have to estimate to learn this model?

(c) Write explicitly the naïve Bayes classifier for this model as a function of the model parameters selected in (a):

(d) Write the expression for L, the log likelihood of the entire data set D, using the parameters that you have identified in (a).

(e) We would like to train a Naïve Bayes classifier on S to help us predict the yield on a new farm S_9.

1. What is the decision rule for the Naïve Bayes classifier trained on S?

2. Predict the yield for the following farm using the decision rule written earlier.

#	*IsIrrigated* (X_1)	*IsFertilized* (X_2)	*UsesPesticide* (X_3)	**Yield** (V)
9	Yes (1)	Yes (1)	Yes (1)	?

6. [**Probability - 10 points**]

You are given the following sample S of data points in order to learn a model. This question will use this data.

Example	A	B	C
1	1	1	0
2	0	1	1
3	1	0	0
4	0	0	0
5	1	1	0
6	0	0	0
7	1	0	1
8	0	1	1
9	1	1	0
10	0	0	0
11	1	1	1
12	0	0	0

(a) What would be your estimate for the probability of the following data points, given the sample S, if you were not given any information on a model? (That is, you would estimate the probability directly from the data.)

1. $P(A = 1, B = 1, C = 0)$

2. $P(A = 0, B = 1, C = 1)$

3. $P(A = 0, B = 0, C = 1)$

(b) Consider the following graphical model M over three variables A, B, and C.

$$A \rightarrow B \rightarrow C$$

1. What are the parameters you need to estimate in order to completely define the model M? Choose these parameters from Table 2.

(1) $P[A = 1]$	(5) $P[B = 1]$	(9) $P[C = 1]$
(2) $P[A = 1\|B = b]$ $b \in \{0, 1\}$	(6) $P[B = 1\|C = c]$ $c \in \{0, 1\}$	(10) $P[C = 1\|A = a]$ $a \in \{0, 1\}$
(3) $P[A = 1\|C = c]$ $c \in \{0, 1\}$	(7) $P[B = 1\|A = a]$ $a \in \{0, 1\}$	(11) $P[C = 1\|B = b]$ $b \in \{0, 1\}$
(4) $P[A = 1\|B, C = b, c]$ $b, c \in \{0, 1\}$	(8) $P[B = 1\|A, C = a, c]$ $a, c \in \{0, 1\}$	(12) $P[C = 1\|A, B = a, b]$ $a, b \in \{0, 1\}$

Table 2: Options to choose from to explain model M.

2. Use the data to estimate the parameters you have chosen in (b).1.

(c) Use the parameters chosen in (b).1 to write down expressions for the probabilities of the same data points according to model M and compute these probabilities using the estimated parameters.

1. $P_M(A = 1, B = 1, C = 0)$

2. $P_M(A = 0, B = 1, C = 1)$

3. $P_M(A = 0, B = 0, C = 1)$

(d) Use the parameters chosen in (b).1 to write down the expressions for the following probabilities for model M and compute these probabilities.

1. $P_M(B = 1)$

2. $P_M(A = 1|B = 0)$

3. $P_M(A = 0|B = 0, C = 0)$

7. [**Expectation Maximization - 16 points**]

Assume that a set of 3-dimensional points (x, y, z) is generated according to the following probabilistic generative model over Boolean variables $X, Y, Z \in \{0, 1\}$:

$$Y \leftarrow X \rightarrow Z$$

(a) What parameters from Table 3 will you need to estimate in order to completely define the model?

(1) `P(X=1)`	(2) `P(Y=1)`	(3) `P(Z=1)`	
(4) `P(X|Y=b)`	(5) `P(X|Z=b)`	(6) `P(Y|X=b)`	(7) `P(Y|Z=b)`
(8) `P(Z|X=b)`	(9) `P(Z|Y=b)`	(10) `P(X|Y=b,Z=c)`	(11) 3

Table 3: Options to choose from. $b, c \in \{0, 1\}$.

(b) You are given a sample of m data points sampled independently at random. However, when the observations are given to you, the value of X is always omitted. Hence, you get to see $\{(y^1, z^1), \ldots, (y^m, z^m)\}$. In order to estimate the parameters you identified in part (a), in the course of this question you will derive update rules for them via the EM algorithm for the given model.

Express $\Pr(y^j, z^j)$ for an observed sample (y^j, z^j) in terms of the unknown parameters.

(c) Let $p_i^j = Pr(X = i \mid y^j, z^j)$ be the probability that hidden variable X has the value $i \in \{0, 1\}$ for an observation $(y^j, z^j), j \in \{1, \ldots, m\}$. Express p_i^j in terms of the unknown parameters.

(d) Let (x^j, y^j, z^j) represent the completed j^{th} example, $j \in \{1, \ldots, m\}$. Derive an expression for the expected log likelihood (LL) of the completed data set $\{(x^j, y^j, z^j)\}_{j=1}^m$, given the parameters in (a).

(e) Maximize LL, and determine update rules for *any two* unknown parameters of your choice (from those you identified in part (a)).