CSCI 544 – Applied Natural Language Processing, Spring 2020

Written Homework 1: Naive Bayes, Linear Classifiers, Perceptron

Out: January 22, 2020

Total: 18 pages.

General instructions

1. This is not a graded assignment. Do not turn it in.
2. The assignment is meant as preparation for the in-class exams. You should aim to answer all the questions on your own, without help.
3. Space is provided as it would be on the exams. Answers should be concise and fit in the space provided; in the exams it will not be possible to add space, and long and rambling answers will be penalized.
4. After solving the problems (or giving them your best try), you are encouraged to discuss and compare solutions with your classmates.
5. You are welcome to discuss the problems with us. We encourage open discussion on Piazza, so that the entire class can benefit from the discussion.
6. Answers to select problems will be distributed at a later time.

Problem 1. You are building a classifier for the sentiment of Russian adjectives. The following 100 adjectives have been sampled from the class of positive adjectives, to use as training data. The adjectives have been analyzed into a stem and suffix.

Adjective	Stem + suffix	Count
красивый	красив + ый	10
красивая	красив + ая	18
красивую	красив + ую	12
приятный	приятн + ый	10
приятная	приятн + ая	32
приятную	приятн + ую	18

a. Based on the training data, give estimates for the probabilities of the individual stems and suffixes below.

P(красив- | positive) =   P(приятн- | positive) =

P(-ый | positive) =   P(-ая | positive) =   P(-ую | positive) =

b. Suppose that the stem and suffix are conditionally independent, given the class (that is, a naive Bayes model). If the probability estimates you just calculated exactly describe the class of positive adjectives, how many instances of each word would you expect to find in a sample of 100 words drawn from the class of positive adjectives?

красивый

красивая

красивую

приятный

приятная

приятную

c. Is it possible to construct any sort of model that better fits the observed sample? If so, how? If not, why not?

d. Roughly speaking (without calculating numbers), does our observed sample provide strong evidence against using a naive Bayes model for describing the class of positive adjectives? Why or why not?

Problem 2. In this problem you will use probabilities to segment Arabic words into prefixes, stems and suffixes. Since we are able to give little data about stems, we concentrate only on prefixes and suffixes. The following segmented words are used as training data (∅ denotes a null prefix or suffix).

Arabic script	Prefix + Stem + Suffix	Meaning
لولده	l + wld + h	to his child
وعدك	∅ + wʕd + k	your promise
وكتبه	w + ktb + h	and his books
فكتبي	f + ktb + y	and my books
فعمله	f + ʕml + h	and his work
لعملك	l + ʕml + k	to your work
وشغل	w + šɣl + ∅	and work
باذنه	b + Aðn + h	with his permission
صحتك	∅ + SHt + k	your health
فابني	f + Abn + y	and my son

a. Give the maximum likelihood estimates for the probability of each prefix (don't forget the null prefix):

b. Give the maximum likelihood estimates for the probability of each suffix (don't forget the null suffix):

c. For segmenting words, we make the simplifying assumption that any sequence of characters is possible and equally likely as a stem; however, we do impose a constraint that a stem is at minimum three characters. Given this constraint, find the most likely segmentation for each word (use the transliteration, not the Arabic characters):

Arabic	Transliteration	Prefix + Stem + Suffix	Likelihood of prefix+stem+suffix
فعلي	fʕly	+ +	
وضحك	wDHk	+ +	

d. Does the segmenter always give the most common prefix that is consistent with the beginning of the word? Why or why not?

e. Does the segmenter always give the most common suffix that is consistent with the ending of the word? Why or why not?

Problem 3. Named entity recognition (NER) is the problem of identifying the names of persons, organizations, locations etc. In this problem you will construct a naive Bayes classifier to identify named entities in Czech. The table below is a snapshot of the data set, where phrases are labeled as to whether or not they represent a named entity. Each phrase is followed by the number of times it appears in the data.

Named entities	Not named entities
Nové Město (3)	Nové Auto (1)
Nové Dillí (5)	Kostel (9)
Kostel Panny Marie (2)	Červený (7)
Pan Červený (1)	Staré Auto (3)
Marie (4)	Nové (12)
	Červený Muž (3)

a. Identify the priors for each class:

Named entity: ______ Not named entity: ______

b. You will be constructing two types of features: *first word*, and *any word*. The *first word* feature of a phrase is the first word of the phrase; the *any word* feature of a phrase will have multiple occurrences – one for each word, including the first (so a three-word phrase, for example, will have three *any word* features).

Start by tabulating the number of instances of each feature, for each class.

	First word		Any word	
	Named Entity	Not Named Entity	Named Entity	Not Named Entity
Červený				
Kostel				
Marie				
Nové				
Pan				
Staré				
Auto				
Dillí				
Město				
Muž				
Panny				

c. Apply Laplace (add-one) smoothing, and calculate the probabilities of each feature, conditional upon class.

	First word		Any word	
	Named Entity	Not Named Entity	Named Entity	Not Named Entity
Červený				
Kostel				
Marie				
Nové				
Pan				
Staré				
Auto				
Dillí				
Město				
Muž				
Panny				

d. Use your classifier to predict for each of the following phrases whether or not they are a named entity: for each phrase, calculate the probability that it belongs to each class, and then select the most probable class. (Some of the phrases below are not proper Czech; don't worry about it for this exercise.)

	P(Named Entity)	P(Not Named Entity)	Chosen label
Červený Kostel			
Červený Město			
Dillí			
Kostel Panny Dillí			
Pan Auto			
Panny Marie			
Nové Kostel			
Nové Marie			
Nové Město			
Staré Dillí			

e. Why do we construct the feature as “any word” rather than “word other than first”? (Hint: how would we classify *Dillí* with such features?)

f. The first word of each phrase contributes two features for classification (*first word* and *any word*), so in effect it is counted twice. Is this justified? What would happen to *Pan Auto* (“Mr. Auto”), *Nové Marie* (“New Mary”), and *Staré Dillí* (“Old Delhi”) if the first word only contributed one feature?

Problem 4. In this problem you will use probabilities to identify German noun phrases as subjects or objects (a very simple form of *semantic role labeling*). The following sentences are used as training data, with noun phrases annotated as subject or object.

German sentence	English translation
[der Mann]$_{subj}$ sieht [die Frau]$_{obj}$	"The man sees the woman"
[das Kind]$_{subj}$ sieht [den Hund]$_{obj}$	"The child sees the dog"
[den Mann]$_{obj}$ sieht [die Katze]$_{subj}$	"The cat sees the man"
[der Hund]$_{subj}$ sieht [den Mann]$_{obj}$	"The dog sees the man"
[die Frau]$_{subj}$ sieht [die Katze]$_{obj}$	"The woman sees the cat"
[das Kind]$_{subj}$ sieht [die Ziege]$_{obj}$	"The child sees the goat"
[den Hund]$_{obj}$ sieht [der Mann]$_{subj}$	"The man sees the dog"
[der Mann]$_{subj}$ sieht [das Kind]$_{obj}$	"The man sees the child"
[die Katze]$_{obj}$ sieht [das Kind]$_{subj}$	"The child sees the cat"
[das Kind]$_{subj}$ sieht [die Frau]$_{obj}$	"The child sees the woman"

We will use a Naive Bayes classifier to classify the nouns based on two features: the article (*der, die, das, den*) and the position in the sentence (first or second).

a. Identify the priors for each class:

Subject: ______ Object: ______

b. Give the maximum likelihood estimates for the probability of each feature value, given the class:

	Article				Position	
	der	*die*	*das*	*den*	First	Second
Subject						
Object						

c. Use the classifier to predict the semantic role for each noun phrase in the following sentence: for each noun phrase, calculate the probability that it's generated as subject or object, and then select the most probable class.

[den Mann] sieht [die Frau] "The woman sees the man"

	P(subject)	P(object)	Chosen label
den Mann			
die Frau			

The correct semantic role labels for the above sentence are:

[den Mann]$_{obj}$ sieht [die Frau]$_{subj}$ "The woman sees the man"

The Naive Bayes classifier should have gotten one of the labels wrong (if not, check your math).

d. Why did the classifier make a wrong prediction? What information is missing from the current model?

e. Would smoothing on one or both of the features help? What assumptions about the German language would make smoothing desirable or undesirable?

Problem 5. This exercise traces through the first few steps of a perceptron training algorithm. The task is to classify a sentence into one of two classes, which are called $+1$ and -1. We will use just two features; unlike the example in class, these features are not binary, but integer-valued. The features are:

pron The number of personal pronouns in the sentence.

noun The number of proper and common nouns in the sentence.

In the data below, each instance of **pron** is marked in **red boldface**, and each instance of ***noun*** is marked in ***green bold italics***. A hyphenated term such as *Cochin-China* or *great-aunt* is considered a single term. The data (classes and sentences) are taken from Argamon et al.: Gender, genre, and writing style in formal written texts, *Text* **23**(3): 321–346, 2003.

a. Count the features in each sentence and update the perceptron weights, using Algorithm 5 of Hal Daumé III, A Course in Machine Learning (v. 0.99 draft), Chapter 4: The Perceptron.

$+1$ ***Clara*** never failed to be astonished by the extraordinary ***felicity*** of **her** own ***name***.

Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

-1 By 1925 present-day ***Vietnam*** was divided into three ***parts*** under French colonial ***rule***.

Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

$+1$ **She** found it hard to trust **herself** to the mercy of ***fate***, which had managed over the ***years*** to convert **her** greatest ***shame*** into one of **her** greatest ***assets***, and even after ***years*** of comparative ***security*** **she** was still prepared for, still half expecting the old ***gibes*** to be revived.

Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

-1 The southern ***region*** embracing ***Saigon*** and the ***Mekong delta*** was the ***colony*** of ***Cochin-China***; the central ***area*** with its imperial ***capital*** at ***Hue*** was the ***protectorate*** of ***Annam***; and the northern ***region***, ***Tongking***, was also a separate ***protectorate*** with its ***capital*** at ***Hanoi***.

Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

$+1$ But whenever **she** was introduced, nothing greeted the amazing, all-revealing ***Clara*** but ***cries*** of "How delightful, how charming, how unusual, how fortunate," and **she** could foresee a ***time*** when ***friends*** would name **their** ***babies*** after **her** and refer back to **her** with ***pride*** as the ***original*** from which ***inspiration*** had first been drawn.

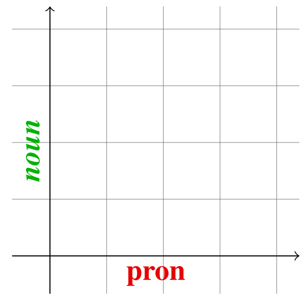
Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

-1 The Annamese ***emperor***, ***Khai Dinh***, in ***theory*** ruled the two northern ***regions*** from ***Hue*** with the ***benefit*** of French ***protection***, while ***Cochin-China*** was governed directly from ***Paris*** but in ***effect*** all three ***territories*** were ruled as ***colonies***.

Feature counts: **pron** ____ ***noun*** ____ Weights: **pron** ____ ***noun*** ____ bias ____

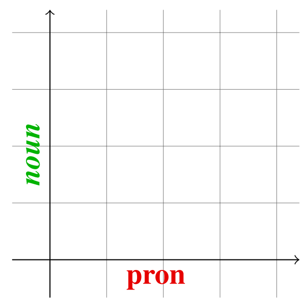
b. What is the decision boundary found by the perceptron? Give a formula, and draw the boundary on the graph with a vector pointing in the direction of the positive class (similar to Figures 4.6 and 4.9 in the reading).

Formula:



c. Suppose instead of the vanilla perceptron algorithm, we used an *averaged* perceptron (section 4.6 in the chapter). What would be the decision boundary? Give a formula and draw the boundary on the graph.

Formula:



d. How would each of the perceptrons (vanilla and averaged) classify each of the following texts?

Finally **her** ***confidence*** grew to such an ***extent*** that **she** was able to explain that **she** had been christened not in the ***vanguard*** but in the extreme ***rearguard*** of ***fashion***, after a Wesleyan ***great-aunt***, and that **her** ***mother*** had formed the ***notion*** not as an unusual and charming ***conceit*** but as a preconceived ***penance*** for **her** ***daughter***, whose only ***offences*** at that tender ***age*** were **her** ***existence*** and **her** ***sex***.

Vanilla: Averaged:

Some backward ***tribes*** inhabited the remoter ***mountains*** and ***jungles*** but the main ***population*** was of the same ***race***; today **they** are known as ***Vietnamese*** but then the outside ***world*** knew **them** as ***Annamites*** or ***Annamese***.

Vanilla: Averaged:

Problem 6. Trace the first steps of training a perceptron to classify a tweet into one of two classes, which are called $+1$ and -1. The perceptron uses just two integer-valued features:

- sent.: The number of sentences in the tweet consisting of a single word.
- CAPS: The number of words in ALL CAPS.

The perceptron training algorithm is given below: w_d is the weight of feature d, x_d is the count of feature d in a particular item, and *class* is $+1$ or -1. (Modeled on the algorithm in the reading: Hal Daumé III, A Course in Machine Learning (v. 0.99 draft), Chapter 4: The Perceptron)

```
w_d ← 0 for all features d                              # Initialize weights
b ← 0                                                   # Initialize bias
for all iterations:
    for all items:
        activation ← ∑_d w_d x_d + b                    #Compute activation
        if class · activation ≤ 0 :
            w_d ← w_d + class · x_d for all features d  # update weights
            b ← b + class                               # update bias
return w_1, w_2, ..., b
```

a. Count the features in each sentence and update the perceptron weights.

$+1$ Epic crowds in Pennsylvania tonight, but FAKE MEDIA won't report it. SAD.

Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

-1 High volatility in NASDAQ before closing with moderate gains.

Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

$+1$ Senate must choose tonight to protect America. Every vote counts!

Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

-1 Immediate release. Fed to raise interest for third month in a row.

Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

$+1$ It's wonderful to see the effect of our TAX CUTS. Phenomenal.

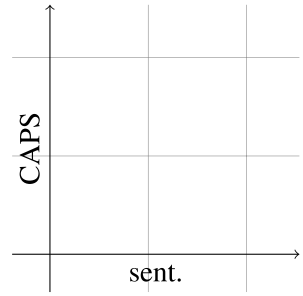
Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

-1 NYSE opening to a hesitant start.

Feature counts: sent. ____ CAPS ____ Weights: sent. ____ CAPS ____ bias ____

b. What is the decision boundary found by the perceptron? Give a formula, and draw the boundary on the graph with a vector pointing in the direction of the positive class.

Formula:



c. How would the perceptron classify each of the following tweets?

Please join me with your thoughts and prayers for both aviators, their families and our incredible @USNavy.

Class:

Today the House took major steps toward securing our schools by passing the STOP School Violence Act. We must put the safety of America's children FIRST.

Class:

We cannot keep a blind eye to the rampant unfair trade practices against our Country!

Class:

Unemployment filings are at their lowest level in over 48 years. Great news for workers and JOBS, JOBS, JOBS!

Class:

Problem 7. We have seen that in a naive Bayes model with features $f_1, f_2 \ldots$, for a specific text with corresponding feature counts $n_1, n_2 \ldots$, the log probability that the text belongs in a particular class is given by the model as follows:

$$\log P(class|text) \approx \log P(class) + n_1 \log P(f_1|class) + n_2 \log P(f_2|class) + \cdots$$

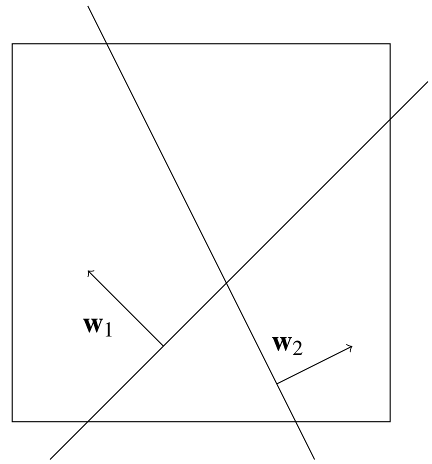
That is, the log probability of class membership is proportional to the distance above a plane corresponding to the class. The normal to the plane is a weight vector $\mathbf{w} = w_1, w_2 \ldots$ where for all features f_i, $w_i = \log P(f_i|class)$. (We can consider the log prior probability $\log P(class)$ as an extra feature w_0 where for all texts, $n_0 = 1$.)

For the following parts, assume we have two classes C_1 and C_2, with associated weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.

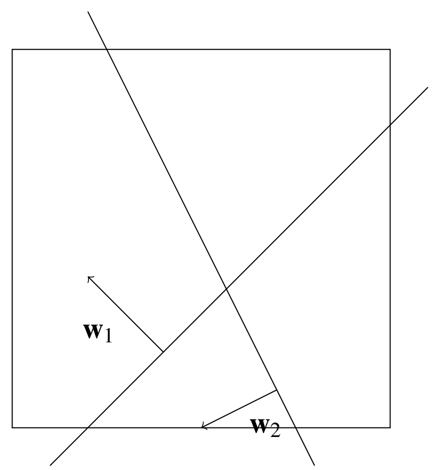
a. Given a text represented by a feature count vector $\mathbf{n} = n_1, n_2 \ldots$, when will the model classify the text as belonging to class C_1? When will the model classify the text as belonging to class C_2? Give the answers in terms of $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{n}$.

b. Given your answer above, how can we represent the *decision boundary* between the classes C_1 and C_2? In which direction from the boundary are texts classified as C_1, and in which direction as C_2? Give the answers in terms of $\mathbf{w}_1$ and $\mathbf{w}_2$.

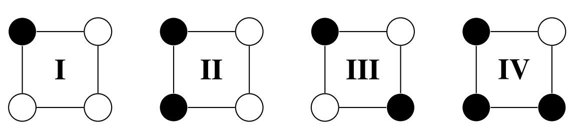
c. The following box represents a 2-dimensional feature space, with the planes and weight vectors associated with C_1 and C_2. Use a diagram and an explanatory sentence to show how these planes determine a decision boundary, and indicate the decision regions (that is, which part of the feature space will be classified as C_1 and which as C_2).



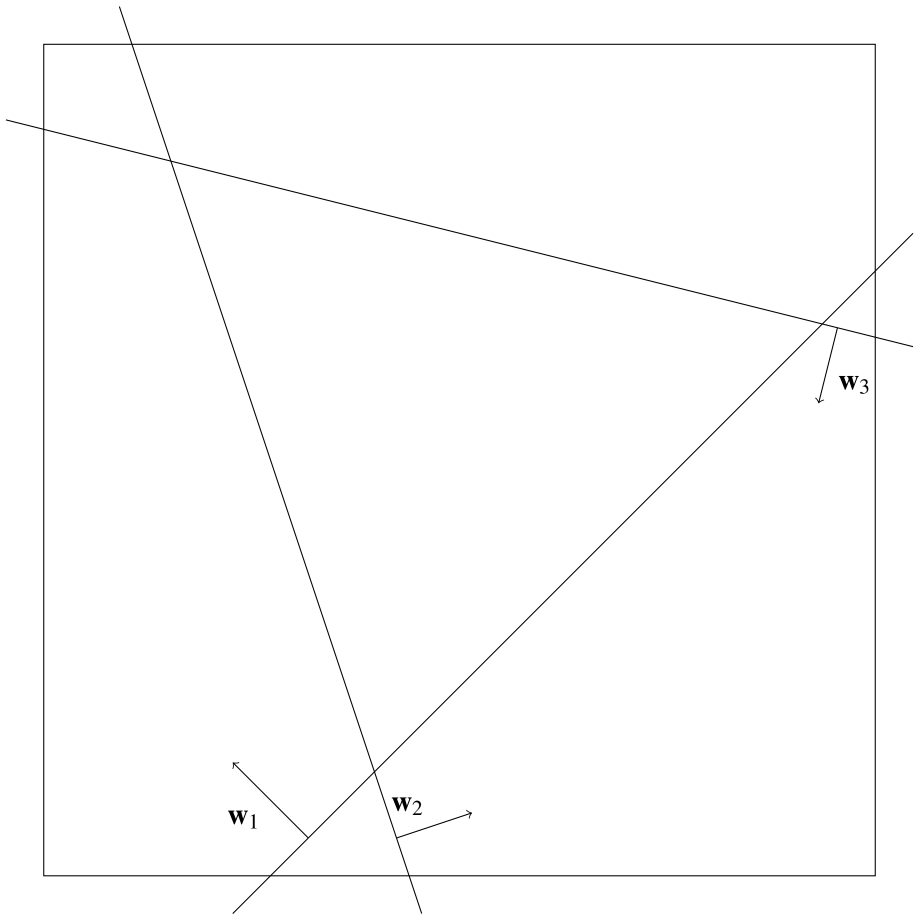
d. Do the same for the following case (diagram and explanatory sentence). What is the difference?



e. The following diagrams represent possible ways to split the four possible observations of two binary features between two classes. Which of the cases below are consistent with conditional independence of the feature values, given the class?



Problem 8. The following box represents a 2-dimensional feature space (in log space), with the planes and weight vectors associated with a three-class naive Bayes classifier. Classes C_1, C_2 and C_3 are associated with weight vectors $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$, respectively. Here, every *pair* of classes determines a decision boundary.



a. On the diagram, draw and label the decision boundary for each pair of classes. That is, draw the decision boundary between C_1 and C_2, and label which class is on each side of the boundary. Do the same for the boundary between C_1 and C_3, and for the boundary between C_2 and C_3.

b. On the diagram, identify and label the decision regions for each class. That is, which parts of the space will be classified as C_1, which as C_2, and which as C_3.