# CS 188 Spring 2010 — Introduction to Artificial Intelligence — Written 7 Naive Bayes and Perceptrons

**Due:** Thursday 4/22 in 283 Soda Drop Box by 11:59pm (no slip days)

**Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually

| | |
|---|---|
| First name | |
| Last name | |
| SID | |
| Login | |
| Collaborators | |

**For staff use only:**

| | |
|---|---|
| Q1. Naive Bayes | /8 |
| Q2. Perceptrons | /15 |
| Total | /23 |

# Q1. [8 pts] Naive Bayes

The Berkeley Police Department has asked you to assist with the interrogation of several suspects accused of cheating at cs188-Blackjack. Each suspect makes a single statement during interrogation, and from this statement you are to determine whether they are Guilty (G) or Innocent (I). To assist you, the police department provides you with records of past cheating cases: statements made by suspects, labeled with whether the suspect ended up being found Guilty (G) or Innocent (I).

| Training Statements | (G/I) |
|---|---|
| I am definitely innocent, officer | (I) |
| Officer, I swear I am not lying | (I) |
| I am not lying, I swear | (I) |
| I am innocent, officer, I swear | (G) |
| Officer, I am definitely not lying | (G) |

**(a)** [2 pts] You plan to apply a Naive Bayes classifier to this problem. Given the class label, this model treats each word in the sentence as an independent feature. The parameters of this model take the form of conditional probabilities $P(G), P(Word = \text{“}am\text{”}|G)$. Using the training data, find the maximum likelihood estimate of the parameters (they will be the class-conditional relative frequencies of each word). Ignore punctuation and capitalization.

| Prior | Prob |
|---|---|
| G | |
| I | |

| Word | $P(\text{Word}\|G)$ | $P(\text{Word}\|I)$ |
|---|---|---|
| I | | |
| am | | |
| definitely | | |
| innocent | | |
| officer | | |
| swear | | |
| not | | |
| lying | | |

**(b)** [2 pts] Using the probabilities found above, compute the following probabilities:

$P(G, \text{“Officer, I am not lying”})$

$P(I, \text{“Officer, I am not lying”})$

What is the most likely classification for the above sentence?

**(c)** [2 pts] Another suspect has been caught; this time, he gives the statement "I am honest, officer". Using the same Naive Bayes model, compute the probability $P(G, \text{"I am honest, officer"})$.

**(d)** [2 pts] Instead of maximum likelihood, use Laplace (add-one) smoothing to find new values for the model parameters (using the same training data as in (a)) and use these new parameters to compute the probability $P(G|\text{"I am honest, officer"})$.

Assume for the purposes of smoothing that every word you will see is contained in your training set and test sentence (but do not use the test sentence when computing parameter estimates). Make sure to smooth both the prior $P(G), P(I)$ and conditional distributions.

# Q2. [15 pts] Perceptrons

Instead of using Naive Bayes, you decide to try applying Perceptron to the interrogation data. You generate features from the training data as follows:

- $\phi_1(x) = n$ where "not" appears n times.
- $\phi_2(x) = m$ where "swear" appears m times.
- $\phi_3(x) = 1$, a bias term

We use the labels +1 for G and -1 for I. Given a weight vector $w = (w_1, w_2, w_3)$, our classifier returns +1 if $w_1\phi_1(x) + w_2\phi_2(x) + w_3 >= 0$ and -1 otherwise.

Our training set from part 1 yields the following features and labels:

| Training Statements | $\phi_1$ | $\phi_2$ | $\phi_3$ | Label |
|---|---|---|---|---|
| I am definitely innocent, officer | 0 | 0 | 1 | +1 |
| Officer, I swear I am not lying | 1 | 1 | 1 | +1 |
| I am not lying, I swear | 1 | 1 | 1 | +1 |
| I am innocent, officer, I swear | 0 | 1 | 1 | -1 |
| Officer, I am definitely not lying | 1 | 0 | 1 | -1 |

**(a)** [2 pts] Compute the first two updates of the Perceptron algorithm and fill in the following table, using the given initial Perceptron weights $w = (w_1, w_2, w_3)$ and data points $(\phi_1, \phi_2, \phi_3, \text{Label})$.

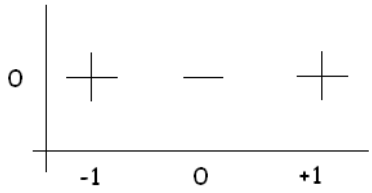
| $w$ | $w_1$ | $w_2$ | $w_3$ |
|---|---|---|---|
| Initial | 1 | 2 | -0.5 |
| Observing $(0, 0, 1, +1)$ | | | |
| Observing $(1, 0, 1, -1)$ | | | |

**(b)** [2 pts] What convergence guarantees can you give for the Perceptron algorithm applied to this data set?

**(c)** [2 pts]

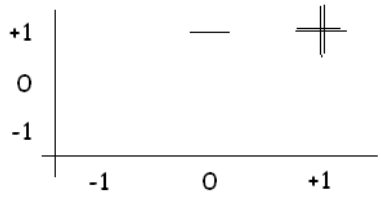
Linear classifiers are often insufficient to represent a dataset using a given set of features. However, it is often possible to find new features using nonlinear functions of our existing features which do allow linear classifiers to separate the data. Nonlinear features result in more expressive linear classifiers.

For example, consider the following data set, where +'s represent positive examples and −'s represent negative examples.




No linear classifier can separate the positive examples $(-1, 0)$ and $(1, 0)$ from the negative example $(0, 0)$.

Rather than using a single feature, if we perform a nonlinear mapping $\phi(x_1, x_2) = (x_1^2, 1)$, the positive examples are both mapped to $(1, 1)$ and the negative example is mapped to $(0, 1)$, and we see the data can be separated by a linear classifier. One example is the line $w = [1, -0.5]$, i.e. the classifier $w^\top \phi(x) = x^2 - 0.5 >= 0$.




For what values of the weight vector $w = (w_1, w_2)$ does the classifier $w^\top \phi(x) >= 0$ separate the given data?

**(d)** [3 pts] Which of the following feature sets allows a linear classifier $w = (w_1, w_2, w_3)$ to separate the original interrogation data set? Justify your answer briefly.

**(i)** [1 pt] $\phi' = (\phi_1 + \phi_2, \phi_1 - \phi_2, 1)$

**(ii)** [1 pt] $\phi' = (\phi_1 \phi_2, \phi_2^2, 1)$

**(iii)** [1 pt] $\phi' = ((\phi_1 \text{ xor } \phi_2), \phi_2, 1)$ where $a$ xor $b$ is 1 if either $a = 1$ or $b = 1$ but not both.

**(e)** [2 pts] Given the features $\phi(x) = [x^2, x, 1]$, how many data points are we guaranteed to be able to separate with zero error using a linear classifier $w^\top \phi(x) = w_1 x^2 + w_2 x + w_3$? Assume that a data point $x$ cannot have conflicting labels. Justify your answer briefly.

**(f)** [2 pts] In general, if we use features $\phi(x) = [x^{N-1}, x^{N-2}, ..., 1]$, i.e. an $N-1$th order polynomial, how many points can we separate with zero error using a linear classifier $w = [w_1, ..., w_N]$? Justify your answer briefly.

**(g)** [2 pts] Assume we have $N$ labeled training data points, which we would like to use for classification i.e. to predict the labels of unseen test data points. What are the disadvantages of using an $N$th order polynomial to fit this data?