CS 188
Summer 2019

Section Handout 12 Solutions

Naive Bayes

$$prediction(x_1, ..., x_n) = \arg\max_y P(Y = y) \prod_{i=1}^{n} P(X_i = x_i | Y = y)$$

Parameter Estimation

Given sample your **maximum likelihood** estimate for an outcome x that can take on $|X|$ different values from a sample of size N is

$$P_{MLE}(x) = \frac{count(x)}{N}.$$

With **Laplace smoothing,** the Laplace estimate with strength k is

$$P_{LAP,k}(x) = \frac{count(x) + k}{N + k|X|}.$$

A similar result holds for computing Laplace estimates for conditionals (which is useful for computing Laplace estimates for outcomes across different classes):

$$P_{LAP,k}(x|y) = \frac{count(x, y) + k}{count(y) + k|X|}.$$

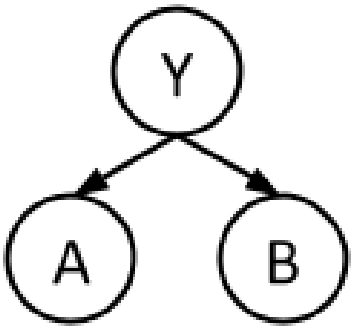
There are two particularly notable cases for Laplace smoothing. The first is when $k = 0$, then $P_{LAP,0}(x) = P_{MLE}(x)$. The second is the case where $k = \infty$. Observing a very large, infinite number of each outcome makes the results of your actual sample inconsequential and so your Laplace estimates imply that each outcome is equally likely. Indeed:

$$P_{LAP,\infty}(x) = \frac{1}{|X|}$$

1 Naive Bayes

In this question, we will train a Naive Bayes classifier to predict class labels Y as a function of input features A and B. Y, A, and B are all binary variables, with domains 0 and 1. We are given 10 training points from which we will estimate our distribution.

A	1	1	1	1	0	1	0	1	1	1
B	1	0	0	1	1	1	1	0	1	1
Y	1	1	0	0	0	1	1	0	0	0



(a) What are the maximum likelihood estimates for the tables $P(Y)$, $P(A|Y)$, and $P(B|Y)$?

Y	$P(Y)$
0	3/5
1	2/5

A	Y	$P(A\|Y)$
0	0	1/6
1	0	5/6
0	1	1/4
1	1	3/4

B	Y	$P(B\|Y)$
0	0	1/3
1	0	2/3
0	1	1/4
1	1	3/4

(b) Consider a new data point $(A = 1, B = 1)$. What label would this classifier assign to this sample?

$$
\begin{aligned}
P(Y=0, A=1, B=1) &= P(Y=0)P(A=1|Y=0)P(B=1|Y=0) && (1)\\
&= (3/5)(5/6)(2/3) && (2)\\
&= 1/3 && (3)\\
P(Y=1, A=1, B=1) &= P(Y=1)P(A=1|Y=1)P(B=1|Y=1) && (4)\\
&= (2/5)(3/4)(3/4) && (5)\\
&= 9/40 && (6)\\
& && (7)
\end{aligned}
$$

Our classifier will predict label 0.

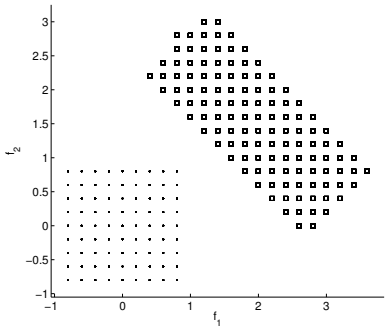
(c) Let's use Laplace Smoothing to smooth out our distribution. Compute the new distribution for $P(A|Y)$ given Laplace Smoothing with $k = 2$.

A	Y	$P(A\|Y)$
0	0	3/10
1	0	7/10
0	1	3/8
1	1	5/8

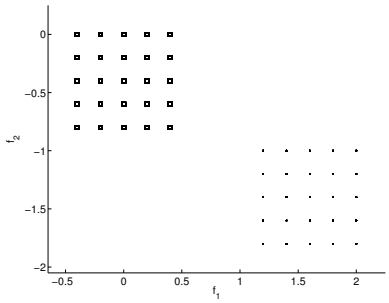
Q2. Naïve Bayes Modeling Assumptions

You are given points from 2 classes, shown as rectangles and dots. For each of the following sets of points, mark if they satisfy all the Naïve Bayes modelling assumptions, or they do not satisfy all the Naïve Bayes modelling assumptions. Note that in (c), 4 rectangles overlap with 4 dots.

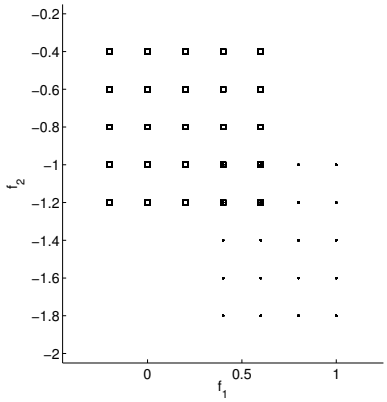
The conditional independence assumptions made by the Naïve Bayes model are that features are conditionally independent when given the class. Features being independent once the class label is known means that for a fixed class the distribution for f_1 cannot depend on f_2, and the other way around. Concretely, for discrete-valued features as shown below, this means each class needs to have a distribution that corresponds to an axis-aligned rectangle. No other assumption is made by the Naïve Bayes model. Note that linear separability is not an assumption of the Naïve Bayes model—what is true is that for a Naïve Bayes model with all binary variables the decision boundary between the two classes is a hyperplane (i.e., it's a linear classifier). That, however, wasn't relevant to the question as the question examined which probability distribution a Naïve Bayes model can represent, not which decision boundaries.



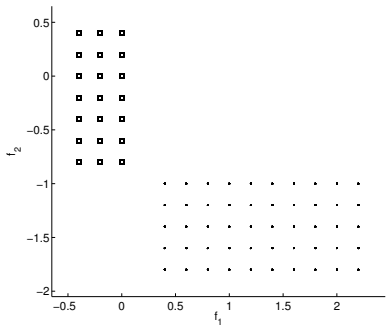
(a) Satisfies Does not Satisfy



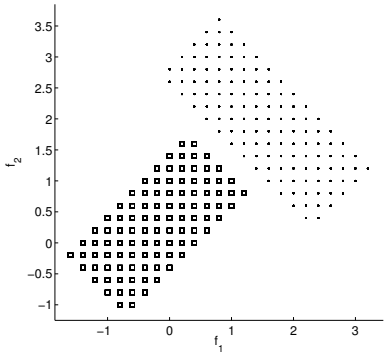
(b) Satisfies Does not Satisfy



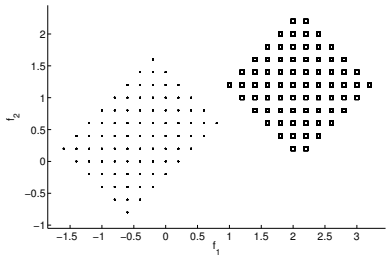
(c) Satisfies Does not Satisfy



(d) Satisfies Does not Satisfy



(e) Satisfies Does not Satisfy



(f) Satisfies Does not Satisfy

A note about feature independence: The Naïve Bayes model assumes features are conditionally independent given the class. Why does this result in axis-aligned rectangles for discrete feature distributions? Intuitively, this is because fixing one value is uninformative about the other: within a class, the values of one feature are constant across the other. For instance, the dark square class in (b) has $f_1 \in [-0.5, 0.5]$ and $f_2 \in [-1, 0]$ and fixing one has no impact on the domain of the other. However, when the features of a class are not axis-aligned then fixing one limits the domain of the other, inducing dependence. In (e), fixing $f_2 = 1.5$ restricts f_1 to the two points at the top, whereas fixing $f_2 = 0$ gives a larger domain.

3 Maximum Likelihood

A Geometric distribution is a probability distribution of the number X of Bernoulli trials needed to get one success. It depends on a parameter p, which is the probability of success for each individual Bernoulli trial. Think of it as the number of times you must flip a coin before flipping heads. The probability is given as follows:

$$P(X = k) = p(1-p)^{k-1} \tag{8}$$

p is the parameter we wish to estimate.

We observe the following samples from a Geometric distribution: $x_1 = 5$, $x_2 = 8$, $x_3 = 3$, $x_4 = 5$, $x_5 = 7$. What is the maximum likelihood estimate for p?

$$\begin{aligned} L(p) &= P(X = x_1)P(X = x_2)P(X = x_3)P(X = x_4)P(X = x_5) && (9)\\ &= P(X = 5)P(X = 8)P(X = 3)P(X = 5)P(X = 7) && (10)\\ &= p^5(1-p)^{23} && (11)\\ \log(L(p)) &= 5\log(p) + 23\log(1-p) && (12)\\ & && (13) \end{aligned}$$

We must maximize the log-likelihood of p, so we will take the derivative, and set it to 0.

$$\begin{aligned} 0 &= \frac{5}{p} - \frac{23}{1-p} && (14)\\ p &= 5/28 && (15) \end{aligned}$$

Q4. Generalization

(a) Suppose you train a classifier and test it on a held-out validation set. It gets 80% classification accuracy on the training set and 20% classification accuracy on the validation set.

From what problem is your model most likely suffering?

○ Underfitting ● Overfitting

Fill in the bubble next to any measure of the following which could reasonably be expected to improve your classifier's performance on the validation set.

● Add extra features ● Remove some features

Briefly justify: Either answer was accepted with justification. Add extra features – adding some really good features could better capture the structure in the data. Remove some features – the model may be using the noise in the abundant feature set to overfit to the training data rather than learning any meaningful underlying structure.

● Collect more training data ○ Throw out some training data

More data should yield a more representative sample of the true distribution of the data. Less data is more susceptible to overfitting.

Assuming features are outcome counts (k is the Laplace smoothing parameter controlling the number of extra times you "pretend" to have seen an outcome in the training data):

● Increase k ○ Decrease k (assuming $k > 0$ currently)

Increasing k reduces the impact of any one training instance to make the classifier less sensitive to overfitting of rare (= low count) patterns.

Assuming your classifier is a Bayes' net:

○ Add edges ● Remove edges

Removing edges reduces the class of distributions the Bayes' net can represent. Adding edges introduces more parameters so that the model could further overfit.

(b) Suppose you train a classifier and test it on a held-out validation set. It gets 30% classification accuracy on the training set and 30% classification accuracy on the validation set.

From what problem is your model most likely suffering?

● Underfitting ○ Overfitting

Fill in the bubble next to any measure of the following which could reasonably be expected to improve your classifier's performance on the validation set.

● Add extra features ○ Remove some features

Briefly justify: Under the current feature representation, we are unable to accurately model the training data for the purpose of the classification task we're interested in. The classifier may be able to deduce more information about the connections between data points and their classes from additional features, allowing it to better model the data for the classification task. For example, a linear perceptron could not accurately model two classes separated by a circle in a 2-dimensional feature space, but by using quadratic features in a kernel perceptron, we can find a perfect separating hyperplane.

● Collect more training data ○ Throw out some training data

More training data can only be a good thing. Marking neither of the bubbles was accepted, too, as given that train and hold-out validation already achieve the same performance, likely the underlying problem is not a lack of training data.

(c) Your boss provides you with an image dataset in which some of the images contain your company's logo, and others contain competitors' logos. You are tasked to code up a classifier to distinguish your company's logos from competitors' logos. You complete the assignment quickly and even send your boss your code for training the classifier, but your boss is furious. Your boss says that when running your code with images and a random label for each of the images as input, the classifier achieved perfect accuracy on the training set. And this happens for all of the many random labelings that were generated.

Do you agree that this is a problem? Justify your answer.

Yes, this is a problem. The classifier is overfitting the training set. The fact that it had perfect accuracy with random labels suggests that it does not learn any real underlying structure in the data; it most likely essentially memorized each of the training cases.