Section 9: Machine Learning

Saturday, February 2, 2019 1:42 PM

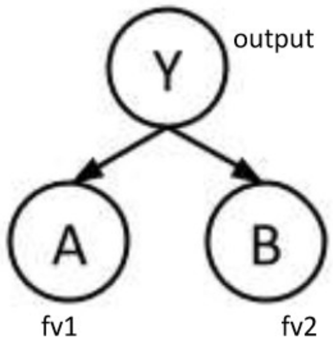
section9

CS188 Fall 2018 Section 9: Machine Learning

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 very powerful (often too powerful, b
in acceptable error rates) assumptions
represented with a BayesNet where th
feature variables. That is,
$P(X_i|Y,X_j) = P(X_i|Y), \forall i,j$ $where\ i \neq j.$
alternatively
$P(X_i,X_j|Y) = P(X_i|Y)P(X_j|Y) \forall i,j$ $where\ i$ =

MLE is our learning algorithm for our model space of a probability space that generates this data.

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

The model space is restricted to probability spaces where the Naïve Bayes assumption is true.

Y	$P(Y)$
0	0.6
1	0.4

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

So, considering our model (probabilty
Some math nerds have reduced our op
specific probability space, goal is max l

It's also fairly intuitive tho am i rite?

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

We have learned a BayesNet, so, we will use it for the query of exact inference of P(Y=0 | A=1, B=1).
Exact inference-o-meter using: Join, Marginalize, BN Assumption, LTP

P(Y|a, b) = P(a,b | Y) P(Y) / P(a, b)	Bayes Rule
=αP(a,b | Y) P(Y)	LTP, associativity of a "constant" value w.r.t. the probability table
=αP(a | Y)P(b|Y) P(Y)	BayesNet CI assumptions regarding feature vectors and label

P(y=0|a = 1, b = 1) = alpha * 0.6 * 5/6*
P(y=1|a = 1, b = 1) = alpha * 0.4 * 3/4*

y = 0 has a higher likelihood, according

3. 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

out usually resulting
s that the probability space can be
he label variable is the parent of all

$\neq j$.

space), with constriants represented by the above BayesNet
ptimization problem (assign values to parameters that specify a
likelihood estimate) into a counting problem. Fuck. Yea. Bitches.

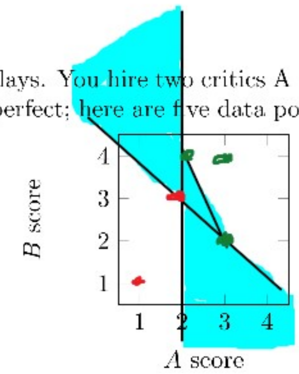
* 2/3 = 0.3333
* 3/4 = 0.225

g to this learned BayesNet

2 Perceptron

You want to predict if movies will be profitable based on their screenplays. You hire two critics A and B to read a script you have and rate it on a scale of 1 to 4. The critics are not perfect; here are five data points including the critics' scores and the performance of the movie:

#	Movie Name	A	B	Profit?
1	~~Pellet Power~~	1	1	-
2	~~Ghosts~~	3	2	+
3	~~[illegible] Bay~~	2	4	+
4	~~[illegible] Pizza~~	3	4	+
5	~~Endless Maze~~	2	3	-



1. First, you would like to examine the linear separability of the data. Plot the data on the 2D plane above; label profitable movies with + and non-profitable movies with − and determine if the data are linearly separable.

   From a visual diagnostic, yes it is linearly separable. There are 3 support vectors

2. Now you decide to use a perceptron to classify your data. Suppose you directly use the scores given above as features, together with a bias feature. That is $f_0 = 1$, $f_1 =$ score given by A and $f_2 =$ score given by B.

   Run one pass through the data with the perceptron algorithm, filling out the table below. Go through the data points in order, e.g. using data point #1 at step 1.

step	Weights	Score	Correct?
1	[-1, 0, 0]	$-1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 = -1$	yes
2	[-1, 0, 0]	-1*1+3*0+2*0 = -1	NO!
3	[0, 3, 2]	0*1+3*2+2*4 = 14	yes
4	[0, 3, 2]	0*1+3*3+4*2 = 17	yes
5	[0, 3, 2]	0*1+3*2+2*3 = 12	no

Invoke learning / update subr

   [-1, 1, -1]

   Final weights:

3. Have weights been learned that separate the data?

   No.
   Test Case on point 3:
   -1*1+2*1-1*4=-3

   But hey, that's cool. just keep training til you get it :)

4. More generally, irrespective of the training data, you want to know if your features are powerful enough to allow you to handle a range of scenarios. Circle the scenarios for which a perceptron using the features above can indeed perfectly classify movies which are profitable according to the given rules:

   (a) Your reviewers are awesome: if the total of their scores is more than 8, then the movie will definitely be profitable, and otherwise it won't be. x+y = 8 defines a linear hyperplane, we can work with this

   (b) Your reviewers are art critics. Your movie will be profitable if and only if each reviewer gives either a score of 2 or a score of 3. again, another kidn of weird criteria

   (c) Your reviewers have weird but different tastes. Your movie will be profitable if and only if both reviewers agree. need something more expressive / nonlinear

YES

outine

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:

chicken before the egg

$$P(X = k) = \underbrace{p}_{\text{success}}\underbrace{(1-p)^{k-1}}_{\text{not success}} \quad (1)$$

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?

geometric distributions are the probability distribution for the random variable that counts the number of successes

d	x	P(x)
1	5	p(1-p)^4
2	8	p(1-p)^7
3	3	p(1-p)^2
4	5	p(1-p)^4
5	7	p(1-p)^6

Assumption: Assume each sampling of the geometric distribution is independent of the other ones (seems reasonable, and, as a geometric distribution we're targeting, this is known)

Since we KNOW (or assume) it's a geometric distribution, our model space can be specified against the value of p in [0, 1]. Great, our model space is now defined by our choice of p. (think argmax / optimize some score against p. YA BOI CALCULUS RDY TO ROCK SOME COCKS)

We define the "goodness" of a model by the one that, when effected, has the "highest probability" that it TOTALLY saw the data set coming.
Specifically, P(D | p) = $\Pi_d P(d|p) = p^5(1-p)^{23}$

mmm. wolfram alpha can do this right? it can. function is maximized at p =5/28