Lecture 8

Oct 15th 2008

Bayes Classifiers in a nutshell

1. Learn the $P(X_1, X_2, \ldots X_m \mid Y=v_i)$ for each value v_i
3. Estimate $P(Y=v_i)$ as fraction of records with $Y=v_i$.
4. For a new prediction:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y=v \mid X_1=u_1 \cdots X_m=u_m)$$
$$= \underset{v}{\operatorname{argmax}}\, P(X_1=u_1 \cdots X_m=u_m \mid Y=v)P(Y=v)$$

Estimating the joint distribution of *X1*, *X2*, … X*m* *given y* can be problematic!

Joint Density Estimator Overfits

- Typically we don’t have enough data to estimate the joint distribution accurately
- So we make some bold assumptions to simplify the joint distribution

Naïve Bayes Assumption

- Assume that each attribute is independent of any other attributes given the class label

$$\boxed{\begin{aligned}&P(X_1 = u_1 \cdots X_m = u_m \mid Y = v_i)\\&= P(X_1 = u_1 \mid Y = v_i) \cdots P(X_m = u_m \mid Y = v_i)\end{aligned}}$$

A note about independence

- Assume A and B are Boolean Random Variables. Then

"A and B are independent"

if and only if

P(A|B) = P(A)

- "A and B are independent" is often notated as

$$A \perp B$$

Independence Theorems

- Assume $P(A|B) = P(A)$
- Then $P(A \wedge B) =$

$= P(A)\,P(B)$

- Assume $P(A|B) = P(A)$
- Then $P(B|A) =$

$= P(B)$

Independence Theorems

- Assume $P(A|B) = P(A)$
- Then $P(\sim A|B) =$

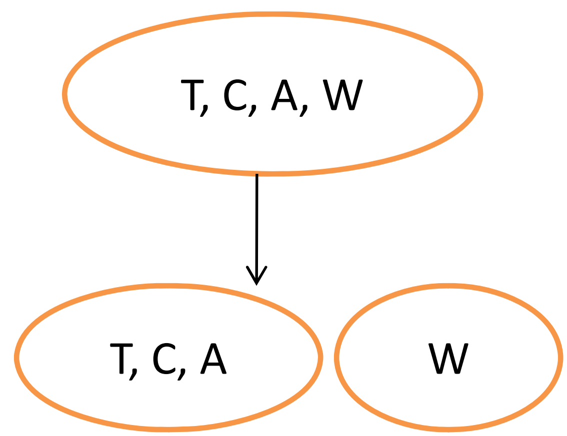
$= P(\sim A)$

- Assume $P(A|B) = P(A)$
- Then $P(A|\sim B) =$

$= P(A)$

Examples of independent events

- Two separate coin tosses
- Consider the following four variables:
  - T: Toothache (I have a toothache)
  - C: Catch (dentist's steel probe catches in my tooth)
  - A: Cavity
  - W: Weather
  - p(T, C, A, W) =p(T, C, A) p(W)



Conditional Independence

- $p(X_1|X_2,y) = p(X_1|y)$
  - X_1 and X_2 are conditionally independent given y
- If X_1 and X_2 are conditionally independent given y, then we have
  - $p(X_1,X_2|y) = p(X_1|y)\, p(X_2|y)$

Example of conditional independence

- T: Toothache (I have a toothache)
- C: Catch (dentist's steel probe catches in my tooth)
- A: Cavity

T and C are conditionally independent given A: P(T, C|A) =P(T|A)*P(C|A)

So , **events that are not independent from each other might be conditionally independent given some fact**

It can also happen the other way around. **Events that are independent might become conditionally dependent given some fact.**

B=Burglar in your house; A = Alarm **(**Burglar**)** rang in your house
E = Earthquake happened
B is independent of E (ignoring some possible connections between them)
However, if we know A is true, then B and E are no longer independent. Why?
P(B|A) >> P(B|A, E) Knowing E is true makes it much less likely for B to be true

Naïve Bayes Classifier

- Assume you want to predict output Y which has arity n_Y and values v_1, v_2, … v_{ny}.
- Assume there are m input attributes called X=(X_1, X_2, … X_m)
- Learn a conditional distribution of p(X|y) for each possible y value, y = v_1, v_2, … v_{ny}, we do this by:
  - Break training set into n_Y subsets called DS_1, DS_2, … DS_{ny} based on the y values, i.e., DS_i = Records in which $Y=v_i$
  - For each DS_i , learn a joint distribution of input distribution

$$\boxed{\begin{aligned}&P(X_1 = u_1 \cdots X_m = u_m \mid Y = v_i)\\ &= P(X_1 = u_1 \mid Y = v_i) \cdots P(X_m = u_m \mid Y = v_i)\end{aligned}}$$

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \mid Y = v) \cdots P(X_m = u_m \mid Y = v) P(Y = v)$$

Example

X_1	X_2	X_3	Y
1	1	1	0
1	1	0	0
0	0	0	0
0	1	0	1
0	0	1	1
0	1	1	1

Apply Naïve Bayes, and make prediction for (1,0,1)?

1. Learn the prior distribution of y.
   P(y=0)=1/2, **P**(y=1)=1/2
2. Learn the conditional distribution of x_i given y for each possible y values
   p(X_1|y=0), **p**(X_1|y=1)
   p(X_2|y=0), **p**(X_2|y=1)
   p(X_3|y=0), **p**(X_3|y=1)

For example, **p**(X_1|y=0):
P(X_1=1|y=0)=2/3, **P**(X_1=0|y=0)=1/3
…

To predict for (1,0,1):

P(y=0|(1,0,1)) = P((1,0,1)|y=0)P(y=0)/P((1,0,1))

P(y=1|(1,0,1)) = P((1,0,1)|y=1)P(y=1)/P((1,0,1))

Final Notes about (Naïve) Bayes Classifier

- Any density estimator can be plugged in to estimate $p(X_1,X_2, \ldots, X_m | y)$, or $p(X_i|y)$ for Naïve bayes
- Real valued attributes can be modeled using simple distributions such as Gaussian (Normal) distribution
- Zero probabilities are painful for both joint and naïve. A hack called Laplace smoothing can help!
  - Original estimation:

  $P(X_1=1|y=0) =$ (# of examples with y=0, X_1=1)/(# of examples with y=0)
  - Smoothed estimation (never estimate zero probability):

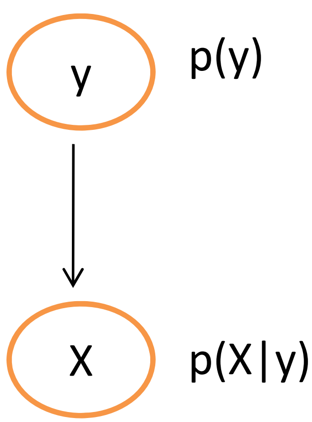
  $P(X_1=1|y=0) =$

  (**1+** # of examples with y=0, X_1=1) /(**k+** # of examples with y=0)
- Naïve Bayes is wonderfully cheap and survives tens of thousands of attributes easily

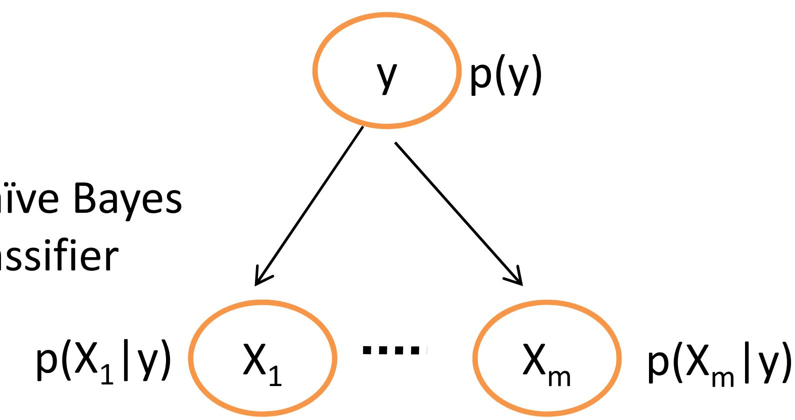
Bayes Classifier is a ***Generative Approach***

- Generative approach:
  - Learn p(y), p(X|y), and then apply bayes rule to compute p(y|X) for making predictions
  - This is in essence assuming that each data point is *independently, identically distributed (i.i.d),* and generated following a ***generative process*** governed by p(y) and p(X|y)

Bayes classifier



Naïve Bayes classifier



- Generative approach is just one type of learning approaches used in machine learning
  - Learning a correct generative model is difficult
  - And sometimes unnecessary
- KNN and DT are both what we call discriminative methods
  - They are not concerned about any generative models
  - They only care about finding a good discriminative function
  - For KNN and DT, these functions are deterministic, not probabilistic
- One can also take a probabilistic approach to learning discriminative functions
  - i.e., Learn $p(y|X)$ directly without assuming X is generated based on some particular distribution given y (i.e., $p(X|y)$)
  - Logistic regression is one such approach

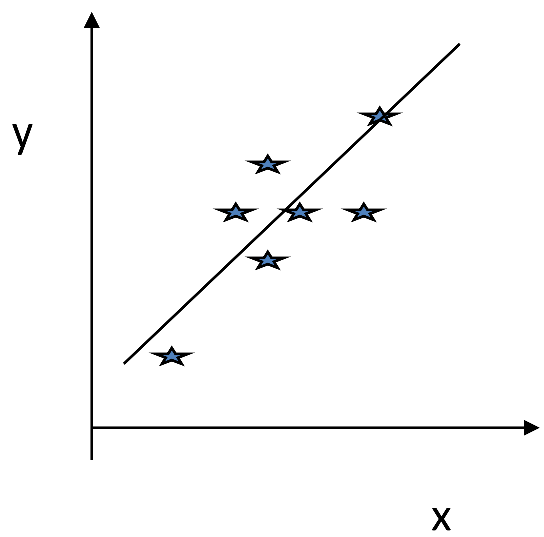
Logistic Regression

- First let’s look at the term regression
- Regression is similar to classification, except that the y value we are trying to predict is a continuous value (as opposed to a categorical value)

Classification: Given income, savings, predict loan applicant as “high risk” vs “low risk”

Regression: Given income, savings, predict credit score

Linear regression



- Essentially try to fit a straight line through a clouds of points
- Look for $w=[w_1,w_2,\ldots,w_m]$

  $\hat{y} = w_0+w_1x_1+\ldots+w_mx_m$ and $\hat{y}$ is as close to y as possible
- Logistic regression can be think of as extension of linear regression to the case where the target value y is binary

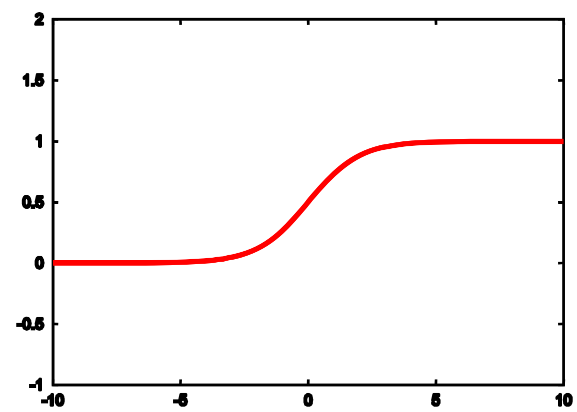
Logistic Regression

- Because y is binary (0, or 1), we can not directly use linear function of x to predict y
- Instead, we use linear function of x to predict the log odds of y=1:

$$\log \frac{P(y=1|x)}{P(y=0|x)} = w_0 + w_1 x_1 + \ldots + w_m x_m$$

- Or equivalently, we predict:

$$P(y=1|x) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + \ldots + w_m x_m)}}$$



Sigmoid function

Learning **w** for logistic regression

- Given a set of training data points, we would like to find a weight vector ***w*** such that $P(y=1|x)=\frac{1}{1+e^{-(w_0+w_1x_1+...+w_mx_m)}}$ is large (e.g. 1) for positive training examples, and small (e.g. 0) otherwise
- This can be captured in the following objective function:

Note that the superscript *i* is an index to the examples in the training set

$$L(\mathbf{w})=\sum_i \log P(y^i|\mathbf{x}^i,\mathbf{w})$$
$$=\sum_i [y^i \log P(y^i=1|\mathbf{x}^i,\mathbf{w})+(1-y^i)\log(1-P(y^i=1|\mathbf{x}^i,\mathbf{w}))]$$

This is call the likelihood function of **w,** and by maximizing this objective function, we perform what we call “maximum likelihood estimation” of the parameter **w.**

Optimizing L(w)

- Unfortunately this does not have a close form solution
- Instead, we iteratively search for the optimal w
- Start with a random w, iteratively improve w (similar to Perceptron)

Logistic regression learning

Given : training examples $(\mathbf{x}^i, y^i),\ i = 1,\ldots,N$

Let $\mathbf{w} \leftarrow (0,0,0,\ldots,0)$

Repeat until convergence

$\mathbf{d} \leftarrow (0,0,0,\ldots,0)$

For $i = 1$ to N do

$$\hat{y} \leftarrow \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}^i}}$$

$$error = y^i - \hat{y}$$

$$\mathbf{d} = \mathbf{d} + error \cdot \mathbf{x}^i$$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta d$$

Learning rate

Logistic regression learns LTU

- We predict y=1 if P(y=1|X)>P(y=0|X)
- You can show that this lead to a linear decision boundary