EE496 : COMPUTATIONAL INTELLIGENCE
PA01 : BAYESIAN DECISION

UGUR HALICI
METU: Department of Electrical and Electronics Engineering (EEE)
METU-Hacettepe U: Neuroscience and Neurotechnology (NSNT)

Probability

- For a given X, the classifiers we learned so far give a single predicted y value
- In contrast, a probabilistic prediction returns a probability over the output space

  P(y=0|X), P(y=1|X)
- We can easily think of situations when this would be very useful!
  - Given P(y=1|X) =0.49, P(y=-1|X)=0.51, how would you predict?
  - What if I tell you it is much more costly to miss an positive example than the other way around?

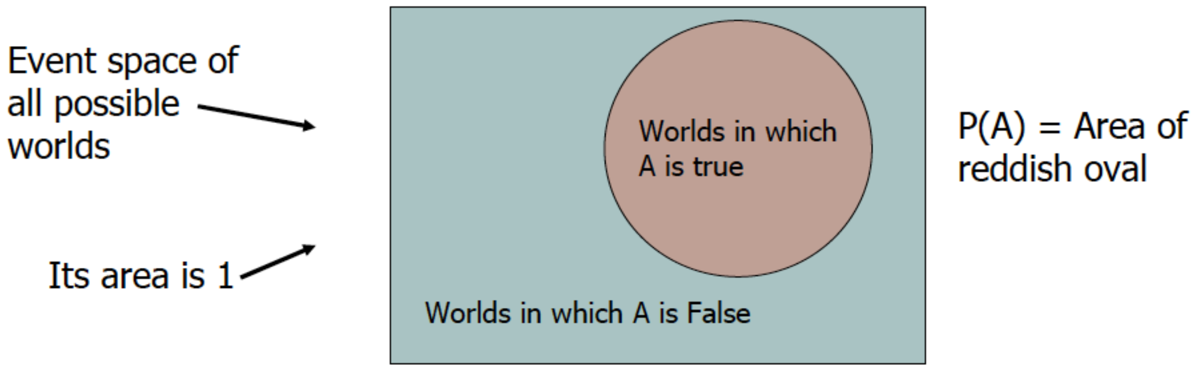
Discrete Random Variables

- A is a Boolean-valued random variable if A denotes an event, and there is some degree of uncertainty as to whether A occurs.

Examples

- A = The US president in 2023 will be male
- A = You wake up tomorrow with a headache
- A = You have Ebola

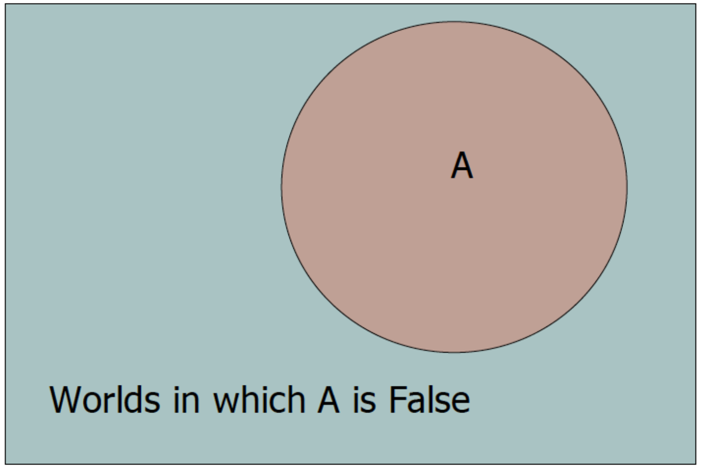
Visualizing A

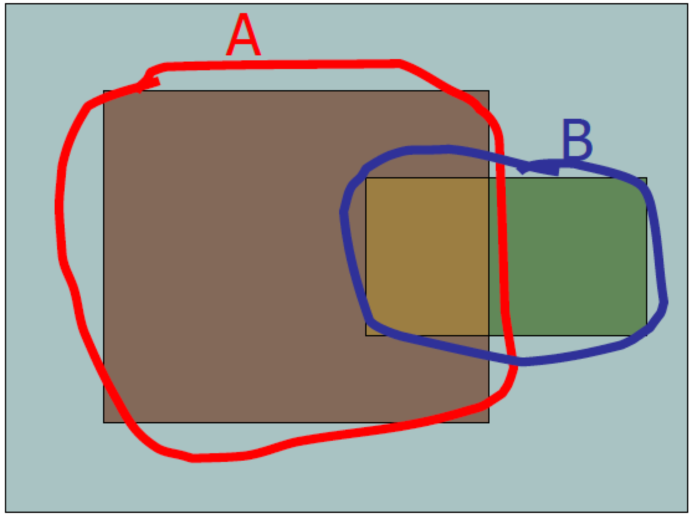


Basic axioms and theorems

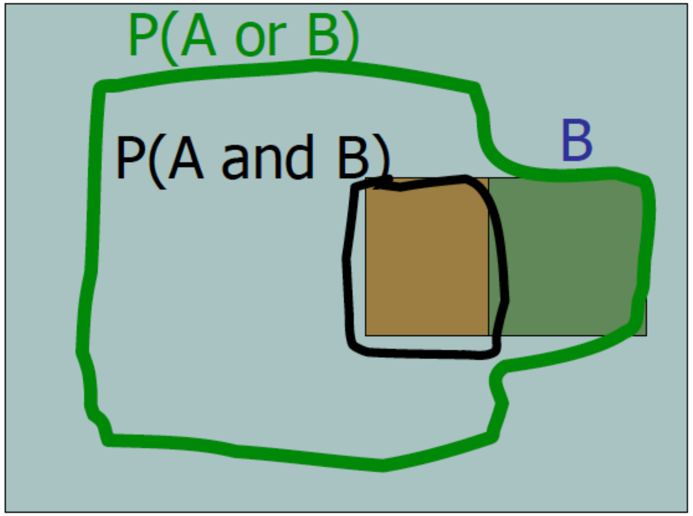
- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)

- P(~A) + P(A) = 1
- P(B) = P(B ^ A) + P(B ^ ~A)





Simple addition and subtraction



Multivalued Random Variables

- Suppose A can take on more than 2 values
- A is a random variable with arity k if it can take on exactly one value out of $\{v_1, v_2, .. \, v_k\}$
- Thus…

$$P(A = v_i \wedge A = v_j) = 0 \text{ if } i \neq j$$
$$P(A = v_1 \vee A = v_2 \vee A = v_k) = 1$$

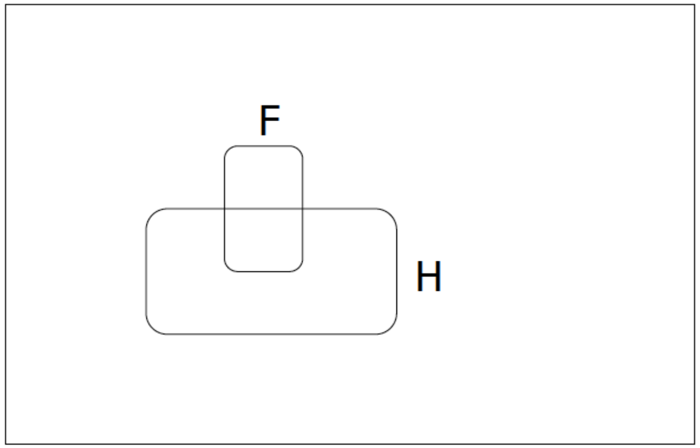
$$P(A = v_1 \vee A = v_2 \vee A = v_i) = \sum_{j=1}^{i} P(A = v_j)$$

$$P(B \wedge [A = v_1 \vee A = v_2 \vee A = v_i]) = \sum_{j=1}^{i} P(B \wedge A = v_j)$$

$$P(B) = \sum_{j=1}^{k} P(B \wedge A = v_j)$$

Conditional Probability

- P(A|B) = Fraction of worlds in which B is true that also have A true



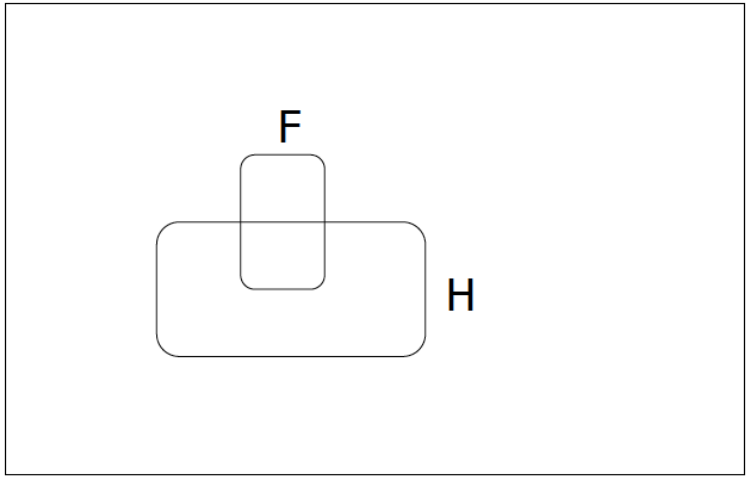
H = "Have a headache"
F = "Coming down with Flu"

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

"Headaches are rare and flu is rarer, but if you're coming down with 'flu there's a 50-50 chance you'll have a headache."

Conditional Probability

•



H = "Have a headache"
F = "Coming down with Flu"

$P(H) = 1/10$
$P(F) = 1/40$
$P(H|F) = 1/2$

$P(H|F)$ = Fraction of flu-inflicted worlds in which you have a headache

$$= \frac{\text{\#worlds with flu and headache}}{\text{\#worlds with flu}}$$

$$= \frac{\text{Area of "H and F" region}}{\text{Area of "F" region}}$$

$$= \frac{P(H \wedge F)}{P(F)}$$

Conditional Probability

- Definition of Conditional Probability

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

- Corollary: The Chain Rule

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

Probabilistic Inference

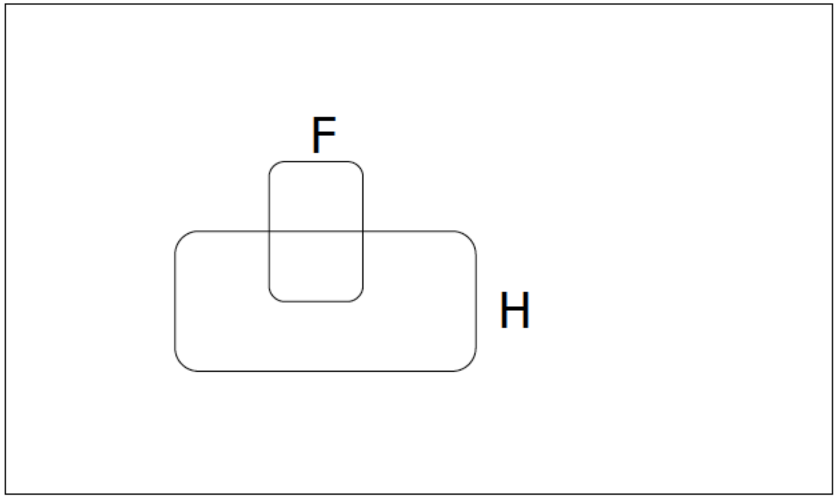
H = “Have a headache”

F = “Coming down with Flu”

$P(H) = 1/10$

$P(F) = 1/40$

$P(H|F) = 1/2$

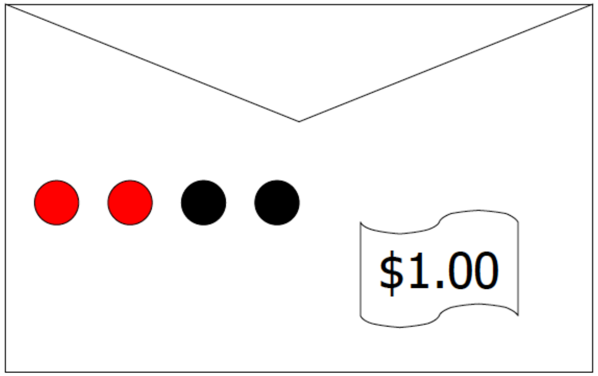


One day you wake up with a headache. You think: “Drat! 50% of flus are associated with headaches so I must have a 50-50 chance of coming down with flu”

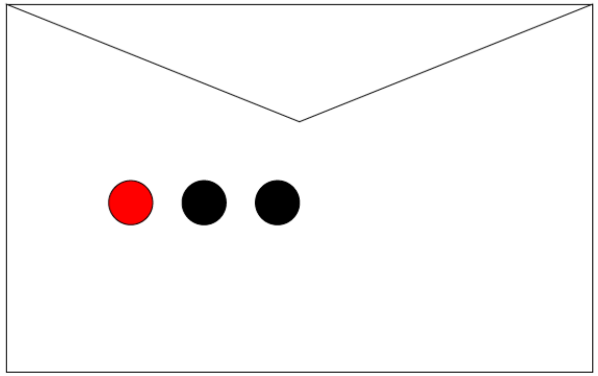
Is this reasoning good?

Bayes Rule

$$P(B|A) = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B)\,P(B)}{P(A)}$$



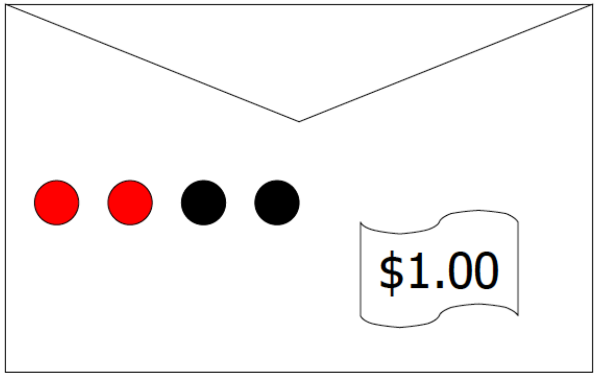
The "Win" envelope has a dollar and four beads in it



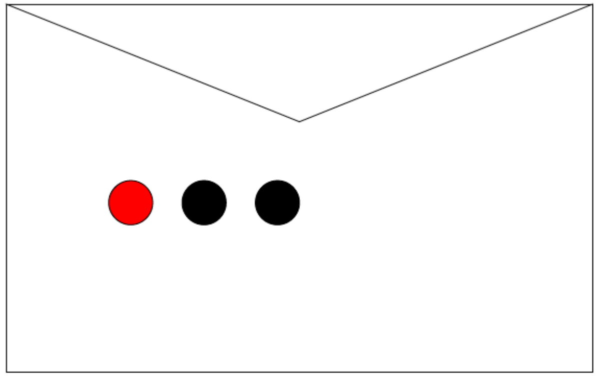
The "Lose" envelope has three beads and no money

Trivial question: someone draws an envelope at random and offers to sell it to you. How much should you pay?

Using Bayes Rule to Gamble



The "Win" envelope has a dollar and four beads in it



The "Lose" envelope has three beads and no money

Interesting question: before deciding, you are allowed to see one bead drawn from the envelope.

- Suppose it's black: How much should you pay?
- Suppose it's red: How much should you pay?

Continuous Probability Distribution

A continuous random variable x can take

any value in an interval on the real line

- X usually corresponds to some real-valued measurements, e.g., today's lowest temperature
- It is not possible to talk about the probability of a continuous random variable taking an exact value --- $P(x=56.2)=0$
- Instead we talk about the probability of the random variable taking a value within a given interval $P(x \in [50, 60])$

If $f(x1)=\alpha * a$ and $f(x2)=a$

Then when x is sampled from this distribution, you are α times more likely to see that x is "very close to" x1 than that x is "very close to" x2

Some commonly used distributions

Bernoulli distribution: Ber(p)

$$P(x)=\begin{cases}1-p & \text{for } x=0\\ p & \text{for } x=1\end{cases} \quad\Rightarrow\quad P(x)=p^x(1-p)^{1-x}$$

Binomial distribution: Binomial(n , p)

the probability to see x heads out of n flips

$$P(x)=\frac{n(n-1)\cdots(n-x+1)}{x!}p^x(1-p)^{n-x}$$

Multinomial distribution: Multinomial(n , [x_1, x_2, …, x_k])

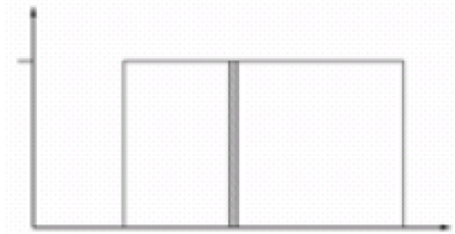
The probability to see x_1 ones, x_2 twos, etc, out of n dice rolls

$$P([x_1,x_2,\ldots,x_k])=\frac{n!}{x_1!x_2!\cdots x_k!}\theta_1^{x_1}\theta_2^{x_2}\cdots\theta_k^{x_k}$$

-

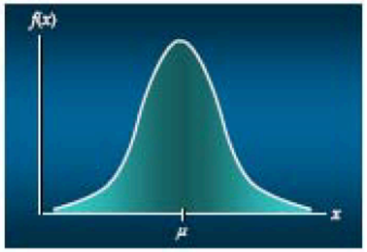
Uniform Probability Density Function

$$f(x) = 1/(b-a) \quad \text{for } a \le x \le b$$
$$= 0 \quad \text{elsewhere}$$



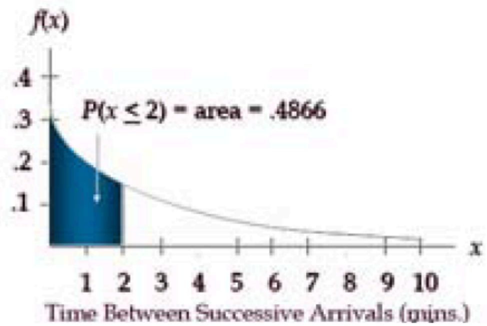
Normal (Gaussian) Probability Density Function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$



Exponential Probability Distribution

$$f(x) = \frac{1}{\mu} e^{-x/\mu}$$



The Joint Distribution

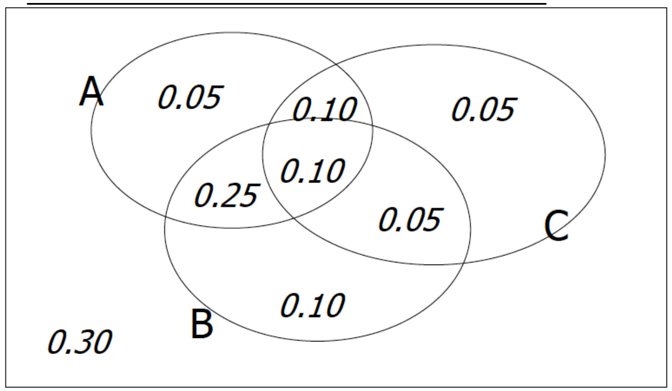
Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables then the table will have 2^M rows).

2. For each combination of values, say how probable it is.

3. If you subscribe to the axioms of probability, those numbers must sum to 1.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

One you have the JD you can ask for the probability of any logical expression involving your attribute

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

P(Poor Male) = 0.4654

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum\limits_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum\limits_{\text{rows matching } E_2} P(\text{row})}$$

P(Male | Poor) = 0.4654 / 0.7604 = 0.612

So we have learned that

Joint distribution is extremely useful! we can do all kinds of cool inference

- I've got a sore neck: how likely am I to have meningitis?
- Many industries grow around Beyesian Inference: examples include medicine, pharma, Engine diagnosis etc.

But, HOW do we get them?

- We can learn from data

Learning a joint distribution

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	**0.25**
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

Bayes Rule

•

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(B|A)\,P(A)}{P(B)}$$

More general forms:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\sim A)P(\sim A)}$$

$$P(A|B \wedge X) = \frac{P(B|A \wedge X)P(A \wedge X)}{P(B \wedge X)}$$

Bayes Classifiers

- 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 $X=(X_1 , X_2 , \ldots X_m)$
- Learn a conditional distribution of p(X|y) for each possible y value, $y = v_1 , v_2 , \ldots v_{ny}$, 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
  - This will give us $p(X| Y=v_i)$, i.e., $P(X_1, X_2, \ldots X_m \mid Y=v_i)$

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

Bayes Classifiers in a nutshell

1. Learn the P(X1 , X2 , … Xm | Y=vi) for each value vi
2. Estimate P(Y=vi) as fraction of records with Y=vi .
3. For a new prediction:

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

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

Example: Spam Filtering

- Bag-of-words representation is used for emails ($X = \{x_1, x_2, \ldots, x_m\}$)
- Assume that we have a dictionary containing all commonly used words and tokens
- We will create one attribute for each dictionary entry
  - E.g., x_i is a binary variable, $x_i = 1$ (0) means the i th word in the
  - dictionary is (not) present in the email
  - Other possible ways of forming the features exist,

    e.g., x_i =the # of times that the ith word appears
- Assume that our vocabulary contains 10k commonly used words --- we have 10,000 attributes
- How many parameters that we need to learn?

- $2*(2^{10,000} - 1)$

Bayes Classifiers

Naïve Bayes Assumption

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

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

Independence Theorems:

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

- P(A^B) = P(A) P(B)
- P(B|A) = P(B)
- P(~A|B) =P(~A)
- P(A|~B) =P(A)

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
  – $P(X_1,X_2|y) = P(X_1|y)\ P(X_2|y)$

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 $A \perp B$

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

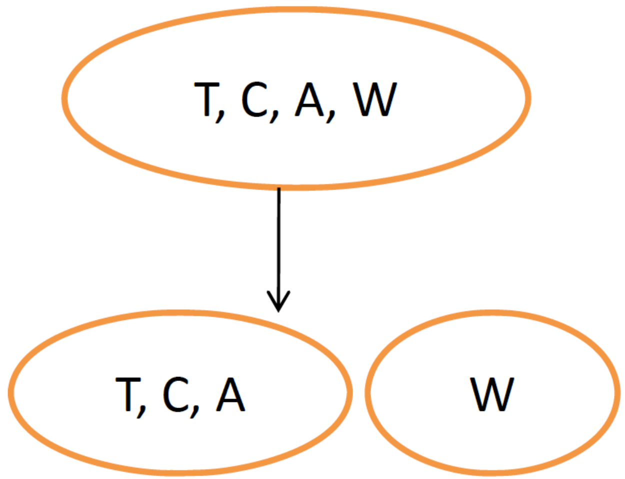
Example

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 makeprediction for (1,1,1)?

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)



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**

Example of conditional independence

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

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

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

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
   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
…

predict for (1,0,1):

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

(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

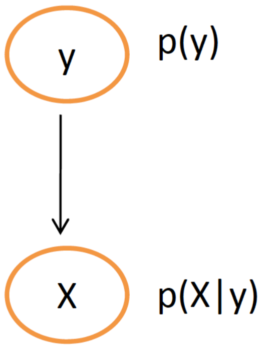
  $p(X_1,X_2, \ldots, X_m \mid y)$, or $p(Xi|y)$ for Naïve bayes
- Real valued attributes can be modeled using simple distributions such as Gaussian (Normal) distribution
- Zero probabilities are painful for both joint and naïve. A hack called Laplace smoothing can help!
  - Original estimation:

  $P(X_1=1|y=0)$ =(# of examples with y=0, X_1=1)/(# of examples with y=0)
  - Smoothed estimation (never estimate zero probability):
  - $P(X1=1|y=0)$ = (**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

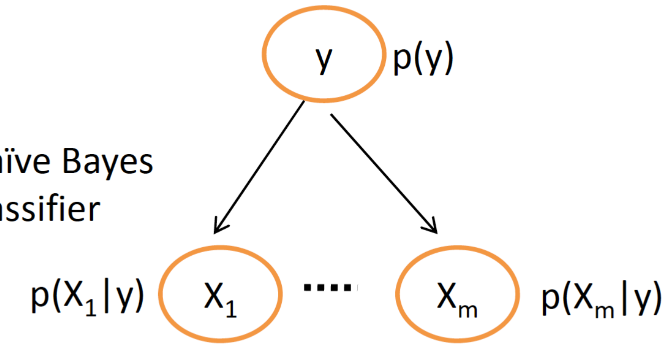
Bayes Classifier is a Generative Approach

- 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

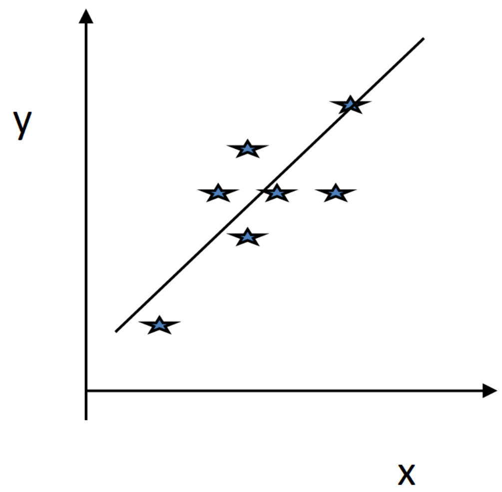
Logistic Regression

- 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 y
- Logistic regression can be think of as extension of linear regression to the case where the target value y is x binary



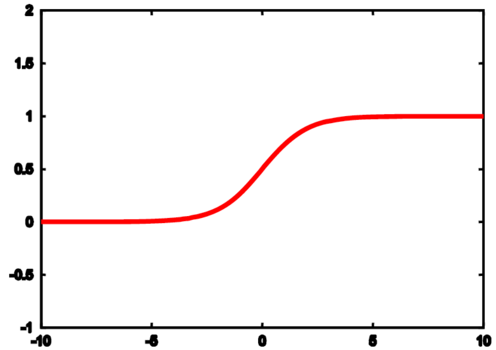
Logistic Regression

- 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:

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

- Or equivalently, we predict:

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



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

$$P(y=1\mid x)=\frac{1}{1+e^{-(w_0+w_1x_1+\ldots+w_mx_m)}}$$

is large (e.g. 1) for positive training examples, and small (e.g. 0) otherwise

- This can be captured in the following objective function:

$$L(\mathbf{w})=\sum_i \log P(y^i\mid \mathbf{x}^i,\mathbf{w})$$

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

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

- 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 $(\mathbf{x}^i, y^i)$, $i = 1,\dots,N$

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

Repeat until convergence

$\quad \mathbf{d} \leftarrow (0,0,0,\dots,0)$

$\quad$ For $i = 1$ to N do

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

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

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

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

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