Naïve Bayes Classification

Background

- There are three methods to establish a classifier

  a) Model a classification rule directly

  Examples: k-NN, decision trees, perceptron, SVM

  b) Model the probability of class memberships given input data

  Example: Logistic Regression

  c) Make a probabilistic model of data within each class

  Examples: naive Bayes, model based classifiers

- *a*) and *b*) are examples of discriminative classification
- *c*) is an example of generative classification
- *b*) and *c*) are both examples of probabilistic classification

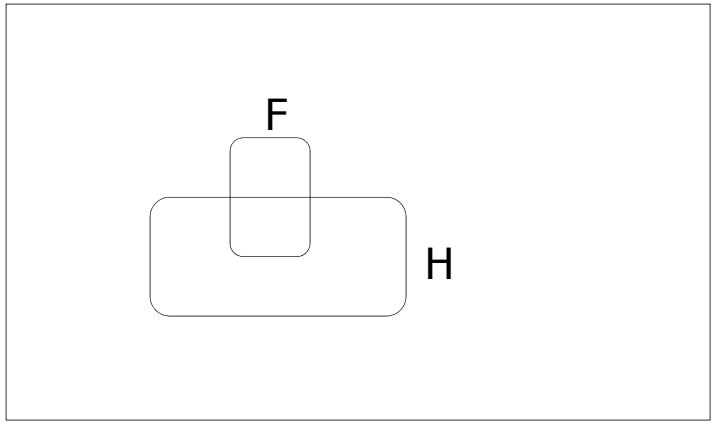
Probability Basics

- Prior, conditional and joint probability
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
  - Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$
- Bayesian Rule

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

For normalization

A side note: probabilistic inference



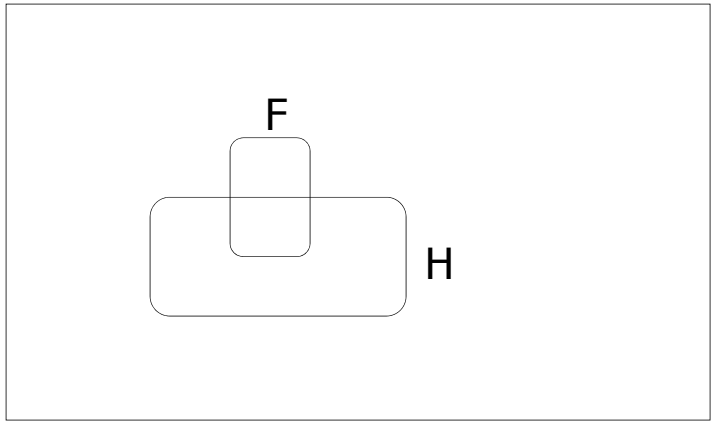
H = "Have a headache"
F = "Coming down with Flu"

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?

Probabilistic Inference



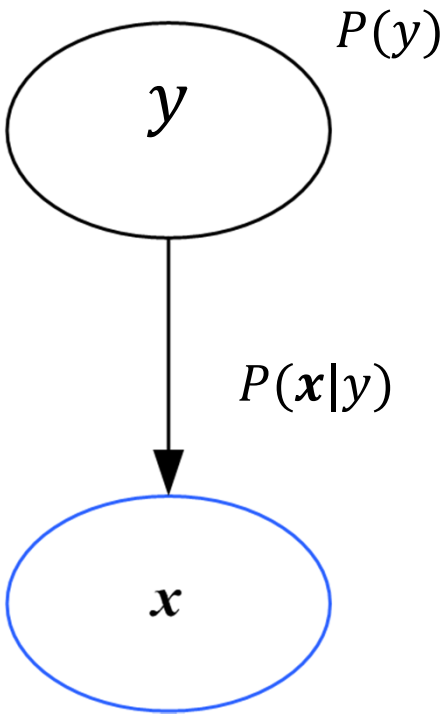
H = "Have a headache"
F = "Coming down with Flu"

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

$$P(F \wedge H) = P(F)P(H \mid F) = \frac{1}{40} * \frac{1}{2} = \frac{1}{80}$$

$$P(F \mid H) = \frac{P(F \wedge H)}{P(H)} = \frac{1}{8}$$

Bayes classifier



A simple bayes net

Likelihood

prior

posterior

$$P(y \mid \mathbf{x}) = \frac{P(y)\, p(\mathbf{x} \mid y)}{p(\mathbf{x})}$$

Given a set of training examples, to build a Bayes classifier, we need to

1. Estimate P(y) from data
2. Estimate P(x|y) from data

Given a test data point x, to make prediction

1. Apply bayes rule: $P(y \mid \mathbf{x}) \propto P(y) P(\mathbf{x} \mid y)$
2. Predict $\arg\max_y P(y \mid \mathbf{x})$

Maximum a Posteriori (MAP)

- Given data D and a set of classes C = {c_1, c_2, …, c_n}, the most probable class is the one having the highest value of posterior probability:

$$c_{MAP} = argmax_{c \in C}\, P(c|D)$$

$$= argmax_{c \in C}\, P(D|c)\, P(c)$$

For all the classes, find the value of $P(D|c)\, P(c)$, the class with the highest value is the most likely class, given the data D.

Bayes Classifiers in a nutshell

1. Estimate P(x_1, x_2, … x_m | $y=v_i$) for each value v_i
3. Estimate P($y=v_i$) as fraction of records with $y=v_i$.

} *learning*

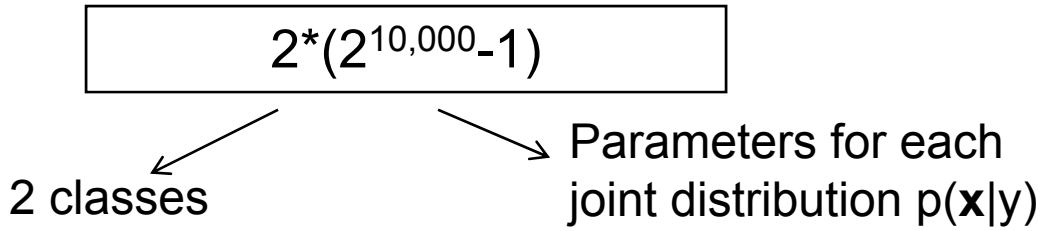
4. For a new prediction:

$$y^{\text{predict}} = \operatorname*{argmax}_v P(y = v \mid x_1 = u_1 \cdots x_m = u_m)$$
$$= \operatorname*{argmax}_v P(x_1 = u_1 \cdots x_m = u_m \mid y = v) P(y = v)$$

Estimating the joint distribution of x_1, x_2, … x_m given y can be problematic!

Example: Spam Filtering

- Assume that our vocabulary contains 10k commonly used words & tokens--- we have 10,000 attributes
- Let's assume these attributes are binary
- How many parameters that we need to learn?

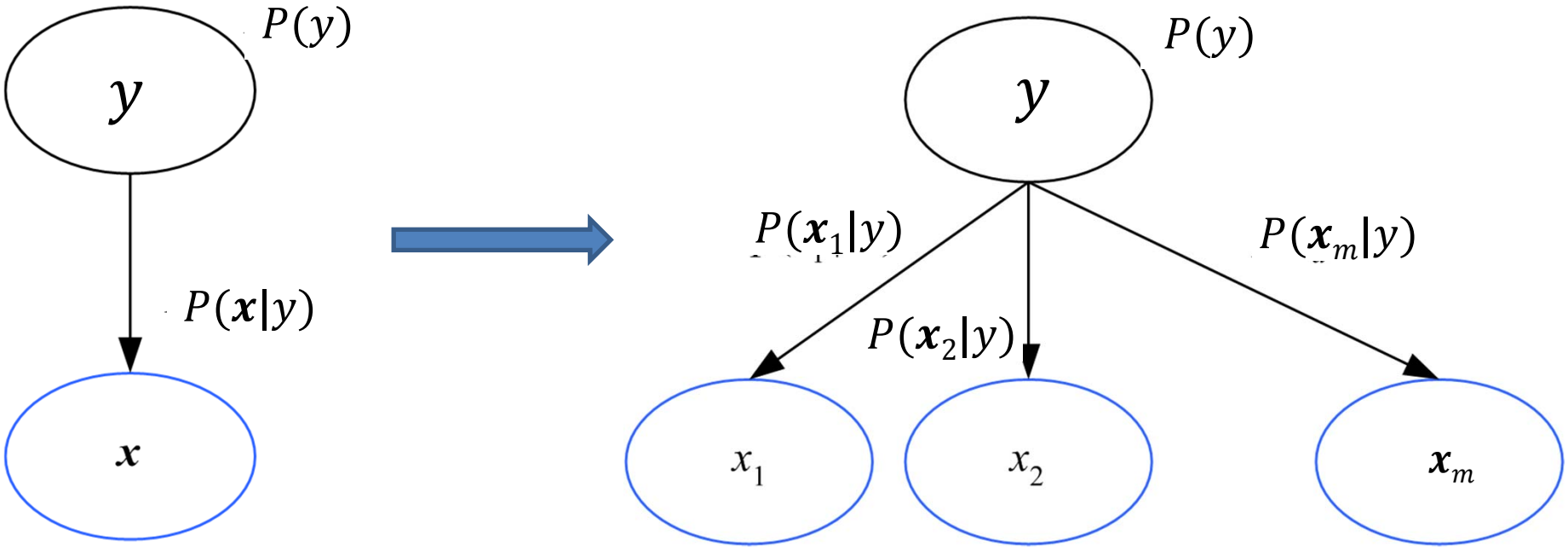


Clearly we don't have enough data to estimate that many parameters

The Naïve Bayes Assumption

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

$$P(x_1 = u_1 \cdots x_m = u_m \mid y = v_i)$$
$$= P(x_1 = u_1 \mid y = v_i) \cdots P(x_m = u_m \mid y = v_i)$$



Naïve Bayes Classifier

- By assuming that each attribute is independent of any other attributes given the class label, we now have a *Naïve* Bayes Classifier
- Instead of learning a joint distribution of all features, we learn $p(x_i|y)$ separately for each feature x_i
- Everything else remains the same

Naïve Bayes Classifier

- Assume you want to predict output *y* which has n_Y 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 S_1, S_2, … S_{ny} based on the y values, i.e., S_i contains examples in which $y=v_i$
  - For each S_i, learn p(y=v_i)= $|S_i|/|S|$
  - For each S_i , learn the conditional distribution each input features, e.g.:

$$\boxed{P(x_1 = u_1 \mid y = v_i), \cdots, P(x_m = u_m \mid y = v_i)}$$

$$y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(x_1 = u_1 \mid y = v) \cdots P(x_m = u_m \mid y = v) P(y = v)$$

Naïve Bayes

- Bayes classification

$$P(C \mid \mathbf{X}) \propto P(\mathbf{X} \mid C)P(C) = P(X_1, \cdots, X_n \mid C)P(C)$$

Difficulty: learning the joint probability $P(X_1, \cdots, X_n \mid C)$

- Naïve Bayes classification
  - Making the assumption that all input attributes are independent

$$\begin{aligned} P(X_1, X_2, \cdots, X_n \mid C) &= P(X_1 \mid X_2, \cdots, X_n; C)P(X_2, \cdots, X_n \mid C) \\ &= P(X_1 \mid C)P(X_2, \cdots, X_n \mid C) \\ &= P(X_1 \mid C)P(X_2 \mid C) \cdots P(X_n \mid C) \end{aligned}$$

  - MAP classification rule

Find the class c* such that likelihood of data is maximized

$$[P(x_1 \mid c^*) \cdots P(x_n \mid c^*)]P(c^*) > [P(x_1 \mid c) \cdots P(x_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Naïve Bayes Classifier

- Based on MAP rule
- Extremely simple yet powerful
- Idea: Given a dataset, find which class is most likely i.e. which class has highest posterior.
- This means you have to find the posterior for every class and then find the one that gives the max value.
- Computationally simple

Naïve Bayes

- Naïve Bayes Algorithm (for discrete input attributes)
  - Learning Phase: Given a training set S,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;

For every attribute value a_{jk} of each attribute x_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

$\hat{P}(X_j = a_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = a_{jk} \mid C = c_i)$ with examples in **S**;

Output: conditional probability tables; for x_j, $N_j \times L$ elements

  - Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$

Look up tables to assign the label c^* to **X'** if

$$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c) \cdots \hat{P}(a'_n \mid c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

P(Outlook=*o* | Play=*b*)

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

P(Temperature=*t* | Play=*b*)

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

P(Humidity=*h* | Play=*b*)

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

P(Wind=*w* | Play=*b*)

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

P(Play=*Yes*) = 9/14

P(Play=*No*) = 5/14

Example

- Test Phase
  - Given a new instance,

    $\mathbf{x}'$ =(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up tables

    P(Outlook=*Sunny* | Play=*Yes*) = 2/9

    P(Temperature=*Cool* | Play=*Yes*) = 3/9

    P(Huminity=*High* | Play=*Yes*) = 3/9

    P(Wind=*Strong* | Play=*Yes*) = 3/9

    P(Play=*Yes*) = 9/14

    P(Outlook=*Sunny* | Play=*No*) = 3/5

    P(Temperature=*Cool* | Play==*No*) = 1/5

    P(Huminity=*High* | Play=*No*) = 4/5

    P(Wind=*Strong* | Play=*No*) = 3/5

    P(Play=*No*) = 5/14
  - MAP rule

    P(*Yes* | $\mathbf{x}'$): [P(*Sunny* | Yes)P(*Cool* | *Yes*)P(*High* | Yes)P(*Strong* | *Yes*)]P(Play=*Yes*) = 0.0053

    P(*No* | $\mathbf{x}'$): [P(*Sunny* | No) P(*Cool* | No)P(*High* | *No*)P(*Strong* | *No*)]P(Play=*No*) = 0.0206

    Given the fact P(*Yes* | $\mathbf{x}'$) < P(*No* | $\mathbf{x}'$), we label $\mathbf{x}'$ to be "*No*".

Naive Bayesian Classifier Example

Outlook	Temperature	Humidity	Windy	Class
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
overcast	cool	normal	true	P
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P

9

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
rain	cool	normal	true	N
sunny	mild	high	false	N
rain	mild	high	true	N

5

Naive Bayesian Classifier Example

- Given the training set, we compute the probabilities:

Outlook	P	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5
Tempreature		
hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

Humidity	P	N
high	3/9	4/5
normal	6/9	1/5
Windy		
true	3/9	3/5
false	6/9	2/5

- We also have the probabilities
  - P = 9/14
  - N = 5/14

Naive Bayesian Classifier Example

- To classify a new sample X:
  - outlook = sunny
  - temperature = cool
  - humidity = high
  - windy = false
- Prob(P|X) = Prob(P)*Prob(sunny|P)*Prob(cool|P)* Prob(high|P)*Prob(false|P) = 9/14*2/9*3/9*3/9*6/9 = 0.01
- Prob(N|X) = Prob(N)*Prob(sunny|N)*Prob(cool|N)* Prob(high|N)*Prob(false|N) = 5/14*3/5*1/5*4/5*2/5 = 0.013
- Therefore X takes class label N

Laplace Smoothing

- With the Naïve Bayes Assumption, we can still end up with zero probabilities
- E.g., if we receive an email that contains a word that has never appeared in the training emails
  - P(**x**|y) will be 0 for all y values
  - We can only make prediction based on p(y)
- This is bad because we ignored all the other words in the email because of this single rare word
- Laplace smoothing can help

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

  = (**1+** # of examples with y=0, $X_1=1$) /(**k+** # of examples with y=0)

  Pretend that you saw each outcome once more than number of times it occured

  k = the total number of possible values of x

- For a binary feature like above, p(x|y) will not be 0

Relevant Issues

- Violation of Independence Assumption
  - For many real world tasks, $P(X_1,\cdots,X_n \mid C) \neq P(X_1 \mid C)\cdots P(X_n \mid C)$
  - Nevertheless, naïve Bayes works surprisingly well anyway!
- Zero conditional probability Problem
  - If no example contains the attribute value $X_j = a_{jk}$, $\hat{P}(X_j = a_{jk} \mid C = c_i) = 0$
  - In this circumstance, $\hat{P}(x_1 \mid c_i)\cdots\hat{P}(a_{jk} \mid c_i)\cdots\hat{P}(x_n \mid c_i) = 0$ during test
  - For a remedy, conditional probabilities estimated with **Laplace smoothing:**

$$\hat{P}(X_j = a_{jk} \mid C = c_i) = \frac{n_c + mp}{n + m}$$

n_c : number of training examples for which $X_j = a_{jk}$ and $C = c_i$

n : number of training examples for which $C = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of X_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)