Lecture 16: Naïve Bayes and Generative model

Winter 2018

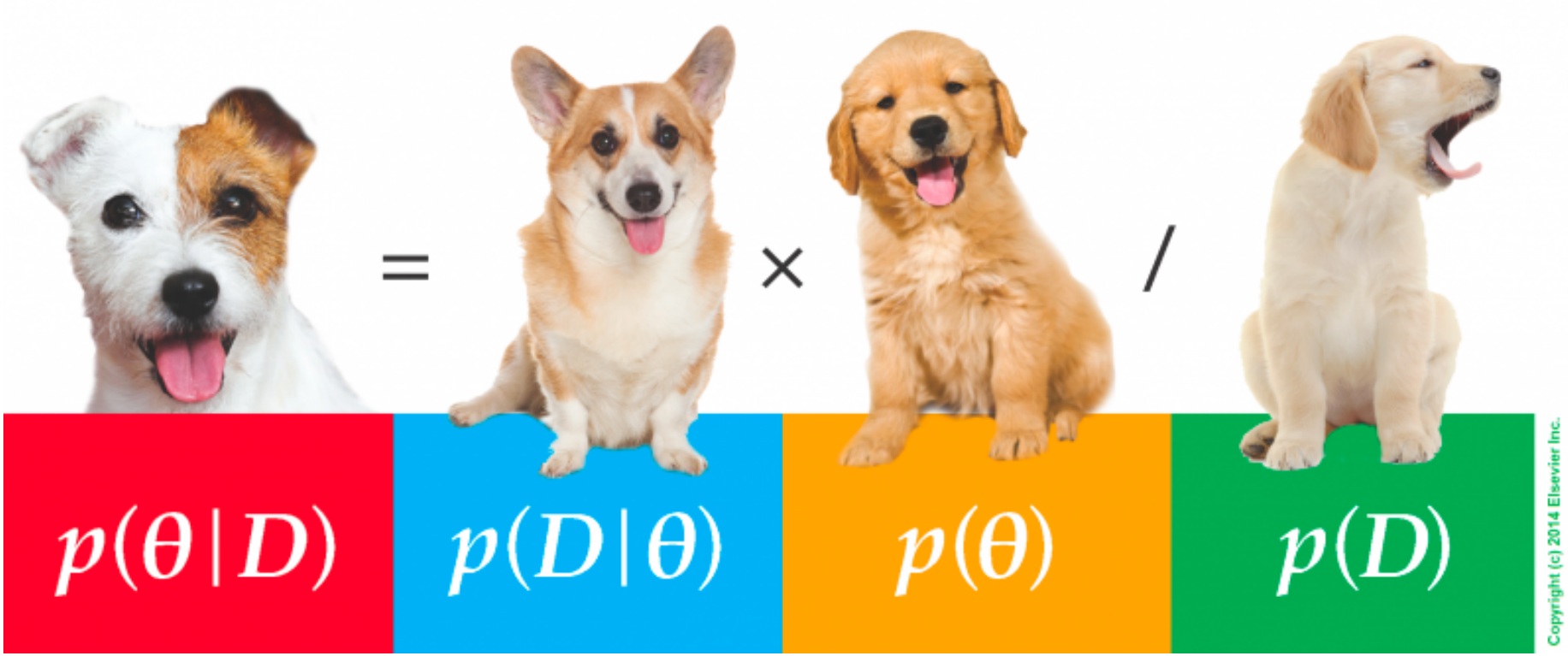
Kai-Wei Chang

CS @ UCLA

kw+cm146@kwchang.net

The instructor gratefully acknowledges Dan Roth, Vivek Srikuar, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Probabilistic models and Bayesian Learning



Probabilistic Learning

Two different notions of probabilistic learning

- Learning probabilistic concepts
  - The learned concept is a function $c:X \rightarrow [0,1]$
  - c(x) may be interpreted as the probability that the label 1 is assigned to x
  - The learning theory that we have studied before is applicable (with some extensions)

- Bayesian Learning: Use of a probabilistic criterion in selecting a hypothesis
  - The hypothesis can be deterministic, a Boolean function
  - The criterion for selecting the hypothesis is probabilistic

Bayesian Learning

Given a dataset D, we want to find the best hypothesis h
What does *best* mean?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Posterior probability: What is the probability that h is the hypothesis, given that the data D is observed?

Likelihood: What is the probability that this data point (an example or an entire dataset) is observed, given that the hypothesis is h?

What is the probability that the data D is observed (independent of any knowledge about the hypothesis)?

Prior probability of h: Background knowledge. What do we expect the hypothesis to be even before we see any data? For example, in the absence of any information, maybe the uniform distribution.

Choosing a hypothesis

Given some data, find the most probable hypothesis

- The Maximum a Posteriori hypothesis h_{MAP}

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

If we assume that the prior is uniform i.e. $P(h_i) = P(h_j)$, for all h_i, h_j

- Simplify this to get the Maximum Likelihood hypothesis

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Often computationally easier to maximize *log likelihood*

MAP v.s. MLE (Bernoulli trials)

- MLE:

$$\underset{p}{\operatorname{argmax}}\, a \log p + b \log(1-p)$$

$$\Rightarrow p_{best} = \frac{a}{a+b}$$

- MAP (with beta distribution as prior)

$$\underset{p}{\operatorname{argmax}}\ (a+\alpha-1)\log p + (b+\beta-1)\log(1-p)$$

$$\Rightarrow p_{best} = \frac{a+\alpha-1}{a+b+\alpha+\beta-2}$$

Learning a logistic regression classifier

Learning a logistic regression classifier is equivalent to solving

$$\min_{\mathbf{w}} \sum_{i}^{m} \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i) + \frac{1}{\sigma^2} \mathbf{w}^T \mathbf{w}$$

Likelihood

Prior

Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Generative model

Where are we?

We have seen Bayesian learning

- Using a probabilistic criterion to select a hypothesis
- Maximum a posteriori and maximum likelihood learning
- Question: What is the difference between them?

We could also learn functions that predict probabilities of outcomes

- Different from using a probabilistic criterion to learn

  Maximum a posteriori (MAP) prediction as opposed to MAP learning

MAP prediction

Let's be use the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y|X = \mathbf{x}) = \frac{P(X = \mathbf{x}|Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Posterior probability of label being y for this input x

MAP prediction

Let's be use the Bayes rule for predicting $\mathbf{y}$ given an input $\mathbf{x}$

$$P(Y = y|X = \mathbf{x}) = \frac{P(X = \mathbf{x}|Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Predict $\mathbf{y}$ for the input $\mathbf{x}$ using

$$\arg\max_y \frac{P(X = \mathbf{x}|Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

MAP prediction

Let's be use the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y|X = \mathbf{x}) = \frac{P(X = \mathbf{x}|Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Predict y for the input $\mathbf{x}$ using

$$\arg\max_{y} P(X = \mathbf{x}|Y = y)P(Y = y)$$

MAP prediction

Predict y for the input $\mathbf{x}$ using

$$\arg\max_{y} P(X = \mathbf{x}|Y = y)P(Y = y)$$

Likelihood of observing this input $\mathbf{x}$ when the label is y

Prior probability of the label being y

All we need are these two sets of probabilities

Example: Tennis

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Without any other information, what is the prior probability that I should play tennis?

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

On days that I do play tennis, what is the probability that the temperature is T and the wind is W?

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

On days that I don't play tennis, what is the probability that the temperature is T and the wind is W?

Example: Tennis again

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

Example: Tennis again

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

argmax_y P(H, W | play?) P (play?)

Example: Tennis again

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W | Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W | Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

argmax_y P(H, W | play?) P (play?)

P(H, W | Yes) P(Yes) = 0.4 x 0.3
= 0.12

P(H, W | No) P(No) = 0.1 x 0.7
= 0.07

Example: Tennis again

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

argmax_y P(H, W | play?) P (play?)

P(H, W | Yes) P(Yes) = 0.4 x 0.3
= 0.12

P(H, W | No) P(No) = 0.1 x 0.7
= 0.07

MAP prediction = Yes

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Temperature: H(ot), M(edium), C(ool)

Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Te

Hu

N(ormal), L(ow)

Wind: S(trong), W(eak)

We need to learn

1. The prior P(Play?)
2. The likelihoods P(X | Play?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Prior P(play?)

- A single number (Why only one?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: $P(x_1, x_2, x_3, x_4 \mid \text{Play?})$

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-
	3	3	3	2	

Values for this feature

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: $P(x_1, x_2, x_3, x_4 \mid \text{Play?})$

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-
	3	3	3	2	

Values for this feature

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: $P(x_1, x_2, x_3, x_4 \mid \text{Play?})$
- $(3 \cdot 3 \cdot 3 \cdot 2 - 1)$ parameters in each case

  One for each assignment

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

Need a lot of data to estimate these many numbers!

How hard is it to learn probabilistic models?

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

Need a lot of data to estimate these many numbers!

High model complexity

If there is very limited data, high variance in the parameters

How can we deal with this?

Answer: Make independence assumptions

Recall: Conditional independence

Suppose X, Y and Z are random variables

X is *conditionally independent* of Y given Z if the probability distribution of X is independent of the value of Y when Z is observed

$$P(X|Y,Z) = P(X|Z)$$

Or equivalently

$$P(X,Y|Z) = P(X|Z)P(Y|Z)$$

Modeling the features

$P(x_1, x_2, \cdots, x_d|y)$ required k(2^d – 1) parameters

What if all the features were conditionally independent given the label?

The Naïve Bayes Assumption

Modeling the features

$P(x_1, x_2, \cdots, x_d|y)$ required k(2^d – 1) parameters

What if all the features were conditionally independent given the label?

The Naïve Bayes Assumption

That is,

$$P(x_1, x_2, \cdots, x_d|y) = P(x_1|y)P(x_2|y)\cdots P(x_d|y)$$

Requires only d numbers for each label. kd parameters overall. Not bad!

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior P(y)
- For each x_j, we have the likelihood $P(x_j \mid y)$

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior P(y)
- For each x_j, we have the likelihood $P(x_j \mid y)$

Decision rule

$$h_{NB}(\boldsymbol{x}) = \underset{y}{\operatorname{argmax}}\, P(y)P(x_1, x_2, \cdots, x_d|y)$$

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior P(y)
- For each x_j, we have the likelihood $P(x_j \mid y)$

Decision rule

$$h_{NB}(\boldsymbol{x}) = \underset{y}{\operatorname{argmax}}\, P(y)P(x_1, x_2, \cdots, x_d|y)$$

$$= \underset{y}{\operatorname{argmax}}\, P(y) \prod_j P(x_j|y)$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Consider the two class case. We predict the label to be + if

$$P(y=+)\prod_j P(x_j|y=+) > P(y=-)\prod_j P(x_j|y=-)$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Consider the two class case. We predict the label to be + if

$$P(y=+)\prod_j P(x_j|y=+) > P(y=-)\prod_j P(x_j|y=-)$$

$$\frac{P(y=+)\prod_j P(x_j|y=+)}{P(y=-)\prod_j P(x_j|y=-)} > 1$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Taking log and simplifying, we can show that the decision boundary of naïve Bayes is a linear function

$$\log \frac{P(y = -|\boldsymbol{x})}{P(y = +|\boldsymbol{x})}$$

This is a linear function of the feature space!

Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Generative model

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities

hypothesis?

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j | y)$

hypothesis?

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j | y)$

Suppose we have a data set $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

A note on convention for this section:

- Examples in the dataset are indexed by the subscript i (e.g. $\boldsymbol{x}_i$)
- Features within an example are indexed by the subscript j
  - The j^{th} feature of the i^{th} example will be x_{ij}

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j | y)$

If we have a data set $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

And we want to learn the classifier in a probabilistic way

- What is a probabilistic criterion to select the hypothesis?

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Here h is defined by all the probabilities used to construct the naïve Bayes decision

Maximum likelihood estimation

Given a dataset $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

$$h_{ML} = \arg\max_{h} \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h)$$

Each example in the dataset is independent and identically distributed

So we can represent P(D| h) as this product

Maximum likelihood estimation

Given a dataset $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h)$$

Each example in the dataset is independent and identically distributed

So we can represent $\mathrm{P(D|\ h)}$ as this product

Asks "What probability would this particular h assign to the pair $(\mathbf{x}_\mathrm{i}, \mathrm{y}_\mathrm{i})$?"

Maximum likelihood estimation

Given a dataset D = $\{(x_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h)$$

$$= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h)P(y_i|h)$$

Maximum likelihood estimation

Given a dataset D = $\{(x_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h)$$

$$= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h) P(y_i|h)$$

$$= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h)$$

$\mathbf{x}_{ij}$ is the j^{th} feature of $\mathbf{x}_i$

The Naïve Bayes assumption

Maximum likelihood estimation

Given a dataset D = $\{(x_i, y_i)\}$ with m examples

$$
\begin{aligned}
h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h) P(y_i|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h)
\end{aligned}
$$

How do we proceed?

Maximum likelihood estimation

Given a dataset D = $\{(x_i, y_i)\}$ with m examples

$$
\begin{aligned}
h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h) P(y_i|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h) \\
&= \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)
\end{aligned}
$$

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

What next?

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p
- **Likelihood** for each feature given a label
  - $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p
- **Likelihood** for each feature given a label
  - $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
  - $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

- Prior: P(y = 1) = p and P (y = 0) = 1 – p

$$P(y_i|h) = p^{[y_i=1]}(1-p)^{[y_i=0]}$$

[z] is called the indicator function or the Iverson bracket

Its value is 1 if the argument z is true and zero otherwise

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

Likelihood for each feature given a label

- $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
- $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

$$P(x_{ij}|y_i, h) = a_j^{[y_i=1, x_{ij}=1]} \times (1 - a_j)^{[y_i=1, x_{ij}=0]} \times b_j^{[y_i=0, x_{ij}=1]} \times (1 - b_j)^{[y_i=0, x_{ij}=0]}$$

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}\,(y_i = 1)}{\text{Count}\,(y_i = 1) + \text{Count}\,(y_i = 0)}$$

← P(y = 1) = p

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}\,(y_i = 1)}{\text{Count}\,(y_i = 1) + \text{Count}\,(y_i = 0)}$$

← P(y = 1) = p

$$a_j = \frac{\text{Count}\,(y_i = 1, x_{ij} = 1)}{\text{Count}\,(y_i = 1)}$$

← $P(x_j = 1 \mid y = 1) = a_j$

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}\,(y_i = 1)}{\text{Count}\,(y_i = 1) + \text{Count}\,(y_i = 0)} \longleftarrow \text{P(y = 1) = p}$$

$$a_j = \frac{\text{Count}\,(y_i = 1, x_{ij} = 1)}{\text{Count}\,(y_i = 1)} \longleftarrow \text{P}(x_j = 1 \mid y = 1) = a_j$$

$$b_j = \frac{\text{Count}\,(y_i = 0, x_{ij} = 1)}{\text{Count}\,(y_i = 0)} \longleftarrow \text{P}(x_j = 1 \mid y = 0) = b_j$$

Let’s learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

P(Play = +) = 9/14 P(Play = -) = 5/14

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

P(Play = +) = 9/14 P(Play = -) = 5/14

P(**O** = S | Play = +) = 2/9

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

P(Play = +) = 9/14 P(Play = -) = 5/14

P(**O** = S | Play = +) = 2/9

P(**O** = R | Play = +) = 3/9

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

P(Play = +) = 9/14 P(Play = -) = 5/14

P(**O** = S | Play = +) = 2/9

P(**O** = R | Play = +) = 3/9

P(**O** = O | Play = +) = 4/9

And so on, for other attributes and also for Play = -

Naïve Bayes: Learning and Prediction

- Learning
  - Count how often features occur with each label. Normalize to get likelihoods
  - Priors from fraction of examples with each label
  - Generalizes to multiclass

- Prediction
  - Use learned probabilities to find highest scoring label

Important caveats with Naïve Bayes

1. Features need not be conditionally independent given the label
   - Just because we assume that they are doesn't mean that that's how they behave in nature
   - We made a modeling assumption because it makes computation and learning easier

2. Not enough training data to get good estimates of the probabilities from counts

Important caveats with Naïve Bayes

2. Not enough training data to get good estimates of the probabilities from counts

The basic operation for learning likelihoods is counting how often a feature occurs with a label.

What if we never see a particular feature with a particular label?

That will make the probabilities zero

Should we treat those counts as zero?

Answer: Smoothing

- Add fake counts (very small numbers so that the counts are not zero)

Example: Classifying text

- Instance space: Text documents
- Labels: Spam or NotSpam

- Goal: To learn a function that can predict whether a new document is Spam or NotSpam

How would you build a Naïve Bayes classifier?

Let us brainstorm

How to represent documents?
How to estimate probabilities?
How to classify?

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents

   1. Priors $P(\text{Spam}) = \frac{\text{Count}(\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

   2. For each word w in vocabulary :

$$P(\text{w}|\text{Spam}) = \frac{\text{Count}(\text{w}, \text{Spam}) + 1}{\text{Count}(\text{Spam}) + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count}(\text{w}, \text{NotSpam}) + 1}{\text{Count}(\text{NotSpam}) + |\text{Vocabulary}|}$$

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents

   1. Priors $P(\text{Spam}) = \frac{\text{Count (Spam)}}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

   2. For each word w in vocabulary :

$$P(\text{w}|\text{Spam}) = \frac{\text{Count (w, Spam)} + 1}{\text{Count (Spam)} + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count (w, NotSpam)} + 1}{\text{Count (NotSpam)} + |\text{Vocabulary}|}$$

How often does a word occur with a label?

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents

   1. Priors $P(\text{Spam}) = \dfrac{\text{Count}\,(\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

   2. For each word w in vocabulary :

Smoothing

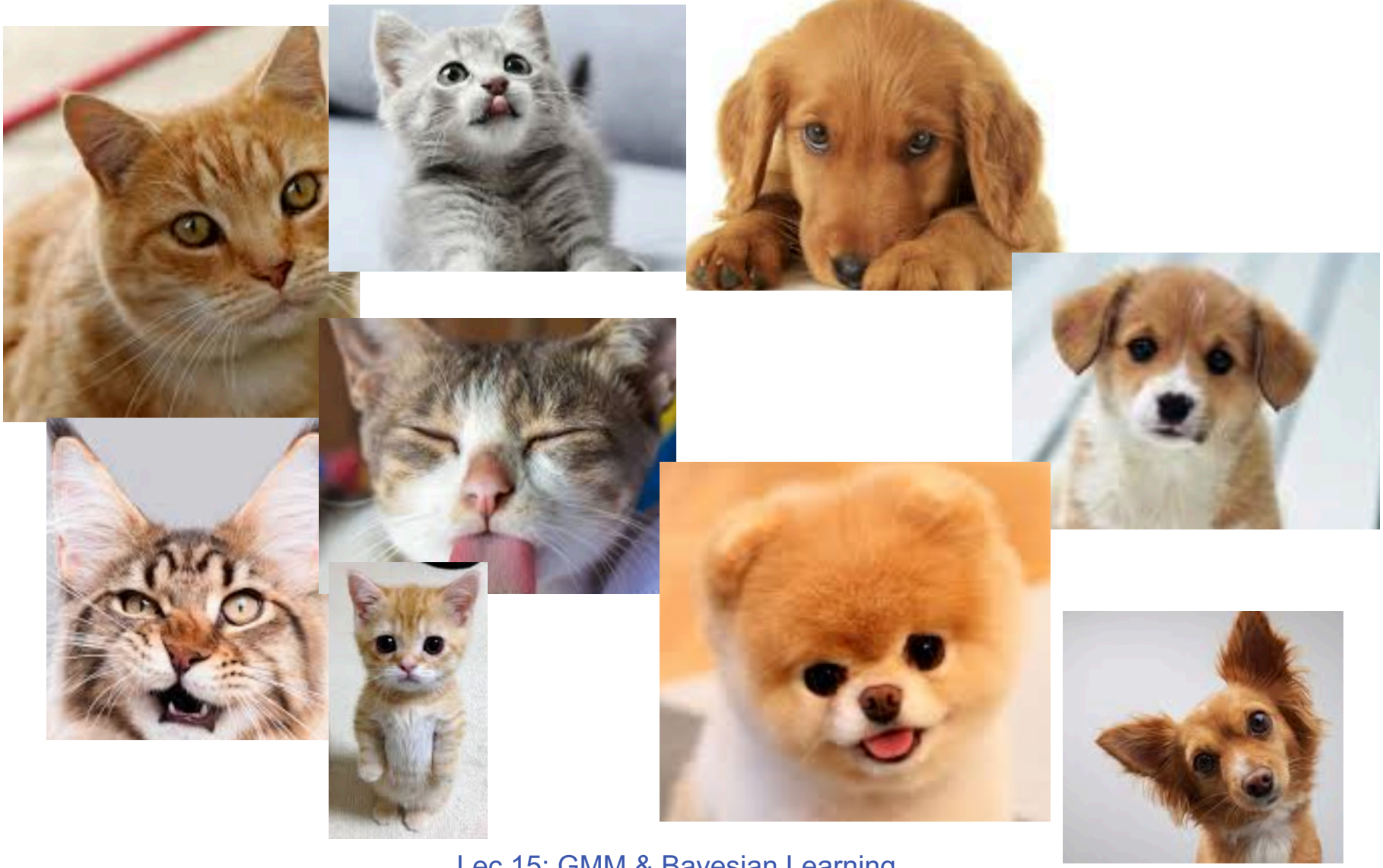
$$P(\text{w}|\text{Spam}) = \frac{\text{Count}\,(\text{w}, \text{Spam}) + 1}{\text{Count}\,(\text{Spam}) + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count}\,(\text{w}, \text{NotSpam}) + 1}{\text{Count}\,(\text{NotSpam}) + |\text{Vocabulary}|}$$

Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Generative model

How to classify cat and dog?



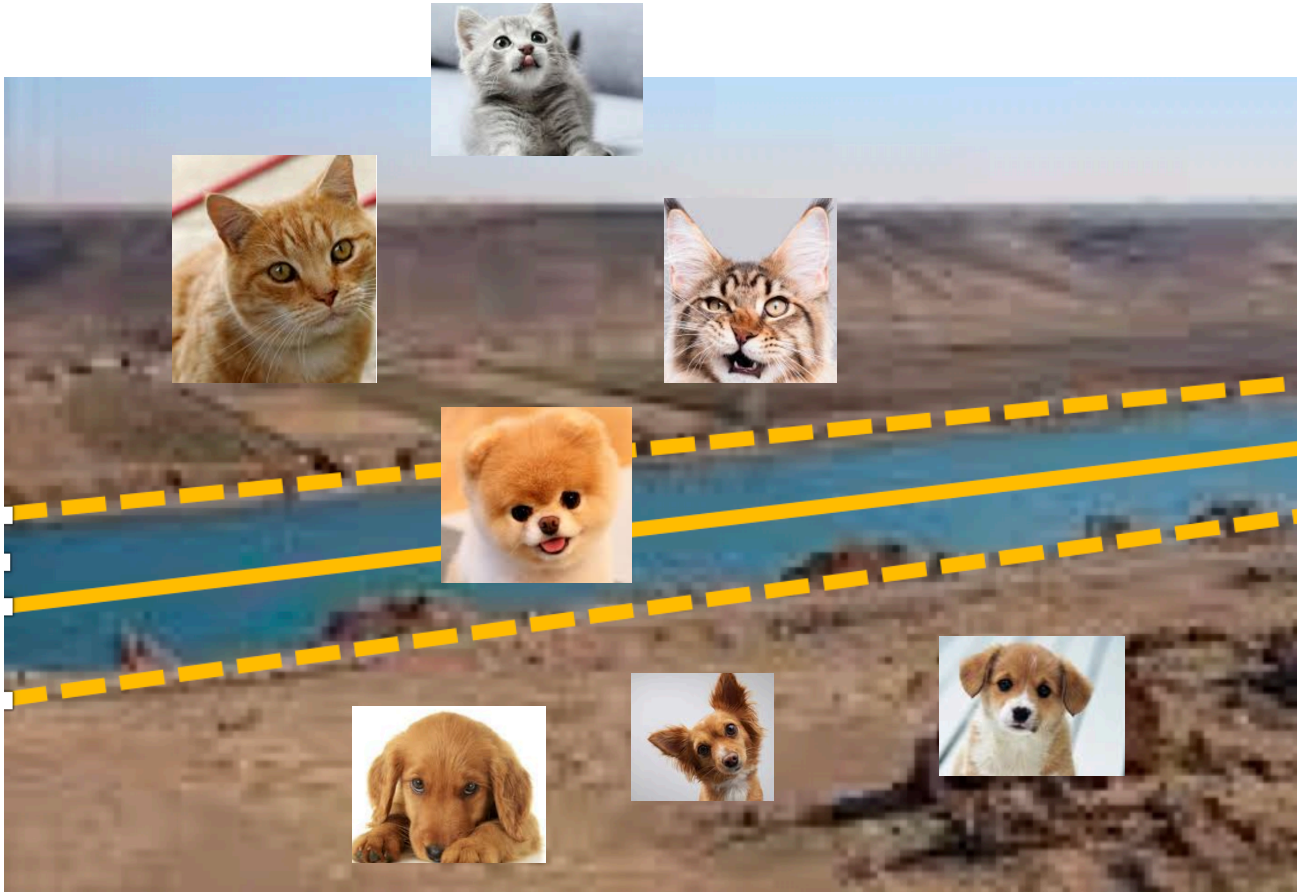
Discriminative models

Goal: learn directly how to make predictions

- Look at many (positive/negative) examples
- Discover regularities in the data
- Use these to construct a prediction policy
- Assumptions come in the form of the hypothesis class

Bottom line: approximating $h : X \rightarrow Y$ is estimating $P(Y|X)$

Discriminative model



margin (upper)

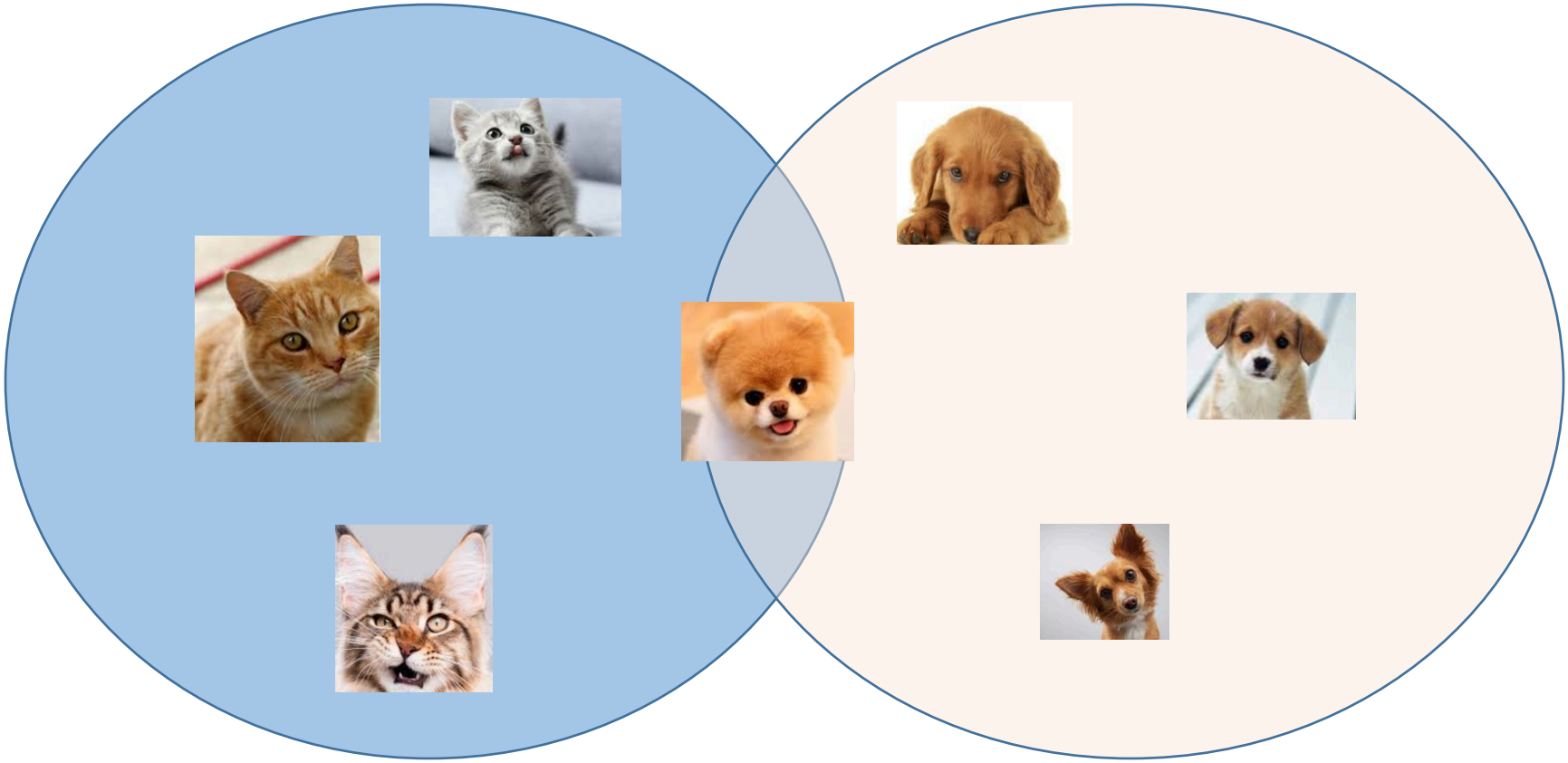
Decision boundary

margin (lower)

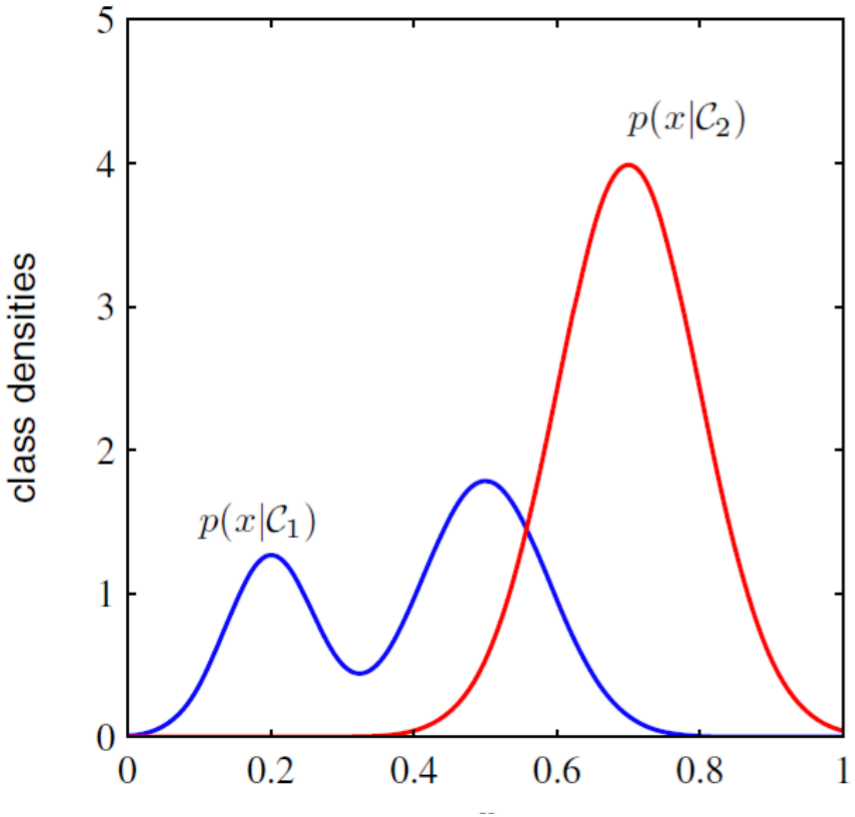
Generative models

- Explicitly model how instances in each category are generated
- That is, learn P(X | Y) and P(Y)
- We did this for naïve Bayes
  - Naïve Bayes is a generative model
- Predict P(Y | X) using the Bayes rule

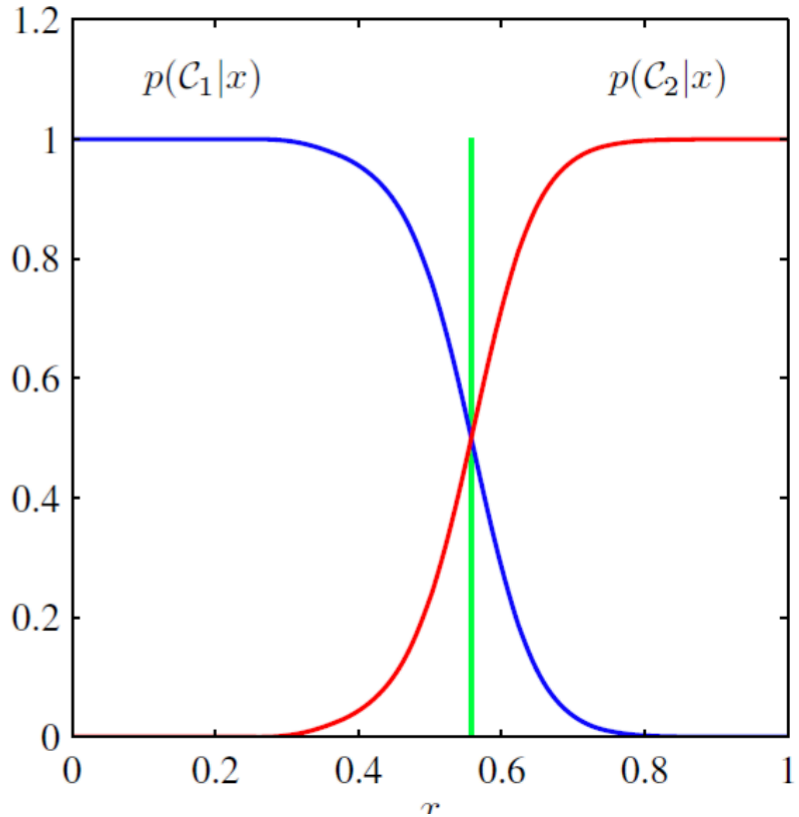
Generative model



Generative Model's view

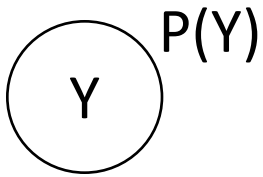


Discriminative Model's view

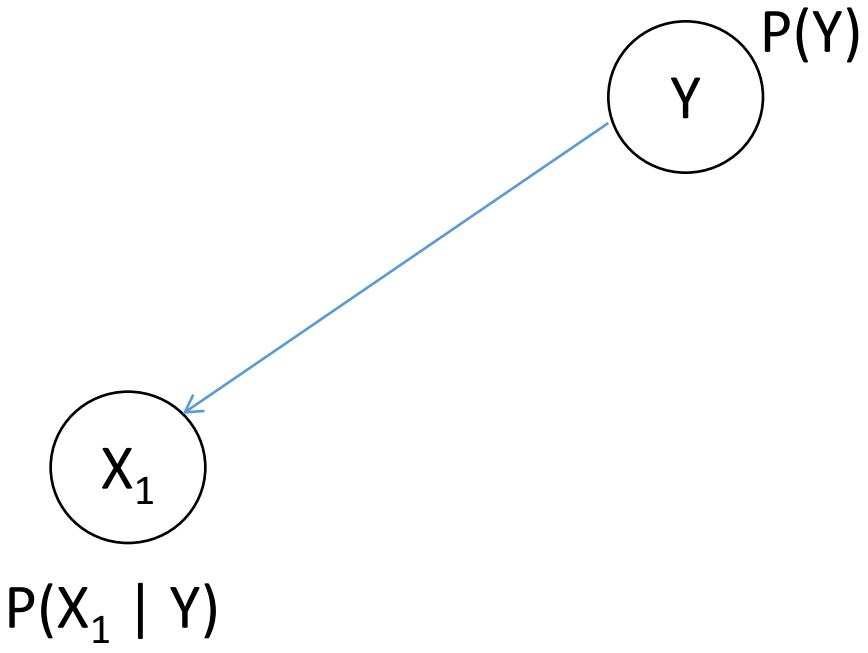


Example: Generative story of naïve Bayes

First sample a label

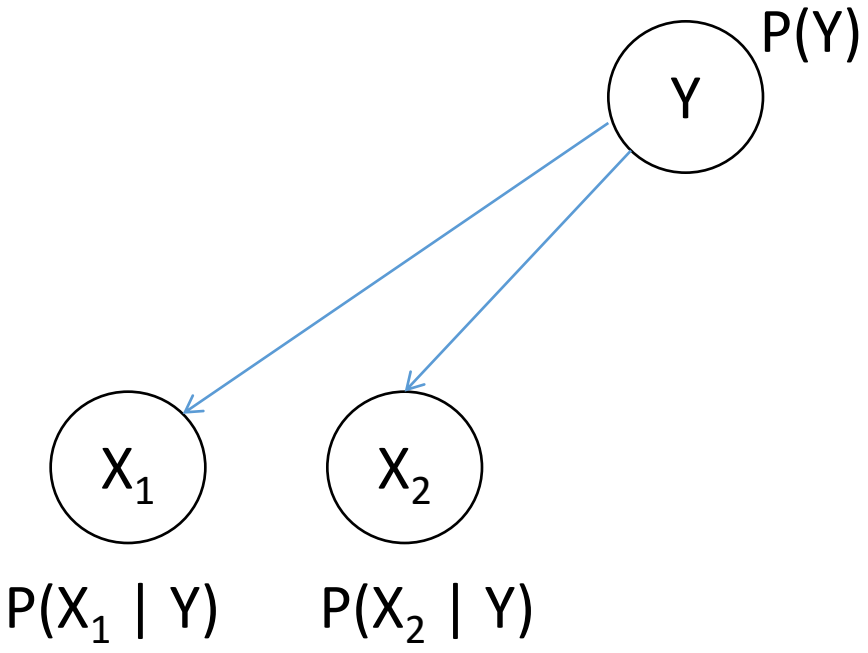


Example: Generative story of naïve Bayes



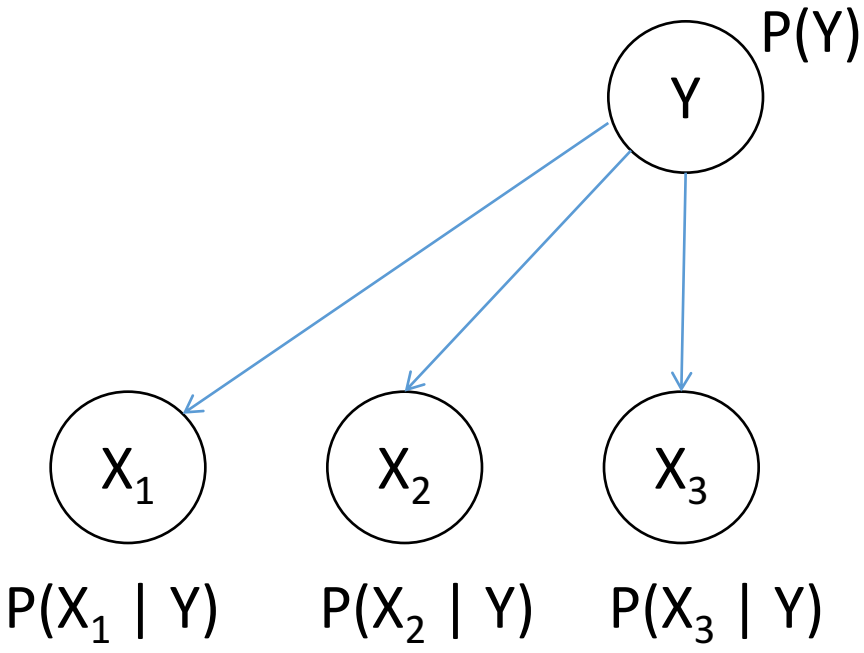
Given the label, sample the features independently from the conditional distributions

Example: Generative story of naïve Bayes



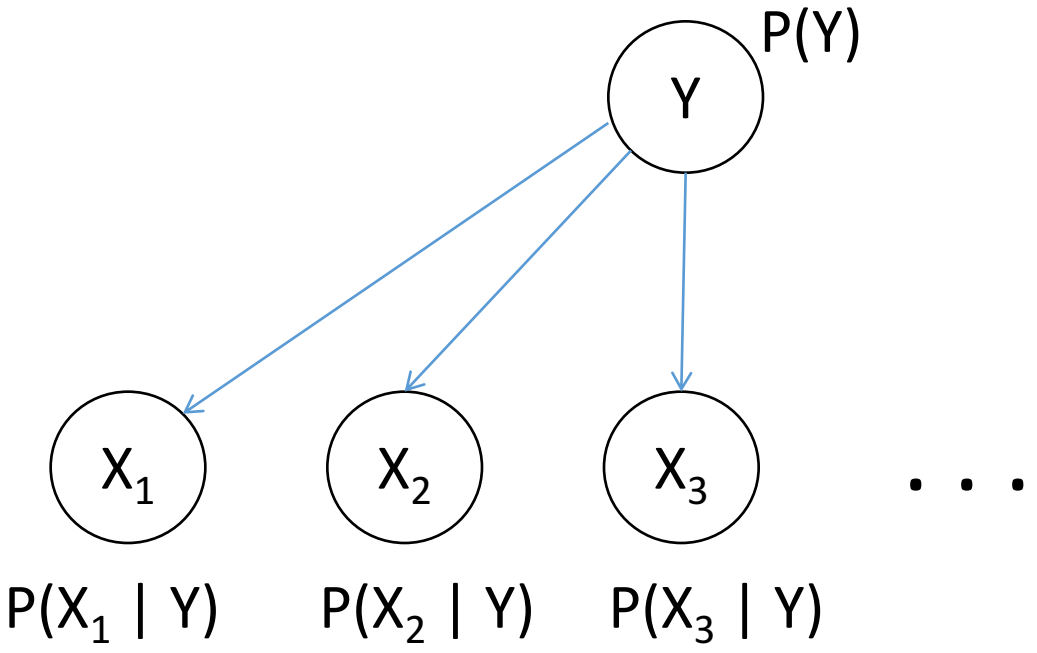
Given the label, sample the features independently from the conditional distributions

Example: Generative story of naïve Bayes



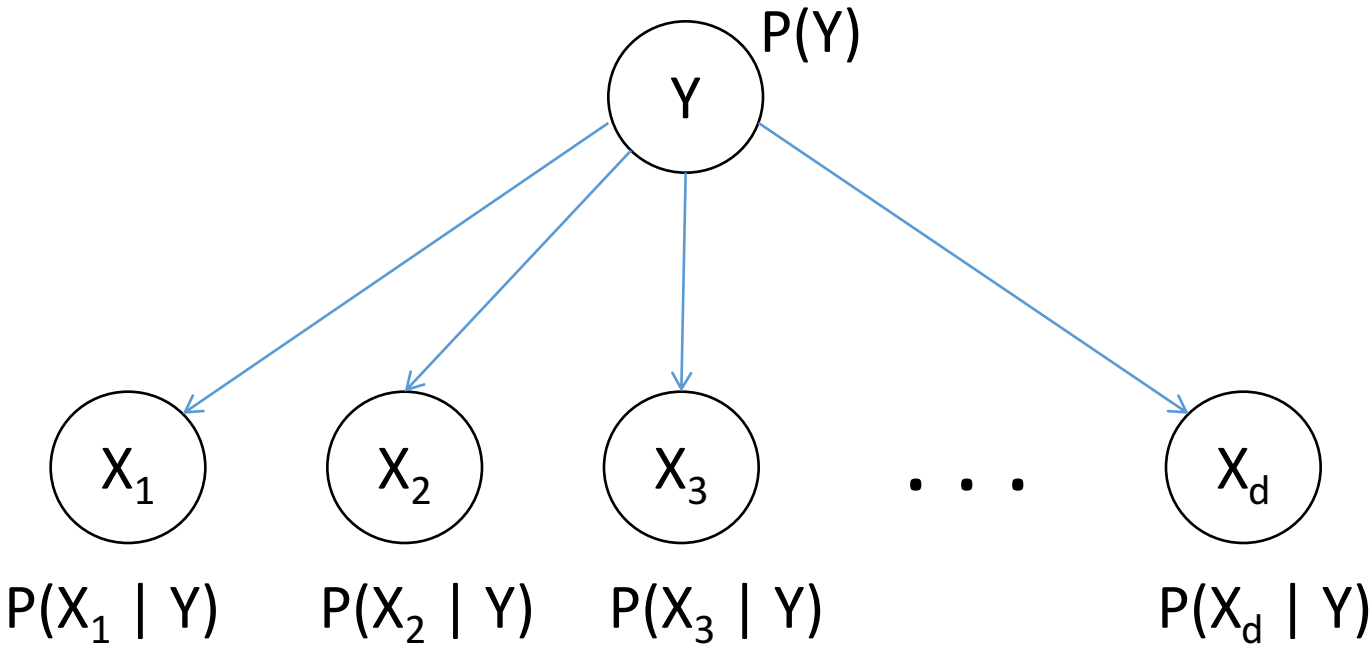
Given the label, sample the features independently from the conditional distributions

Example: Generative story of naïve Bayes



Given the label, sample the features independently from the conditional distributions

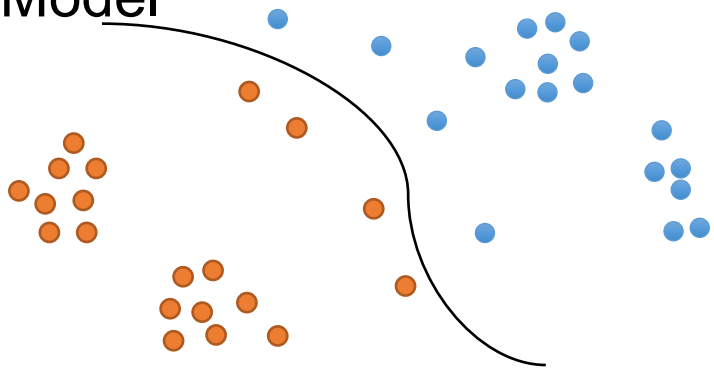
Generative vs Discriminative models



Given the label, sample the features independently from the conditional distributions

Generative vs Discriminative models

- Generative models
  - learn P(x, y)
  - Characterize how the data is generated (both inputs and outputs)
  - Eg: Naïve Bayes, Hidden Markov Model



- Discriminative models
  - learn P(y | x)
  - Directly characterizes the decision boundary only
  - Eg: Logistic Regression, Conditional models (several names)