Probabilistic Learning

Lecture 12 (a)

Machine Learning
Fall2015

What we have seen so far

What does it mean to learn?

- Mistake-driven learning
  - Learning by counting (and bounding) number of mistakes
- PAC learnability
  - Sample complexity and bounds on errors on unseen examples

Various learning algorithms

- Analyzed algorithms under these models of learnability
- In all cases, the algorithm outputs a function that produces a label y for a given input x

Coming up

Another way of thinking about "What does it mean to learn?"

- Bayesian learning

Different learning algorithms in this regime

- Naïve Bayes
- Logistic Regression

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative learning

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative 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: The basics

- Goal: To find the ***best*** hypothesis from some space H of hypotheses, using the observed data D
- Define best = most probable hypothesis in H
- In order to do that, we need to assume a probability distribution over the class H
- We also need to know something about the relation between the data observed and the hypotheses
  - As we will see, we can be Bayesian about other things. e.g., the parameters of the model

Bayesian methods have multiple roles

- Provide practical learning algorithms
- Combining prior knowledge with observed data
  - Bias the model towards something we know
- Provide a conceptual framework
  - For evaluating other learners
- Provide tools for analyzing learning

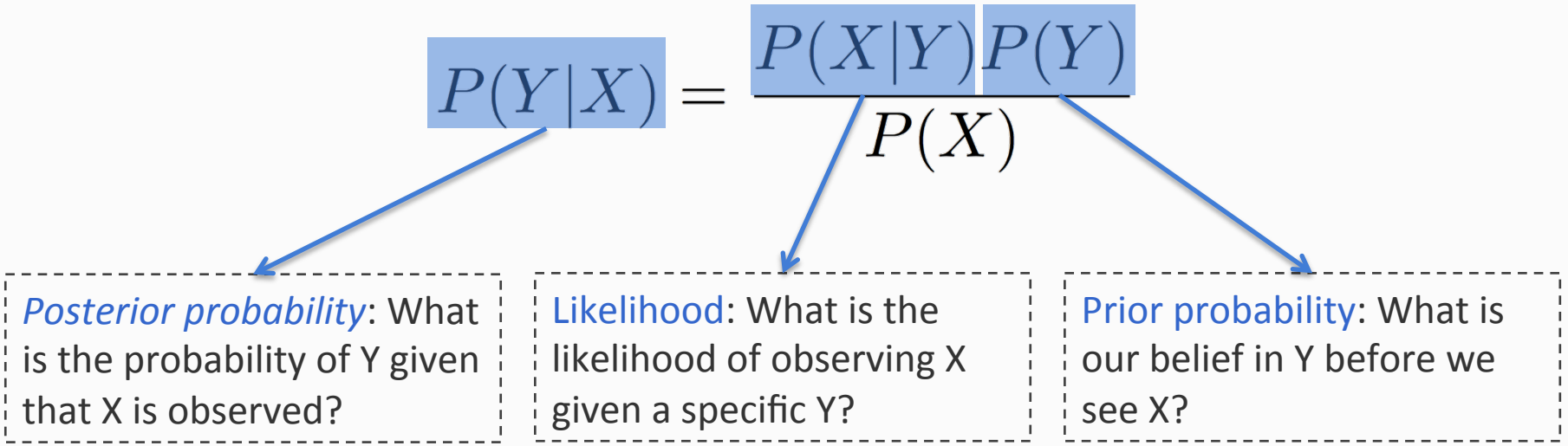
Bayes Theorem

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

Short for

$$\forall x_i, y_j, \quad P(Y = y_i|X = x_j) = \frac{P(X = x_j|Y = y_i)P(Y = y_i)}{P(X = x_j)}$$

Bayes Theorem



$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

Probability Refresher

- Product rule: $P(A \wedge B) = P(A, B) = P(A|B)P(B) = P(B|A)P(A)$
- Sum rule: $P(A \vee B) = P(A) + P(B) - P(A, B)$
- Events A, B are independent if:
  - P(A, B) = P(A) P(B)
  - Equivalently, P(A | B) = P(A), P(B | A) = P(B)
- Theorem of Total probability:

  For mutually exclusive events $A_1, A_2, \cdots, A_n$ (i.e $A_i \cap A_j = \emptyset$) with $\sum_i P(A_i) = 1$

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

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? In the absence of any information, the uniform distribution.

Choosing a hypothesis

Given some data, find the most probable hypothesis

- The Maximum a Posteriori hypothesis h_{MAP}

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

Choosing a hypothesis

Given some data, find the most probable hypothesis

– The Maximum a Posteriori hypothesis h_{MAP}

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} P(h|D) \\
&= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
&= \arg\max_{h \in H} P(D|h)P(h)
\end{aligned}
$$

Posterior $\propto$ Likelihood $\times$ Prior

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*

Brute force MAP learner

Input: Data D and a hypothesis set H

1. Calculate the posterior probability for each h $\in$ H

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

Difficult to compute, except for the most simple hypothesis spaces

2. Output the hypothesis with the highest posterior probability

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

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative learning

Maximum Likelihood estimation

Maximum Likelihood estimation (MLE)

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

What we need in order to define learning:

1. A hypothesis space H
2. A model that says how data D is generated given h

Example 1: Binomial distributions

The CEO of a startup hires you for your first consulting job

- *CEO*: My company makes glass beads. I need to know what is the probability they will shatter when I drop them from 20 feet?
- *You*: Sure. I can help you out. Are they all identical?
- *CEO*: Yes!
- *You*: Excellent. I know how to help. We need to experiment…

Breaking bead

The experiment:

Drop 5 beads from twenty feet

3 break, 2 don't

You: The probability is P(failure) = 0.6

CEO: But why?

You: Because…

The binomial distribution

- P(failure) = p, P(success) = 1 – p
- Each trial is i.i.d
  - Independent and identically distributed
- You have seen D = {*Break, Safe, Break, Break, Safe*}
  $P(data \mid p) = p^3 (1-p)^2$
- The most likely value of p for this observation is?
  $\max_p P(data \mid p) = \max_p p^3 (1-p)^2$

The “learning” algorithm

Say you have *a* Safes and b Breaks

$$P_{best} = \text{argmax}_p \log P(D \mid p)$$

Log likelihood

$$= \text{argmax}_p \log [p^b (1\text{-}p)^a]$$

Calculus 101: Set the derivative to zero

$$P_{best} = b/(a + b)$$

The model we assumed is binomial. *You could assume a different model!* Next we will consider other models and see how to learn their parameters.

Example 2:

Maximum Likelihood and least squares

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

Suppose H consists of real valued functions

Inputs are vectors $\mathbf{x} \in \Re^d$ and the output is a real number $y \in \Re$

Suppose the training data is generated as follows:

- An input x_i is drawn randomly (say uniformly at random)
- The true function f is applied to get $f(\mathbf{x}_i)$
- This value is then perturbed by noise e_i
  - Drawn independently according to an unknown Gaussian with zero mean

$$y_i = f(x_i) + e_i$$

Say we have m training examples (x_i, y_i) generated by this process

Example:

Maximum Likelihood and least squares

Recall that a normal distribution is parameterized by its mean μ and variance σ

If h were the true function, the mean of y_i is $h(x_i)$

So we have the probability density

$$p(y_i|h, x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}}$$

Probability of observing one data point (x_i, y_i), if it were generated using the function h

Example:

Maximum Likelihood and least squares

Probability of observing one data point (x_i, y_i), if it were generated using the function h

$$p(y_i|h, x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}}$$

Each example in our dataset D = $\{(x_i, y_i)\}$ is generated independently by this process

$$p(D|h) = \prod_{i=1}^{m} p(y_i, x_i|h) \propto \prod_{i=1}^{m} p(y_i|h, x_i)$$

Example:

Maximum Likelihood and least squares

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

Our goal is to find the most likely hypothesis

$$h_{ML} = \arg\max_{h \in H} p(D|h) = \arg\max_{h \in H} \prod_{i=1}^{m} p(y_i|h, x_i)$$

Example:

Maximum Likelihood and least squares

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

Our goal is to find the most likely hypothesis

$$h_{ML} = \arg\max_{h \in H} p(D|h) = \arg\max_{h \in H} \prod_{i=1}^{m} p(y_i|h, x_i)$$

$$= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}}$$

How do we maximize this expression? Any ideas?

Answer: Take the logarithm to simplify

Example:

Maximum Likelihood and least squares

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

Our goal is to find the most likely hypothesis

$$\begin{aligned} h_{ML} &= \arg\max_{h \in H} p(D|h) = \arg\max_{h \in H} \prod_{i=1}^{m} p(y_i|h, x_i) \\ &= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}} \\ &= \arg\max_{h \in H} \sum_{i=1}^{m} \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2} \end{aligned}$$

Example:

Maximum Likelihood and least squares

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

Our goal is to find the most likely hypothesis

$$\begin{aligned}
h_{ML} &= \arg\max_{h \in H} p(D|h) = \arg\max_{h \in H} \prod_{i=1}^{m} p(y_i|h, x_i) \\
&= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}} \\
&= \arg\max_{h \in H} \sum_{i=1}^{m} \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2} \\
&= \arg\max_{h \in H} - \sum_{i=1}^{m} \frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}
\end{aligned}$$

Example:

Maximum Likelihood and least squares

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

Our goal is to find the most likely hypothesis

$$\begin{aligned}
h_{ML} &= \arg\max_{h \in H} p(D|h) = \arg\max_{h \in H} \prod_{i=1}^{m} p(y_i|h, x_i) \\
&= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2}} \\
&= \arg\max_{h \in H} \sum_{i=1}^{m} \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2} \\
&= \arg\max_{h \in H} -\sum_{i=1}^{m} \frac{(y_i - h(\mathbf{x}_i))^2}{2\sigma^2} \\
&= \arg\min_{h \in H} \sum_{i=1}^{m} (y_i - h(\mathbf{x}_i))^2
\end{aligned}$$

Example:

Maximum Likelihood and least squares

The most likely hypothesis is

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (y_i - h(\mathbf{x}_i))^2$$

If we consider the set of linear functions as our hypothesis space: $h(x_i) = \mathbf{w}^T \mathbf{x}_i$

$$h_{ML} = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

This is the probabilistic explanation for least squares regression

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative learning

Most probable classification

- So far: What is the most probable hypothesis given data?
- We can also ask: For a new test point, what is the most probable label, given training data?
- Is this the same as the prediction of the maximum a posteriori hypothesis?

Bayes Optimal Classifier

- How should we use the general formalism?
  - What should H be?
- H can be a collection of functions.
  - Given the training data, choose an optimal function
  - Then, given new data, evaluate the selected function on it
- H can be a collection of possible predictions
  - Given the data, try to directly choose the optimal prediction
- These two could be different!

Selecting a function vs. entertaining all options until the last minute

Most probable classification

Suppose our hypothesis space H has three functions h_1, h_2 and h_3

- $P(h_1 \mid D) = 0.4$, $P(h_2 \mid D) = 0.3$, $P(h_3 \mid D) = 0.3$
- What is the MAP hypothesis? h_1
- For a new instance **x**, suppose $h_1(\mathbf{x}) = +1$, $h_2(\mathbf{x}) = -1$ and $h_3(\mathbf{x}) = -1$
- What is the most probable classification of **x**? -1

$P(+1 \mid \mathbf{x}) = 0.4$ $\quad$ $P(-1 \mid \mathbf{x}) = 0.3 + 0.3$

The most probable classification is not the same as the prediction of the MAP hypothesis

Bayes Optimal Classification

Defined as the label produced by the most probable classifier

$$\arg\max_{y} \sum_{h_i \in H} P(y|h_i)P(h_i|D)$$

Computing this can be hopelessly inefficient

And yet an interesting theoretical concept because, no other classification method can outperform this method on average

(using the same hypothesis space and prior knowledge)

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative learning

Predicting with probabilities

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 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

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

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

Don't confuse with *MAP learning*: finds hypothesis by

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

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 again

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

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?

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 × 0.3
= 0.12

P(H, W | No) P(No) = 0.1 × 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	-

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: $P(x_1, x_2, x_3, x_4 \mid \text{Play?})$
- $(2^4 - 1)$ 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?)

Likelihood P(**X** | Y)

- If there are d features, then:
- 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 features, then:
- 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

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

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

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

The Naïve Bayes Assumption

That is, $P(x_1, x_2, \cdots, x_d \mid y) = P(x_1 \mid y)\, P(x_2 \mid y) \ldots P(x_d \mid y)$

Requires only d numbers for each label. kd features 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_i, we have the likelihood $P(X_i \mid y)$

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_i, we have the likelihood $P(X_i \mid y)$

Decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y)P(x_1, x_2, \cdots x_d|y)$$

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_i, we have the likelihood $P(X_i \mid y)$

Decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y)P(x_1, x_2, \cdots x_d|y)$$

$$= \arg\max_y P(y)\prod_{i=1}^{d} P(x_i|y)$$

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

And we want to learn the classifier in a probabilistic way

- What is the 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

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

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

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

Each example in the dataset is independent and identically distributed

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

Maximum likelihood estimation

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

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

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

Each example in the dataset is independent and identically distributed

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

Asks "What probability would this particular h assign to the pair ($\mathbf{x}_i$, y_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

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

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

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

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?

We need to make a modeling assumption about these probability distributions

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

Learning the naïve Bayes Classifier

Maximum likelihood estimation

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

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

h consists of p, all the a's and b's

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

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

← P(y = 1) = p

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

← $P(x_j = 1 \mid y = 1) = a_j$

$$b_j = \frac{\text{Count}\,(y_i = 0, x_{ij} = 1)}{\text{Count}\,(y_i = 0)}$$

← $P(x_j = 1 \mid y = 0) = b_j$

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	-

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

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

Important caveats with Naïve Bayes

1. Features are not conditionally independent

All bets are off if the naïve Bayes assumption is not satisfied

$$P(\mathbf{x}|y) \neq \prod P(x_j|y)$$

And yet, very often used in practice because of simplicity

Words reasonably well even when the assumption is violated

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?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero?

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?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero? But that will make the probabilities zero

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?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero? But that will make the probabilities zero

Answer: Smoothing

- Add fake counts (very small numbers so that the counts are not zero)
- The Bayesian interpretation of smoothing: Priors on the hypothesis (MAP learning)

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?

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_i, we have the likelihood $P(X_i \mid y)$

Decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y)P(x_1, x_2, \cdots x_d|y)$$

$$= \arg\max_y P(y)\prod_{i=1}^{d} P(x_i|y)$$

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

   2. For each word w in vocabulary :

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

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

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

   2. For each word w in vocabulary :

Smoothing

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

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

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

$$P(y=+)\prod_j P(x_{ij}|y=+) > P(y=-)\prod_j P(x_{ij}|y=-)$$

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

$$P(y=+)\prod_j P(x_{ij}|y=+) > P(y=-)\prod_j P(x_{ij}|y=-)$$

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

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Taking log and simplifying, we get

$$\log \frac{P(y=0|\mathbf{x})}{P(y=1|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + b$$

This is a linear function of the feature space!

Easy to prove. See note on course website

Continuous features

- So far, we have been looking at discrete features
  - $P(x_j \mid y)$ is a Bernoulli trial (i.e. a coin toss)

- We could model $P(x_j \mid y)$ with other distributions too
  - This is a separate assumption from the independence assumption that naive Bayes makes
  - Eg: For real valued features, $(X_j \mid Y)$ could be drawn from a normal distribution

- **Exercise**: Derive the maximum likelihood estimate when the features are assumed to be drawn from the normal distribution

Summary: Naïve Bayes

- Independence assumption
  - All features are independent of each other given the label
- Maximum likelihood learning: Learning is simple
  - Generalizes to real valued features
- Prediction via MAP estimation
  - Generalizes to beyond binary classification
- Important caveats to remember
  - Smoothing
  - Independence assumption may not be valid
- Decision boundary is linear for binary classification

Today's lecture

- Bayesian Learning
  - Maximum a posteriori and maximum likelihood estimation
- Examples
  - Binomial distribution
  - Normal distribution
- Bayes Optimal Classifier
- Naïve Bayes Classification
- Discriminative and Generative learning

What we saw most of the semester

- A fixed, unknown distribution D over $X \times Y$
  - X: Instance space, Y: label space (eg: {+1, -1})
- Given a dataset $S = \{(x_i, y_i)\}$
- Learning
  - Identify a hypothesis space H, define a loss function L(h, x, y)
  - Minimize average loss over training data (plus regularization)
- The guarantee
  - If we find an algorithm that minimizes loss on the observed data
  - Then, learning theory guarantees good future behavior (as a function of H)

Is this different from assuming a distribution over X and a fixed oracle function f?

Discriminative models

Goal: learn directly how to make predictions

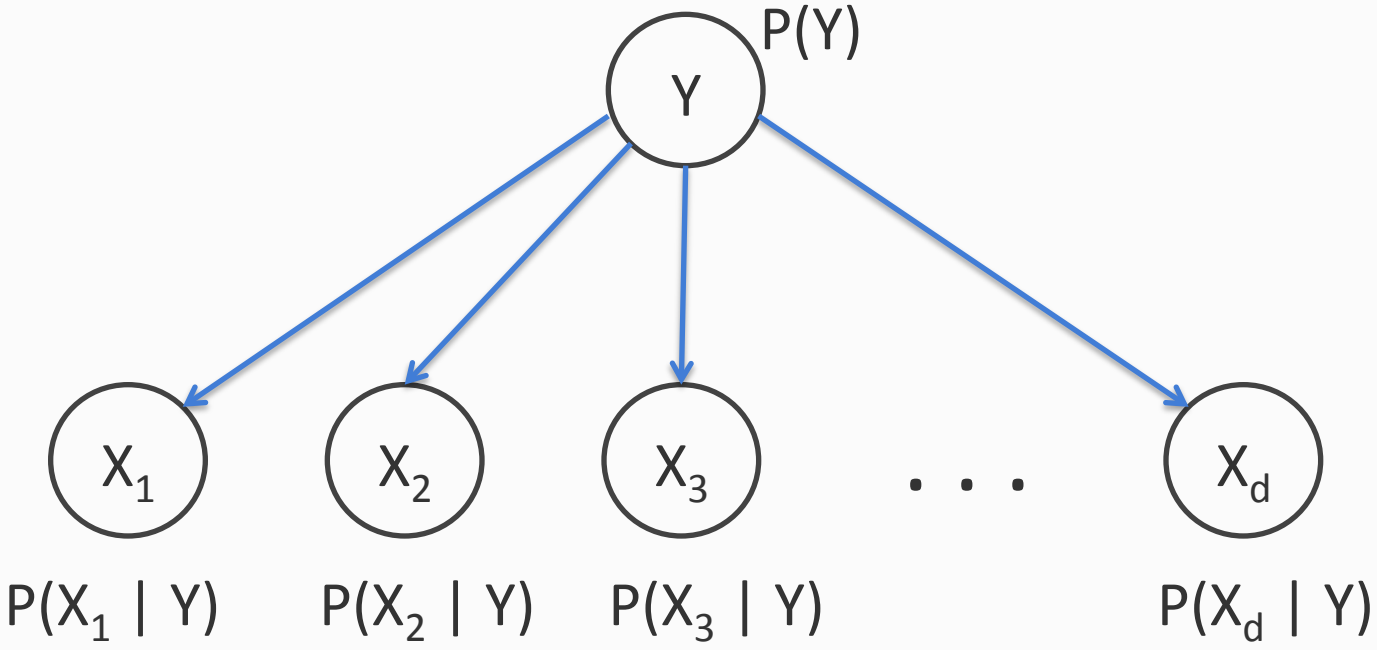
- Look at many (positive/negative) examples
- Discover regularities in the data
- Use these to construct a prediction policy
- Assumptions come in the form of the hypothesis class

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

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

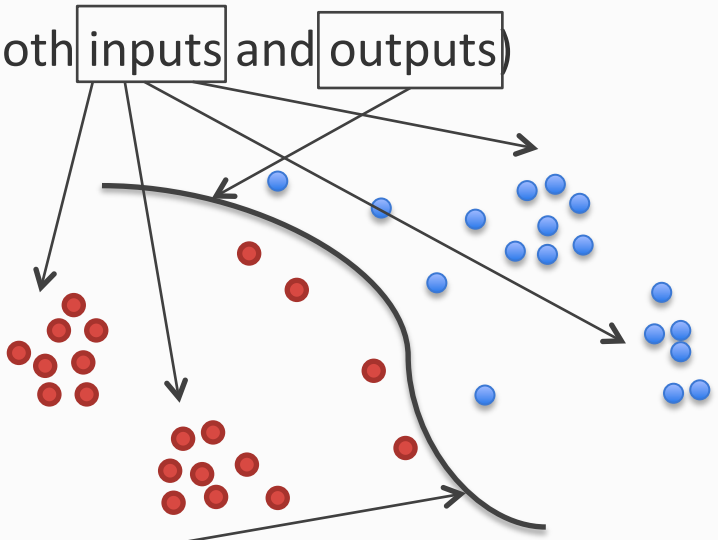
Example: Generative story of naïve Bayes



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

A generative model tries to characterize the distribution of the inputs, a discriminative model doesn't care



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

Coming up next

- More on discriminative vs generative:
  - Naive Bayes is a generative model, while logistic regression is discriminative
- MAP estimation for learning logistic regression
- Logistic regression as loss minimization