Lecture 16: Naïve Bayes, Generative Model, EM

Winter 2018

Kai-Wei Chang

CS @ UCLA

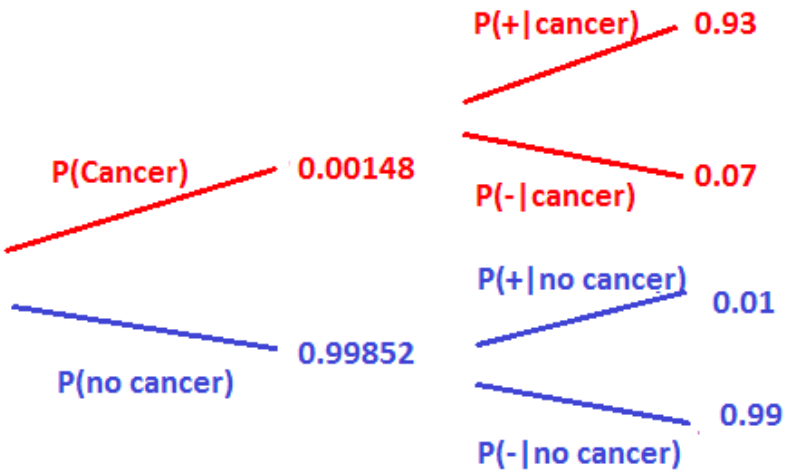
kw+cm146@kwchang.net

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.

Recap: Bayes Theorem Example

- How likely the patient got cancer if the test is positive?

$$P(\text{CANCER}|+) = \frac{P(cancer\ and +)}{P(cancer\ and +) + P(no\ cancer\ and +)} = 0.12$$



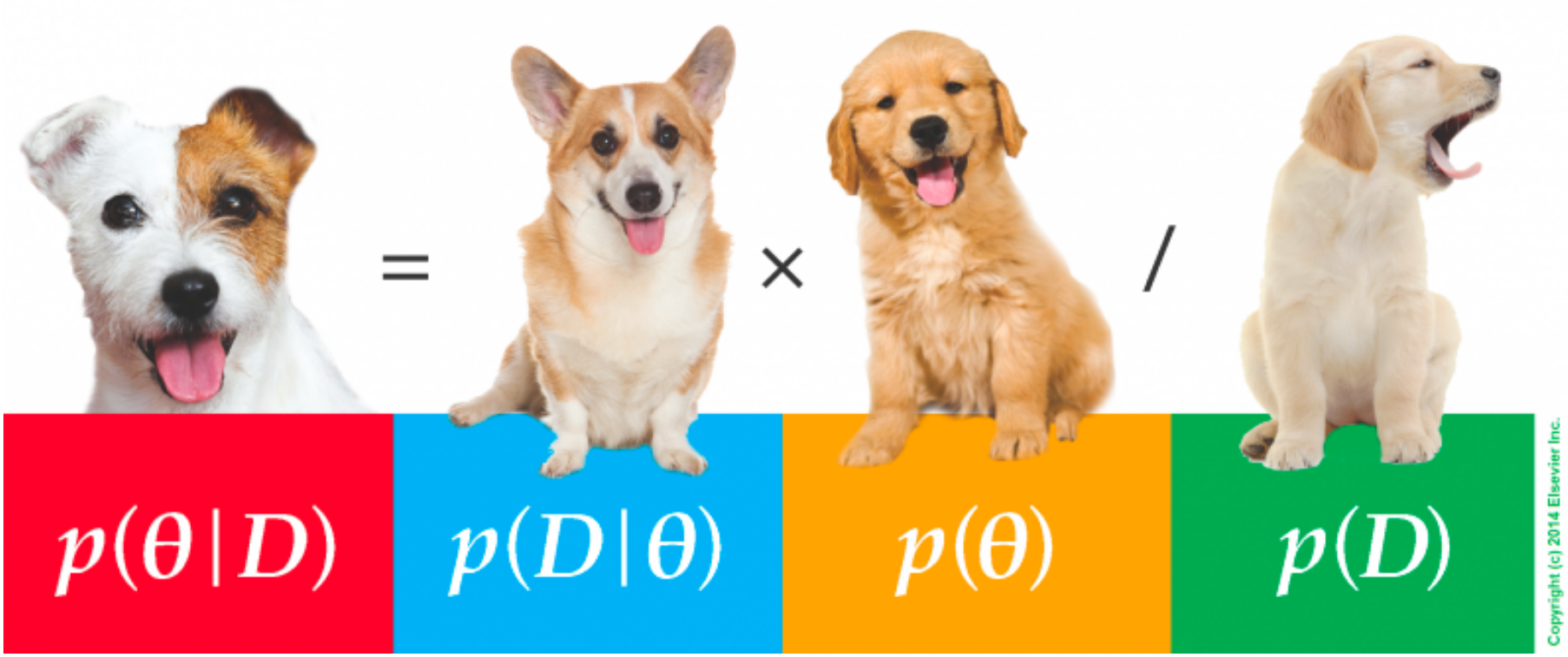
Recap: Bayes Theorem

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

Short for

$$\forall x, y \quad P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

Probabilistic models and Bayesian Learning



Probabilistic Learning

Two different notions of probabilistic learning

- Learning probabilistic concepts (P(Y|X))
  - The learned concept is a function c:X→[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)

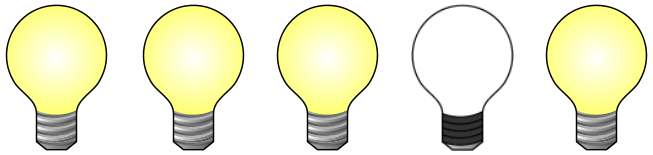
- Bayesian Learning: Use of a probabilistic criterion in selecting a hypothesis (P($\Theta|D$))
  - The hypothesis can be deterministic, a Boolean function
  - The criterion for selecting the hypothesis is probabilistic

Faulty lightbulbs

The experiment:

Try out 100 lightbulbs

80 work, 20 don't



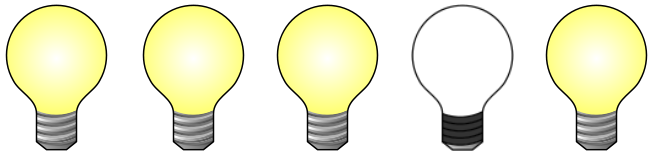
You: The probability is P(failure) = 0.2

CEO: But how do you know?

You: Because…

Bernoulli trials

- P(failure) = p, P(success) = 1 – p



- Each trial is i.i.d
  - Independent and identically distributed

MAP v.s. MLE

- MLE:

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

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

- MAP

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

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

Today's lecture

- Naïve Bayes
- Generative Models
- EM

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

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?

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

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?

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

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 = -

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

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

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

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

← P(y = 1) = p

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

← $P(x_j = 1 \mid y = 1) = a_j$

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

← $P(x_j = 1 \mid y = 0) = b_j$

Naïve Bayes: Learning and Prediction

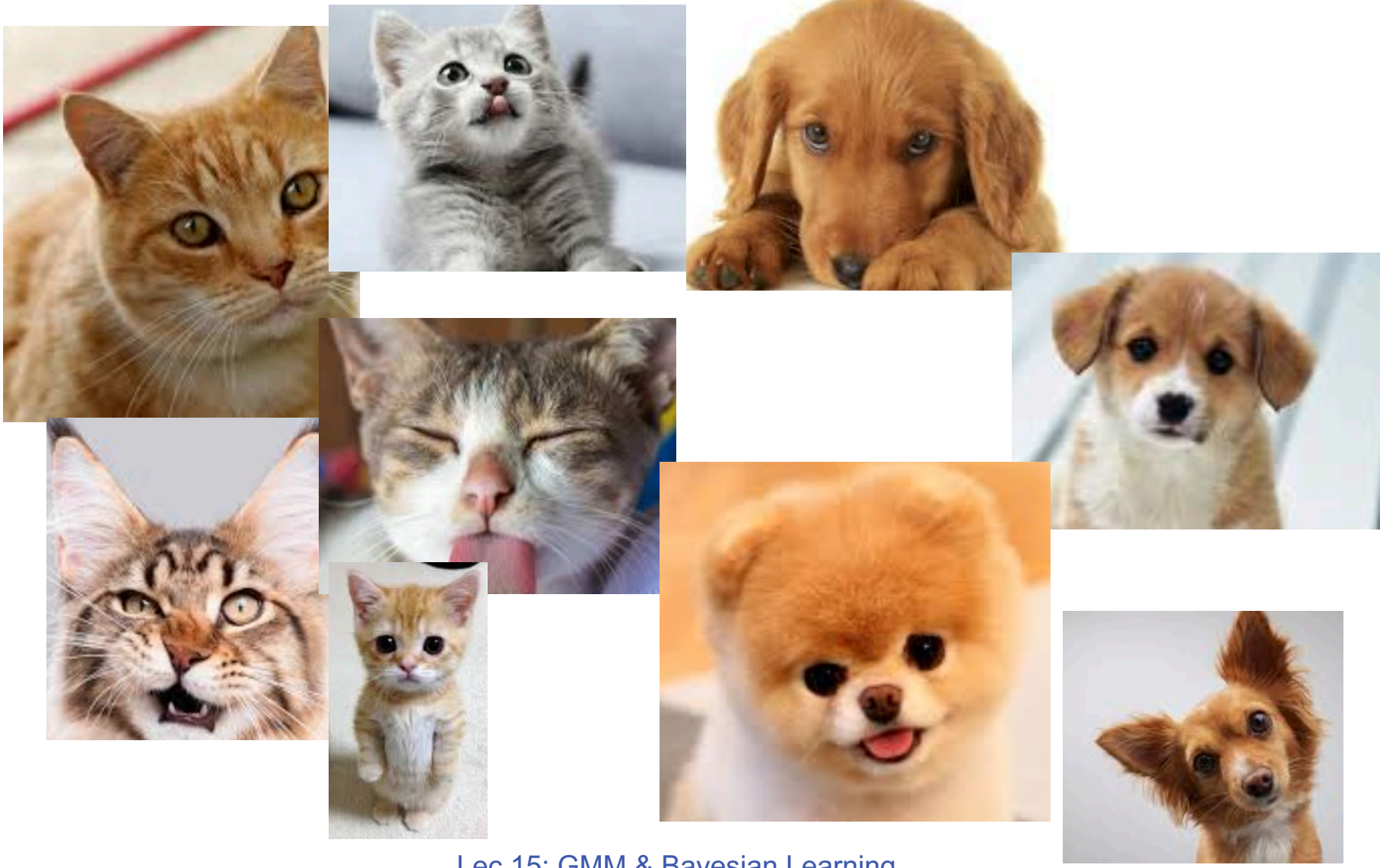
- Learning
  - Count how often features occur with each label. Normalize to get likelihoods
  - Priors from fraction of examples with each label
  - Generalizes to multiclass

- Prediction
  - Use learned probabilities to find highest scoring label

Today's lecture

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

How to classify cat and dog?



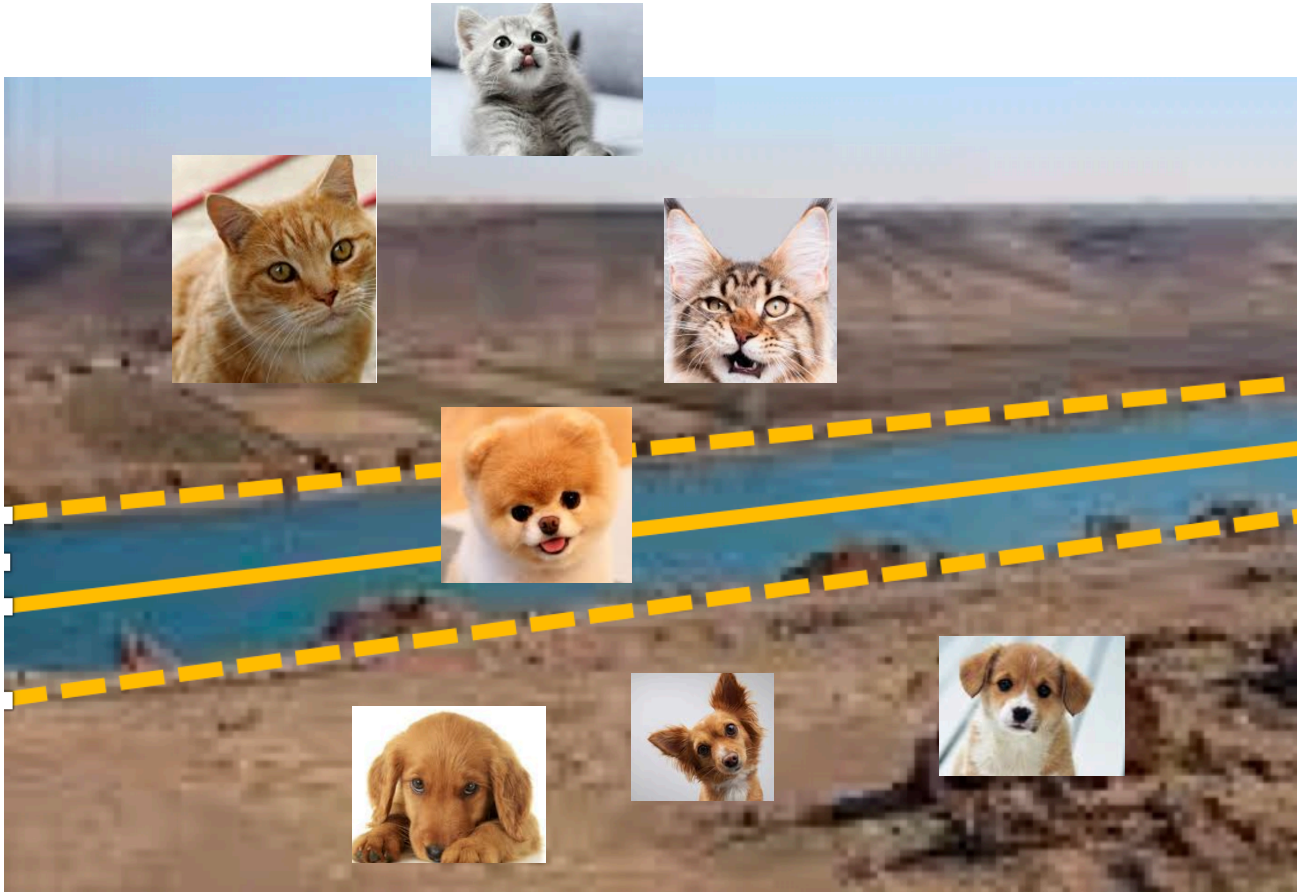
Discriminative models

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)

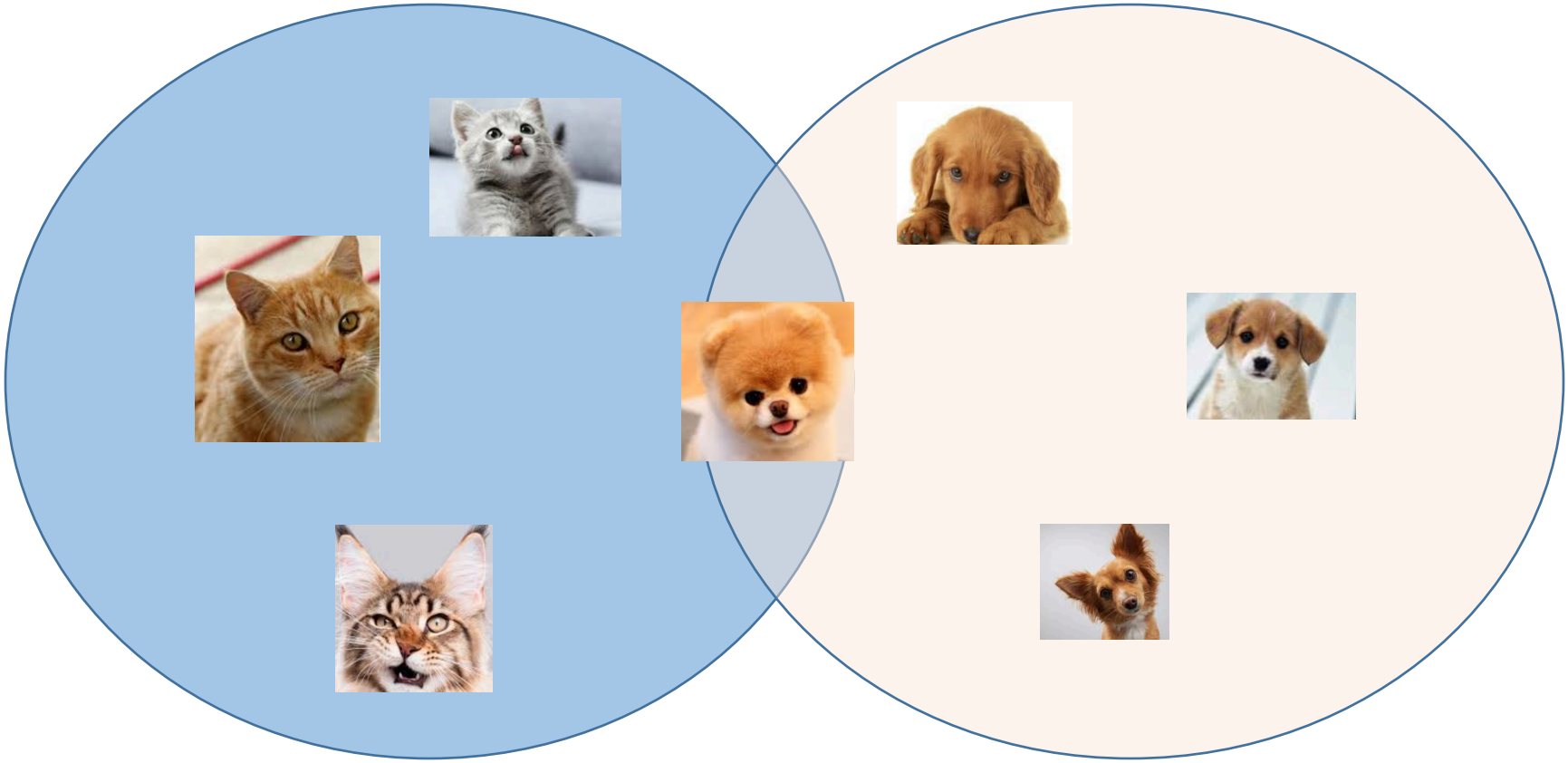
Decision boundary

margin (lower)

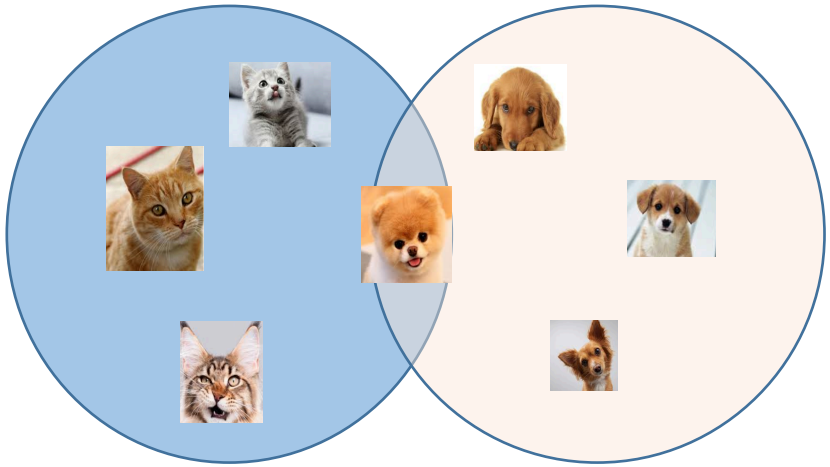
Generative models

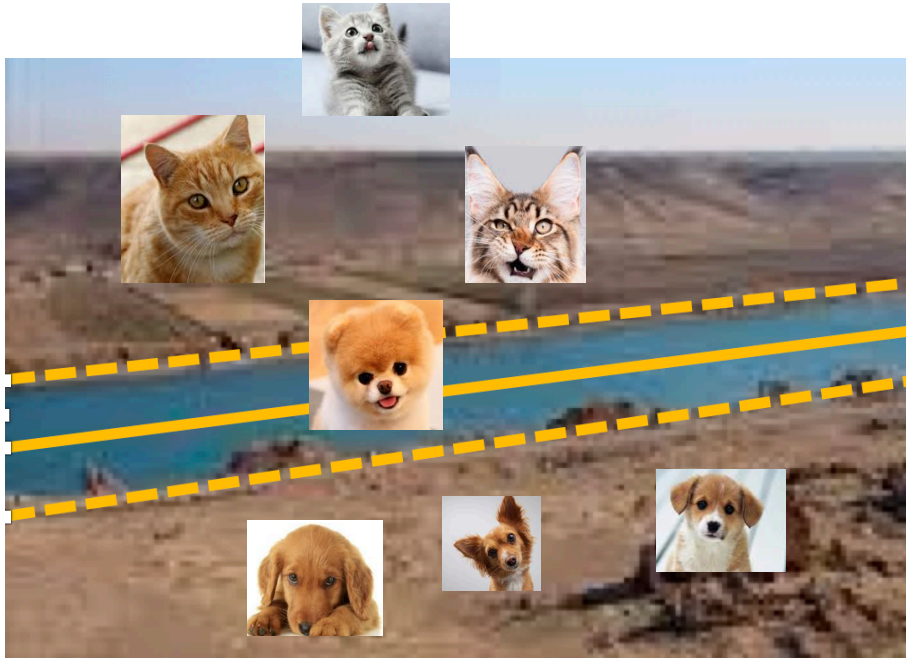
- Explicitly model how instances in each category are generated
- That is, learn P(X | Y) and P(Y)
- We did this for naïve Bayes
  - Naïve Bayes is a generative model
- Predict P(Y | X) using the Bayes rule

Generative model



Generative Model v.s. Discriminative model

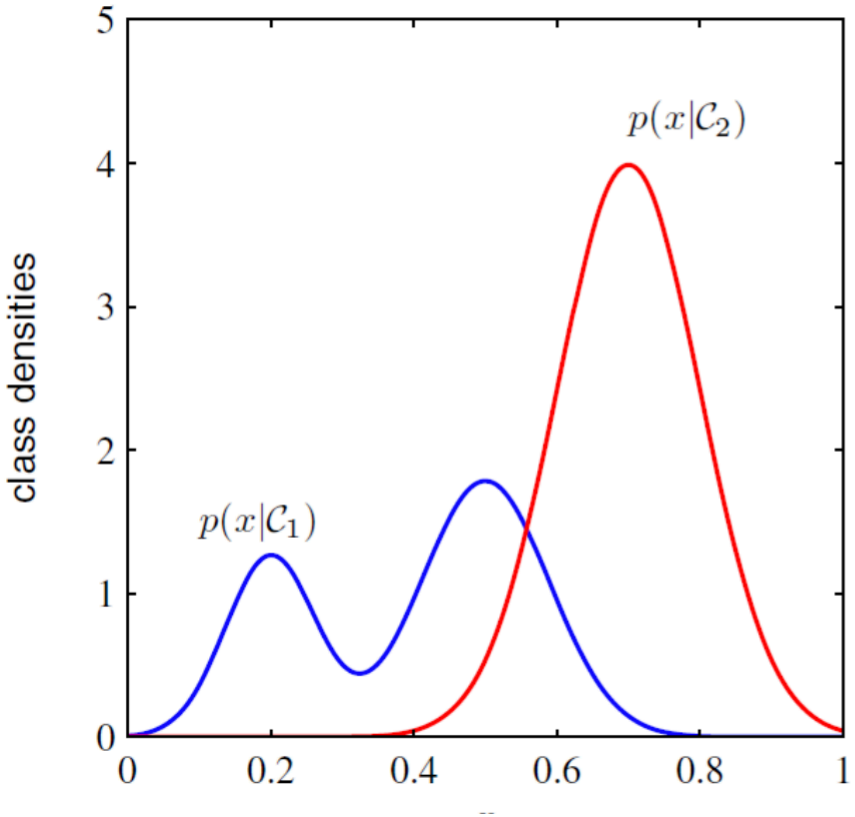




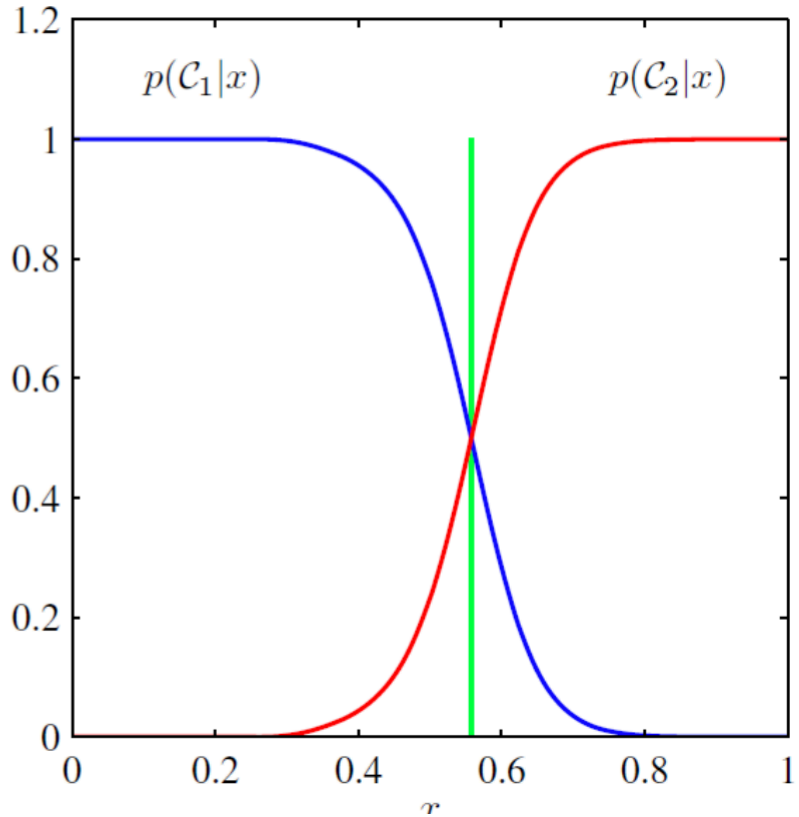
Learn $P(X, Y \mid \Theta)$ or
Learn $P(X \mid Y, \Theta)$ and $P(Y \mid \Theta)$

Learn $P(Y \mid X, \Theta)$

Generative Model's view

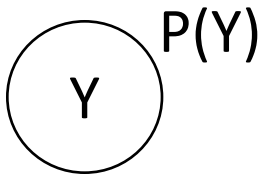


Discriminative Model's view

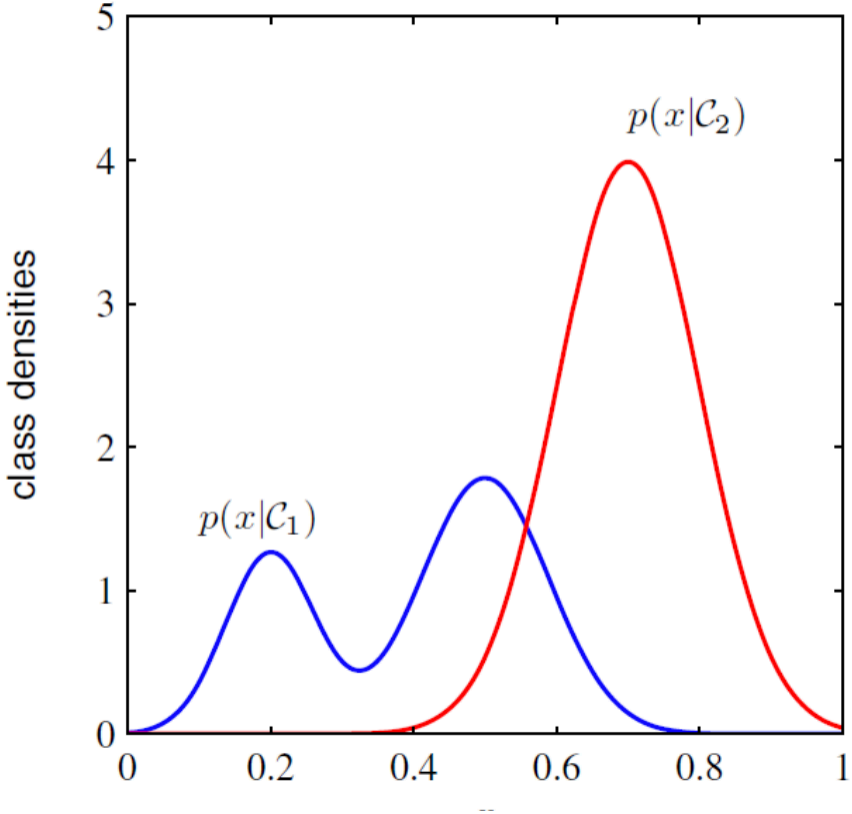


Example: Generative story of naïve Bayes

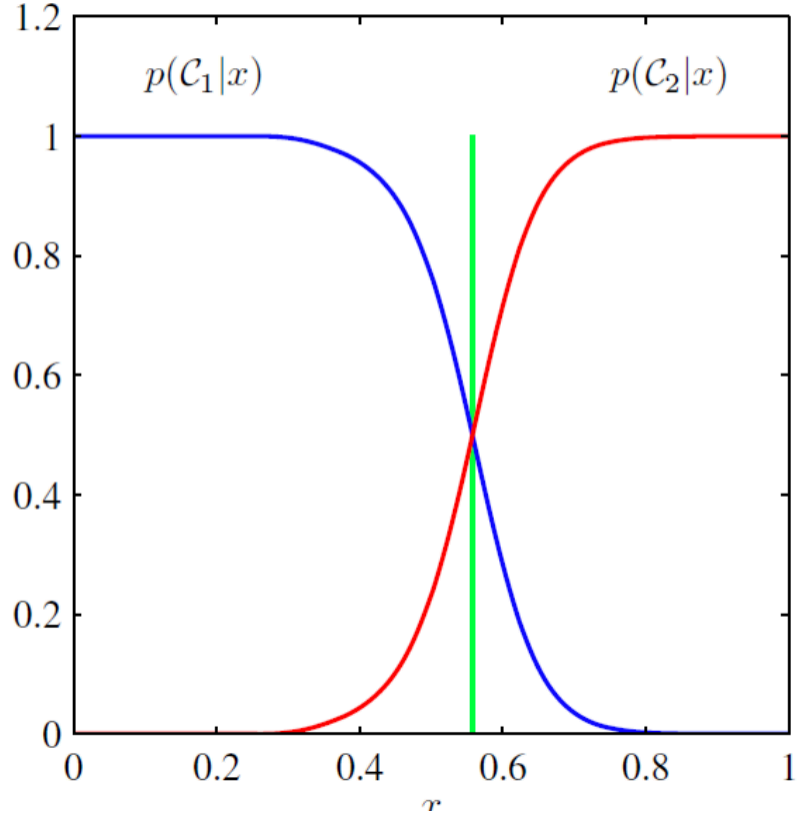
First sample a label



Generative Model's view



Discriminative Model's view

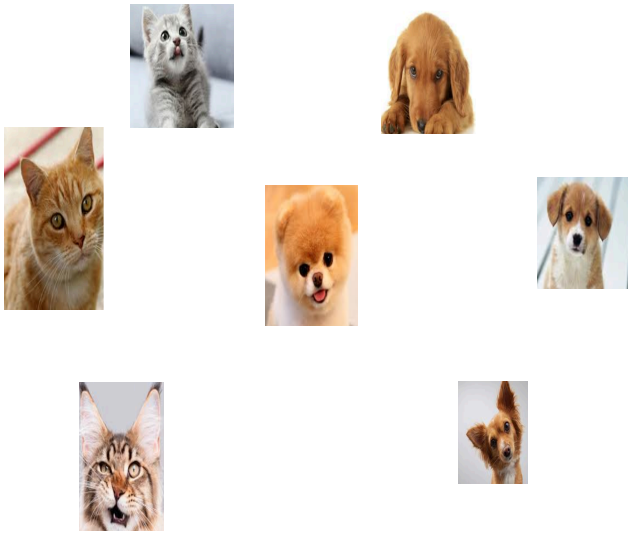


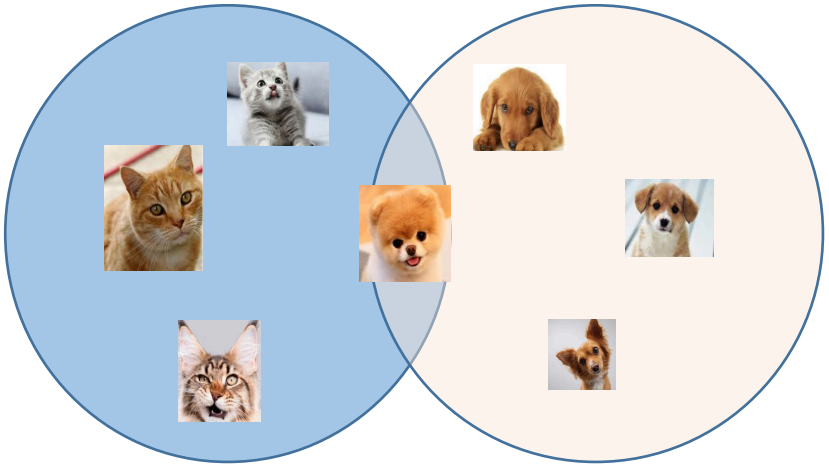
Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- EM

How about unsupervised learning

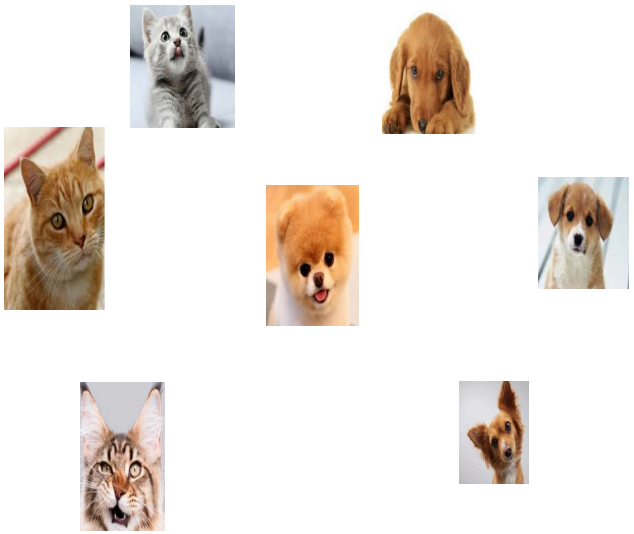
- In unsupervised learning, we only observed input distribution $\tilde{P}(X)$

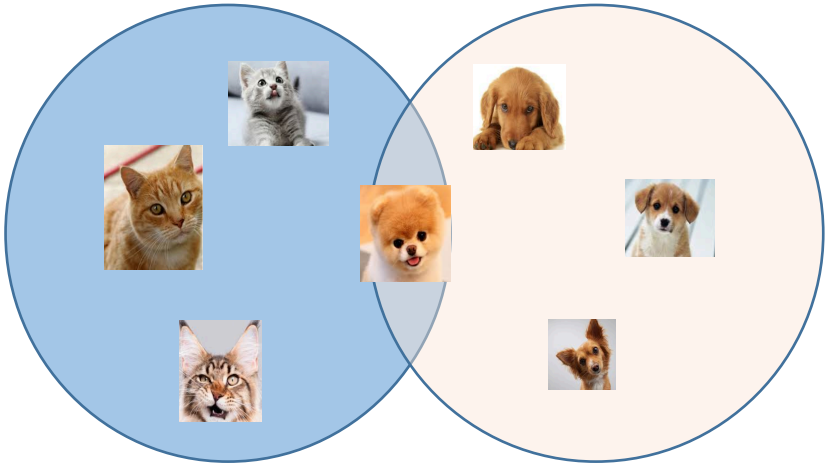




MLE in unsupervised learning

- We only have observation of $\tilde{P}(X)$
- In generative model, we have $P(X, Y \mid \Theta)$
- In discriminative model, we have $P(Y \mid \Theta, X)$
- Which model is more suitable for unsupervised learning?





MLE in unsupervised learning

- We only have observation of $\tilde{P}(X)$
- In generative model, we have $P(X, Y \mid \Theta)$
- We know $P(X \mid \Theta) = \sum_Y P(X, Y \mid \Theta)$
- Therefore, MLE is
  $\text{argmax}_\Theta P(X \mid \Theta) =$
  $\text{argmax}_\Theta \sum_Y P(X, Y \mid \Theta)$

EM algorithm

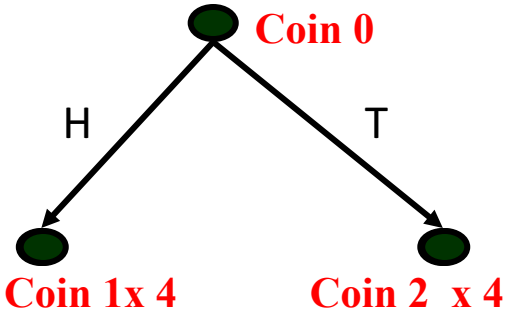
- EM algorithm solves $\text{argmax}_\Theta \sum_Y P(X, Y \mid \Theta)$ by iteratively updating Θ
- In general, known to converge to a local maximum of the maximum likelihood function

Three Coins Example

- We observe a series of coin tosses generated in the following way:
- A person has three coins.
  - Coin 0: probability of Head is α
  - Coin 1: probability of Head p
  - Coin 2: probability of Head q

- Consider the following coin-tossing scenarios:

Scenario I



- Toss coin 0.
  If Head – toss coin 1; o/w – toss coin 2

Observing the sequence
HHHHT, THTHT, HHHHT, HHTTH

produced by Coin 0 , Coin1 and Coin2

Question: Estimate most likely values for p, q (the probability of H in each coin) and the probability to use each of the coins (a)

Scenario I

Supervised Learning

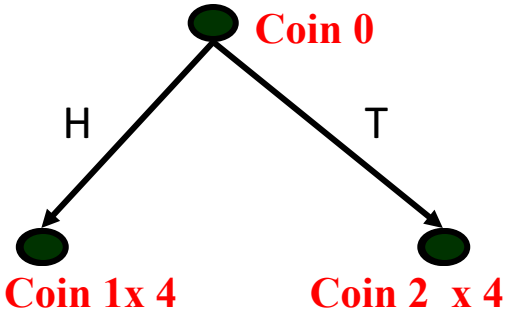
- Toss coin 0.
  If Head – toss coin 1; o/w – toss coin 2

  Observing the sequence
  HHHHT, THTHT, HHHHT, HHTTH

  produced by Coin 0 , Coin1 and Coin2

  Question: Estimate most likely values for p, q (the probability of H in each coin) and the probability to use each of the coins (a)

Scenario II



- Toss coin 0.
  If Head – toss coin 1; o/w – toss coin 2

Observing the sequence
HHHT, HTHT, HHHT, HTTH

produced by ~~Coin 0~~, Coin1 and/or Coin2

Question: Estimate most likely values for p, q (the probability of H in each coin) and the probability to use each of the coins (a)

Intuition of EM algorithm

- Use an iterative approach for estimating the parameters:
  - Guess the probability that a given data point came from Coin 1 or 2; Generate fictional labels, weighted according to this probability.
  - Now, compute the most likely value of the parameters. [recall the scenario I]
  - Compute the likelihood of the data given this model.
  - Re-estimate the initial parameter setting: set them to maximize the likelihood of the data.

Step 1: initialization

Coin 0: probability of Head is α
Coin 1: probability of Head p
Coin 2: probability of Head q

- Guess the probability that a given data point came from Coin 1 or 2; Generate fictional labels, weighted according to this probability.

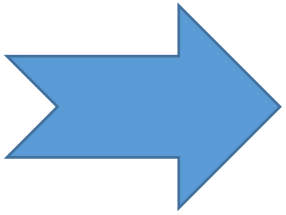
	coin 1=H	coin 1=T
HHHT	100%	0 %
HTHT	100%	0%
HHHT	100%	0%
HTTH	0%	100%

Step 2: Maximum Conditional Likelihood

Coin 0: probability of Head is α
Coin 1: probability of Head p
Coin 2: probability of Head q

- Now, compute the most likely value of the parameters. [recall the scenario I]

	coin 1=H	coin 1=T
HHHT	100%	0 %
HTHT	100%	0%
HHHT	100%	0%
HTTH	0%	100%



HHHHT
HHTHT
HHHHT
THTTH

Step 2: Maximum Conditional Likelihood

Coin 0: probability of Head is α
Coin 1: probability of Head p
Coin 2: probability of Head q

- Now, compute the most likely value of the parameters. [recall the scenario I]

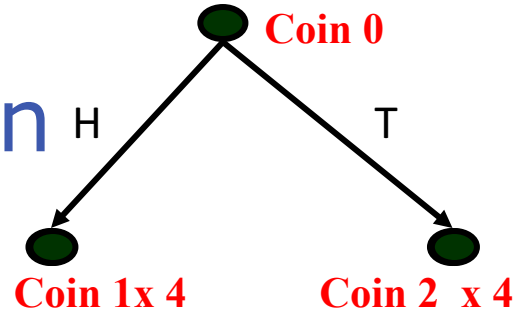
HHHHT
HHTHT
HHHHT
THTTH

$$\alpha_1 = \frac{3}{3+1} = \frac{3}{4}$$

$$p_1 = \frac{8}{8+4} = \frac{2}{3}$$

$$q_1 = \frac{2}{2+2} = \frac{1}{2}$$

Step3: Likelihood Estimation



- Compute the likelihood of the data given this model

$\alpha_1 p_1^{\#H}(1-p_1)^{\#T}$ $\quad$ $(1-\alpha_1)q_1^{\#H}(1-q_1)^{\#T}$

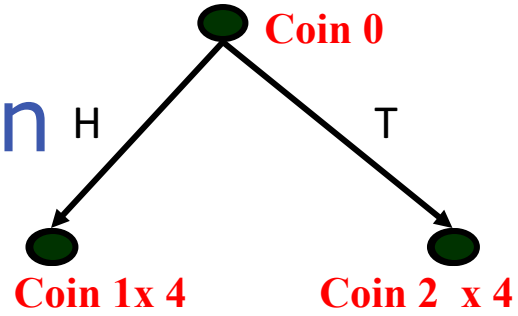
	coin 1=H	coin 1=T
HHHT	$\frac{3}{4}\left(\frac{2}{3}\right)^3\frac{1}{3}$	$\frac{1}{4}\left(\frac{1}{2}\right)^3\frac{1}{2}$
HTHT	$\frac{3}{4}\left(\frac{2}{3}\right)^2\left(\frac{1}{3}\right)^2$	$\frac{1}{4}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^2$
HHHT	$\frac{3}{4}\left(\frac{2}{3}\right)^3\frac{1}{3}$	$\frac{1}{4}\left(\frac{1}{2}\right)^3\frac{1}{2}$
HTTH	$\frac{3}{4}\left(\frac{2}{3}\right)^2\left(\frac{1}{3}\right)^2$	$\frac{1}{4}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^2$

$$\alpha_1 = \frac{3}{4}$$

$$p_1 = \frac{2}{3}$$

$$q_1 = \frac{1}{2}$$

Step3: Likelihood Estimation



- Compute the likelihood of the data given this model

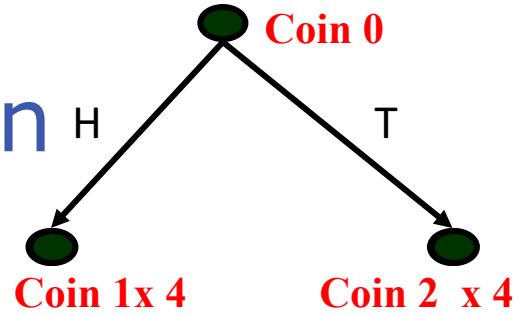
	$\alpha_1 p_1^{\#H}(1-p_1)^{\#T}$ coin 1=H	$(1-\alpha_1)q_1^{\#H}(1-q_1)^{\#T}$ coin 1=T
HHHT	0.074	0.0156
HTHT	0.037	0.0156
HHHT	0.074	0.0156
HTTH	0.037	0.0156

$$\alpha_1 = \frac{3}{4}$$

$$p_1 = \frac{2}{3}$$

$$q_1 = \frac{1}{2}$$

Step3: Likelihood Estimation



- Compute the likelihood of the data given this model

$\alpha_1 p_1^{\#H}(1-p_1)^{\#T}$ $(1-\alpha_1) q_1^{\#H}(1-q_1)^{\#T}$

	coin 1=H	coin 1=T
HHHT	82.6%	17.4%
HTHT		29.7%
HHHT	82.6%	17.4%
HTTH	*70.3%*	29.7%

$$\alpha_1 = \frac{3}{4}$$

$$p_1 = \frac{2}{3}$$

$$q_1 = \frac{1}{2}$$

e.g., $\frac{0.074}{0.074+0.0156} = 82.6\%$

Step 2: Maximum Conditional Likelihood

Coin 0: probability of Head is α
Coin 1: probability of Head p
Coin 2: probability of Head q

- Now, compute the most likely value of the parameters. [recall the scenario I]

	coin 1=H	coin 1=T
HHHT	82.6%	17.4%
HTHT		29.7%
HHHT	82.6%	17.4%
HTTH	*70.3%*	29.7%

HHHHT 82.6%
THHHT 17.4%
HHTHT 70.3%
THTHT 29.7%
HHHHT 82.6%
THHHT 17.4%
HHTTH 70.3%
THTTH 29.7%

Step 2: Maximum Conditional Likelihood

Coin 0: probability of Head is α
Coin 1: probability of Head p
Coin 2: probability of Head q

- Now, compute the most likely value of the parameters. [recall the scenario I]

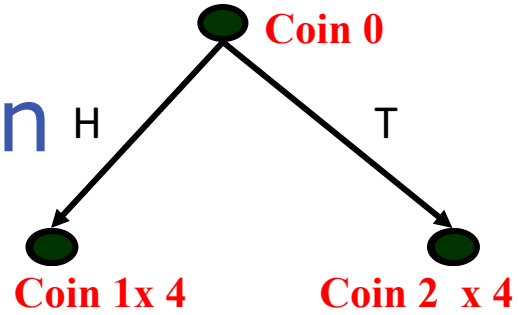
HHHHT 82.6%
HHTHT 70.3%
HHHHT 82.6%
HHTTH 70.3%
THHHT 17.4%
THTHT 29.7%
THHHT 17.4%
THTTH 29.7%

$$\alpha_2 = \frac{82.6\times 2 + 70.3\times 2}{400} = 76.5\%$$

$$p_2 = \frac{82.6\times 6 + 70.3\times 4}{82.6\times 8 + 70.3\times 8} = 63.5\%$$

$$q_2 = \frac{17.4\times 6 + 29.7\times 4}{17.4\times 8 + 29.7\times 8} = 59.2\%$$

Step3: Likelihood Estimation



- Compute the likelihood of the data given this model

$\alpha_2 p_2^{\#H}(1-p_2)^{\#T}$ coin 1=H

$(1-\alpha_2)q_2^{\#H}(1-q_2)^{\#T}$ coin 1=T

HHHT
HTHT
HHHT
HTTH

$$\alpha_2 = 76.5\%$$

$$p_2 = 63.5\%$$

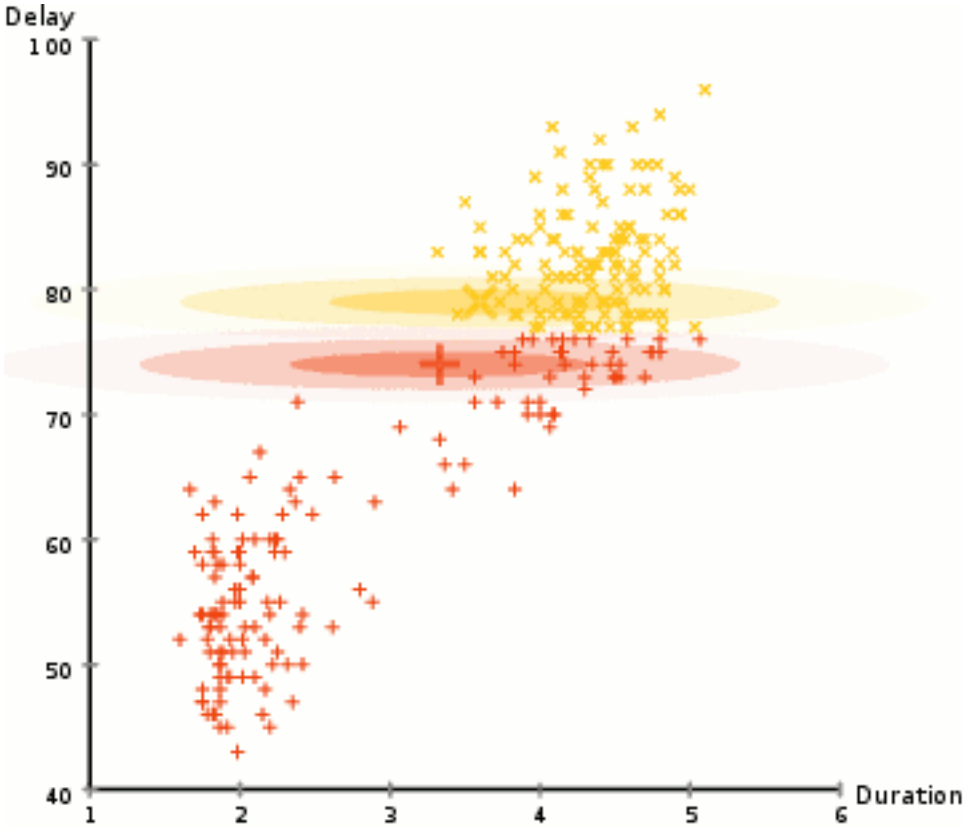
$$q_2 = 59.2\%$$

Intuition of EM algorithm

- Use an iterative approach for estimating the parameters:
  - Guess the probability that a given data point came from Coin 1 or 2; Generate fictional labels, weighted according to this probability.
  - Now, compute the most likely value of the parameters. [recall the scenario I]
  - Compute the likelihood of the data given this model.
  - Re-estimate the initial parameter setting: set them to maximize the likelihood of the data.

Real world Example

GMM clustering of Old Faithful eruption data



GMM as the marginal distribution P(x) of a joint distribution P(x, z)

- Remember, in GMM, we model the marginal probability as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\omega_k = p(z = k)$$

Iterative procedure

- LetType equation here. θ represent all parameters $\{\omega_k, \mu_k, \Sigma_k\}$

Step 0: initialize $\boldsymbol{\theta}$ with some values (random or otherwise)

Step 1: compute γ_{nk} using the current $\boldsymbol{\theta}$

Step 2: update $\boldsymbol{\theta}$ using the just computed γ_{nk}

Step 3: go back to Step 1

γ_{nk}: given a data point x_n how likely it belongs to k^{th} cluster

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Since γ_{nk} is binary, the previous solution is nothing but

- For ω_k: count the number of data points whose z_n is k and divide by the total number of data points (note that $\sum_k \sum_n \gamma_{nk} = N$)
- For $\boldsymbol{\mu}_k$: get all the data points whose z_n is k, compute their mean
- For $\boldsymbol{\Sigma}_k$: get all the data points whose z_n is k, compute their covariance matrix

Estimate γ_{nk}

- γ_{nk} the assignment of instance n to cluster k, can be defined as $\gamma_{nk} = P(z_n = k|x_n)$
- Can be computed vis the posterior probability

$$p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$

$N(x|\mu_k, \Sigma_k)$ ω_k

Parameter estimation for GMMs

- If cluster assignments are observed $\{z_n\}$ are given
  - We know the cluster of each point
  - Let $\gamma_{nk} = 1$ if instance n belongs to cluster k, otherwise $\gamma_{nk} = 0$
- Then the maximum likelihood estimation is

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$