Lecture 15: Bayesian Learning
Winter 2018

Kai-Wei Chang

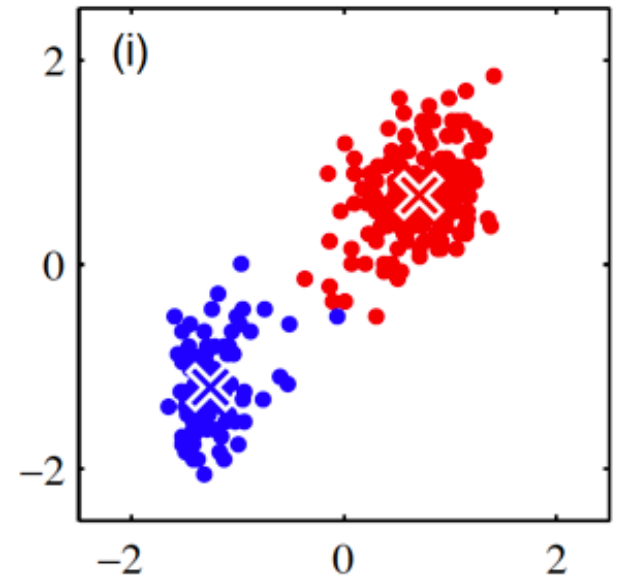
CS @ UCLA

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 interpretation of clustering?

- Until now, we make a hard assignment for clustering
  - Each point assigns to one cluster
  - Can we allow probability in the assignment?



Recap: Bayes Theorem

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

Short for

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

Bayes Theorem

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

Bayes Theorem

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

Prior probability: What is our belief in Y before we see X?

Bayes Theorem

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

Likelihood: What is the likelihood of observing X given a specific Y?

Prior probability: What is our belief in Y before we see X?

Bayes Theorem

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

Posterior probability: What is the probability of Y given that X is observed?

Likelihood: What is the likelihood of observing X given a specific Y?

Prior probability: What is our belief in Y before we see X?

Bayes Theorem

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

Posterior probability: What is the probability of Y given that X is observed?

Likelihood: What is the likelihood of observing X given a specific Y?

Prior probability: What is our belief in Y before we see X?

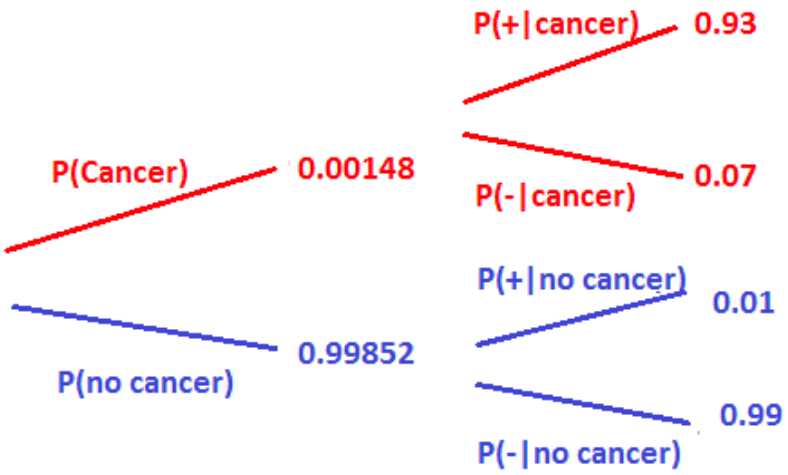
$$\forall x, y \quad P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

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

Recap: Bayes Theorem Example

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

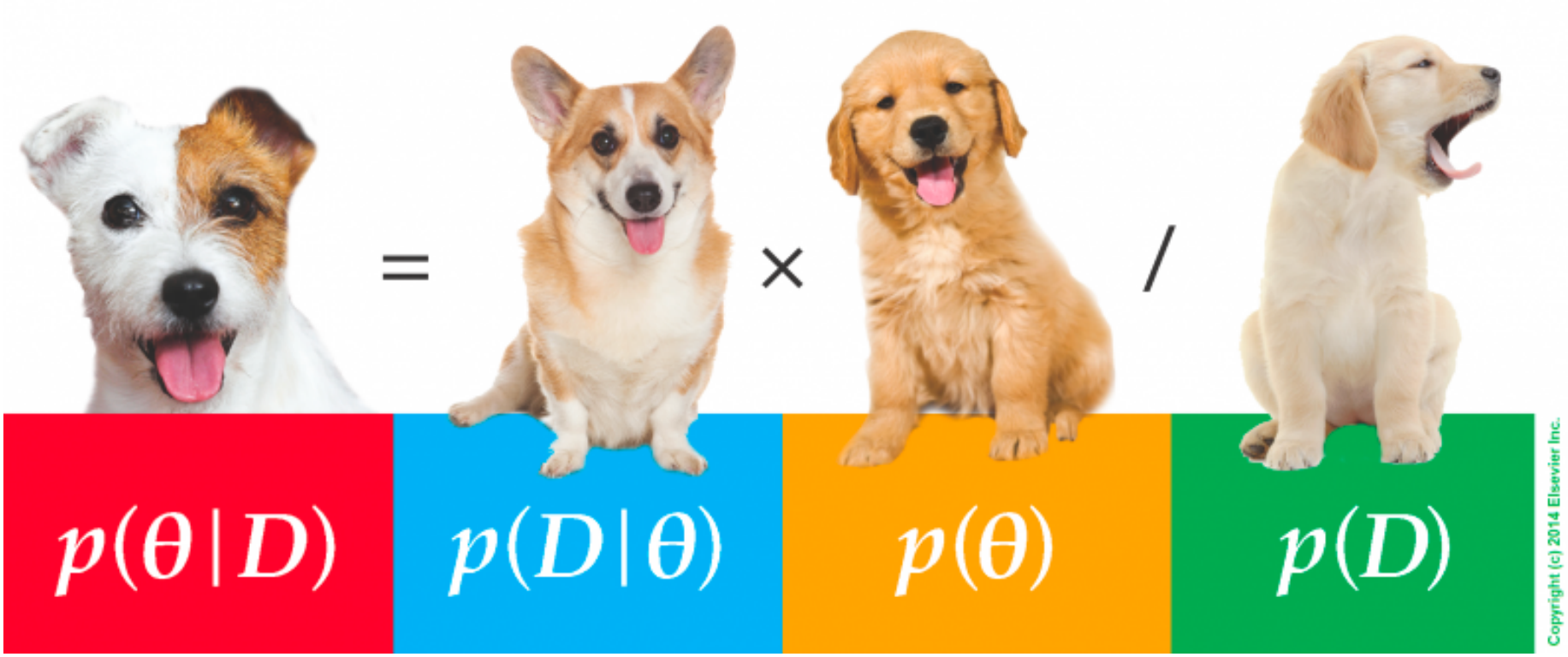
$$P(\text{CANCER}|+) = \frac{P(cancer\ and +)}{P(cancer\ and +) + P(no\ cancer\ and +)} = 0.12$$



Today's lecture

- Bayesian Learning
- Maximum a posteriori and maximum likelihood estimation
- Naïve Bayes

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

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
  - Guide the model towards something we know
- Provide a conceptual framework
  - For evaluating and analyzing learners

Bayesian Learning

Given a dataset D, we want to find the best hypothesis h

What does *best* mean?

Bayesian learning uses $P(h \mid D)$, the conditional probability of a hypothesis given the data, to define *best*.

Bayesian Learning

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

$$P(h|D)$$

Bayesian Learning

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

$$P(h|D)$$

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

Bayesian Learning

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

$$P(h|D)$$

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

Key insight: Both h and D are events.

- D: The event that we observed *this* particular dataset
- h: The event that the hypothesis h is the true hypothesis

So we can apply the Bayes rule here.

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?

Key insight: Both h and D are events.

- D: The event that we observed *this* particular dataset
- h: The event that the hypothesis h is the true hypothesis

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?

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.

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?

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.

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.

Today's lecture

- Bayesian Learning
- Maximum a posteriori and maximum likelihood estimation
- Naïve Bayes

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}

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

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 / Likelihood £ Prior

Choosing a hypothesis

Given some data, find the most probable hypothesis

- The Maximum a Posteriori hypothesis h_{MAP}

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

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 ∈ H

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

2. Output the hypothesis with the highest posterior probability

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

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: Bernoulli trials

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

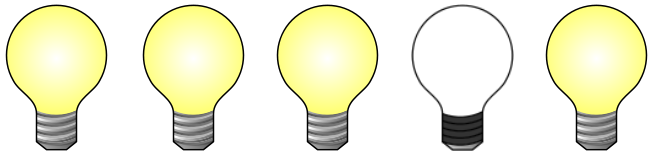
- *CEO*: My company makes light bulbs. I need to know what is the probability they are faulty.
- *You*: Sure. I can help you out. Are they all identical?
- *CEO*: Yes!
- *You*: Excellent. I know how to help. We need to experiment…

Faulty lightbulbs

The experiment:

Try out 100 lightbulbs

80 work, 20 don't



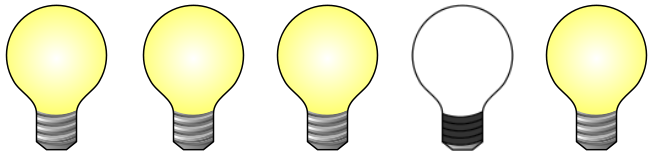
You: The probability is P(failure) = 0.2

CEO: But how do you know?

You: Because…

Bernoulli trials

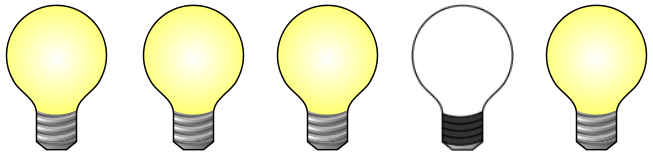
- P(failure) = p, P(success) = 1 – p



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

Bernoulli trials

- P(failure) = p, P(success) = 1 – p



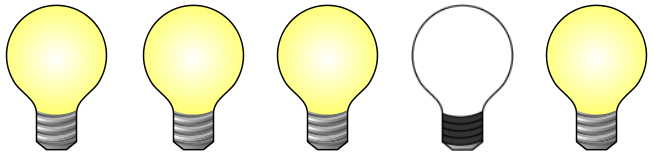
- Each trial is i.i.d
  - Independent and identically distributed

- You have seen D = {80 work, 20 don't}

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

Bernoulli trials

- P(failure) = p, P(success) = 1 – p



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

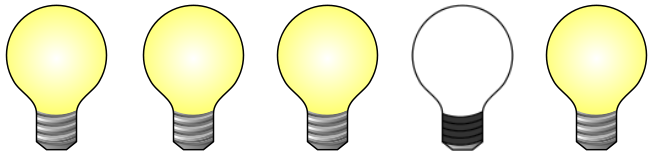
- You have seen D = {80 work, 20 don't}

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

- The most likely value of p for this observation is?

Bernoulli trials

- P(failure) = p, P(success) = 1 – p



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

- You have seen D = {80 work, 20 don't}

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

- The most likely value of p for this observation is?

$$\underset{p}{\operatorname{argmax}}\, P(D|p) = \underset{p}{\operatorname{argmax}} \binom{100}{80} p^{80}(1-p)^{20}$$

The "learning" algorithm

Say you have *a* Work and b Not-Work

$$
\begin{aligned}
p_{best} &= \operatorname*{argmax}_p P(D|h) \\
&= \operatorname*{argmax}_p \log P(D\,|h) \\
&= \operatorname*{argmax}_p \log \left(\binom{a+b}{a} p^a (1-p)^b \right) \\
&= \operatorname*{argmax}_p a \log p + b \log(1-p)
\end{aligned}
$$

Calculus 101: Set the derivative to zero

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

The "learning" algorithm

Say you have *a* Work and b Not-Work

$$p_{best} = \underset{p}{\operatorname{argmax}}\, P(D|h)$$

$$= \underset{p}{\operatorname{argmax}} \log P(D\,|h)$$

Log likelihood

$$= \underset{p}{\operatorname{argmax}} \log \left(\binom{a+b}{a} p^a (1-p)^b \right)$$

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

Calculus 101: Set the derivative to zero

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

The "learning" algorithm

Say you have *a* Work and b Not-Work

$$p_{best} = \underset{p}{\operatorname{argmax}}\, P(D|h)$$

$$= \underset{p}{\operatorname{argmax}} \log P(D\,|h)$$

Log likelihood

$$= \underset{p}{\operatorname{argmax}} \log \left(\binom{a+b}{a} p^a (1-p)^b \right)$$

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

Calculus 101: Set the derivative to zero

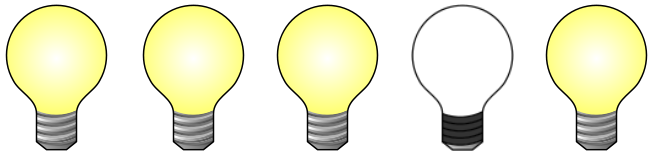
$$P_{best} = b/(a + b)$$

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…

CEO: Okay, but you only test 100 lightbulbs, can you calibrate your results based on our prior test?

MAP estimation

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

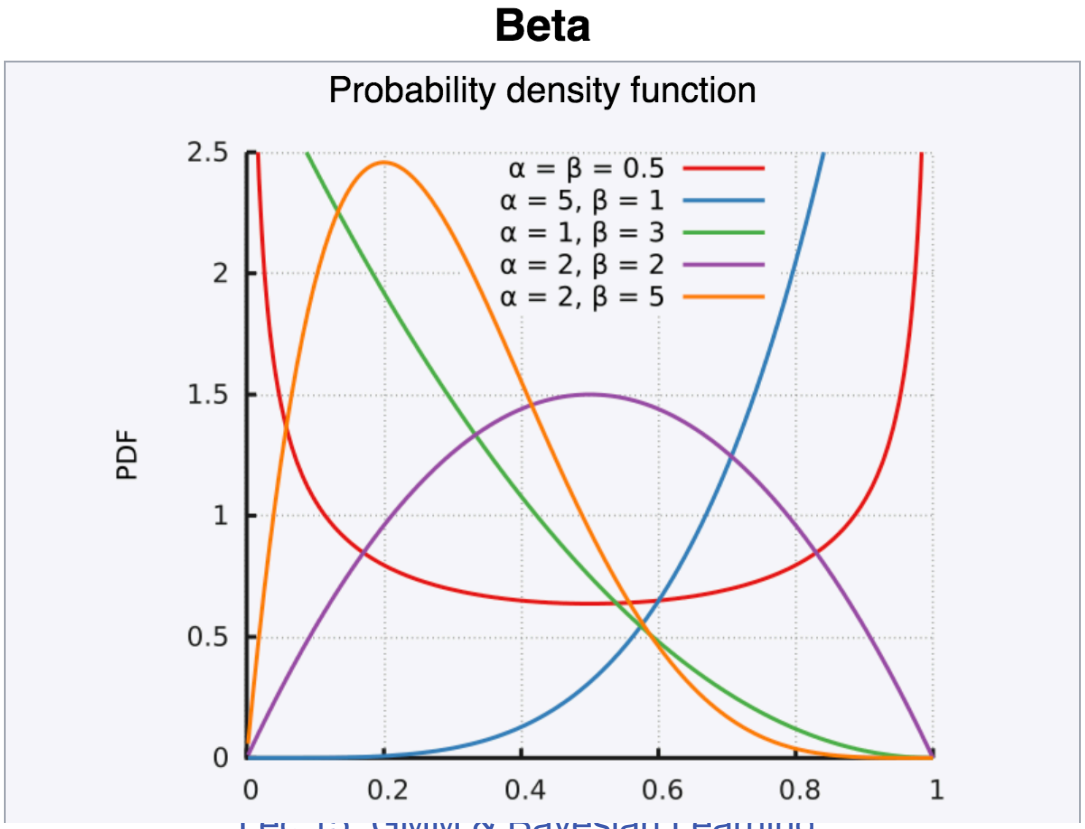
- Simplify this to get the Maximum Likelihood hypothesis

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

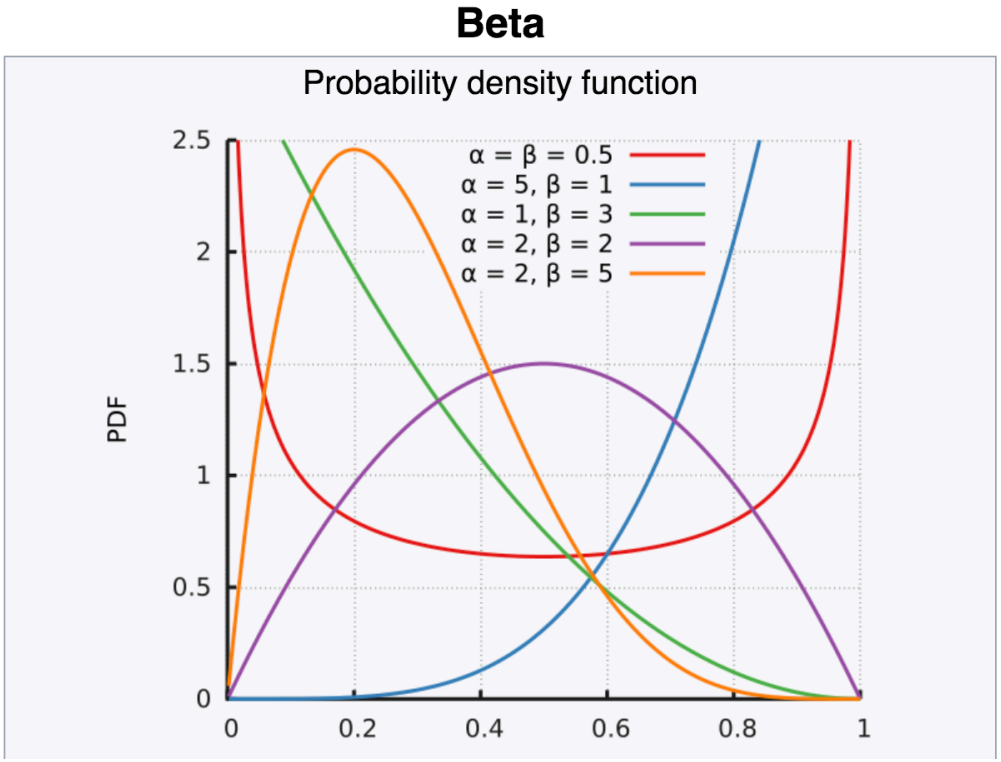
Often computationally easier to maximize *log likelihood*

Prior distribution

- The boss has a prior belief of the distribution of faulty lightbulb



Beta distribution



$$f(x;\alpha,\beta) = \text{constant} \cdot x^{\alpha-1}(1-x)^{\beta-1}$$

$$= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}$$

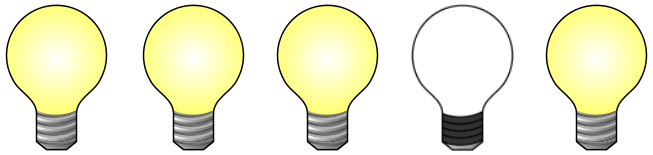
$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$$= \frac{1}{\mathrm{B}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

MAP for Bernoulli trials

p is the parameter for hypothesis

- P(failure) = p, P(success) = 1 – p



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

- You have seen D = {80 work, 20 don't}

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

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

MAP estimation

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

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

$$P(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

$$p_{best} = \operatorname*{argmax}_p P(D|h)\, P(h)$$

$$= \operatorname*{argmax}_p \log P(D\,|h) + \log P(h)$$

$$= \operatorname*{argmax}_p \log \left(\frac{\binom{a+b}{a}}{B(\alpha,\beta)} p^a (1-p)^b p^{\alpha-1}(1-p)^{\beta-1} \right)$$

$$= \operatorname*{argmax}_p \; (a+\alpha-1)\log p + (b+\beta-1)\log(1-p)$$

MAP estimation

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

$$P(D|p) = \binom{100}{80} p^{80}(1-p)^{20}$$

$$P(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

$$p_{best} = \operatorname*{argmax}_p P(D|h)\, P(h)$$

$$= \operatorname*{argmax}_p \log P(D\,|h) + \log P(h)$$

$$= \operatorname*{argmax}_p \log \left(\frac{\binom{a+b}{a}}{B(\alpha,\beta)} p^a (1-p)^b p^{\alpha-1}(1-p)^{\beta-1} \right)$$

$$= \operatorname*{argmax}_p \; (a+\alpha-1)\log p + (b+\beta-1)\log(1-p)$$

MAP v.s. MLE

- MLE:

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

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

- MAP

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

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

MAP v.s. MLE

- MAP

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

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

- Let $\alpha=100,\ \ \beta=10$
- $a=10, b=20 \Rightarrow p_{best}\approx 0.79$
- $a=1000, b=2000 \Rightarrow p_{best}\approx 0.36$
- $a=100{,}000, b=200{,}000 \Rightarrow p_{best}\approx 0.33$

Advanced topic (not cover in exam) MAP for logistic regression

Let's get back to the MLE for logistic regression

- Training data
  - S = $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, m examples
- What we want
  - Find a **w** such that P(S | **w**) is maximized
  - We know that our examples are drawn independently and are identically distributed (i.i.d)
  - How do we proceed?

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}}\, P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i, \mathbf{w})$$

The usual trick: Convert products to sums by taking log

Recall that this works only because log is an increasing function and the maximizer will not change

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}} P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}} P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i, \mathbf{w})$$

$$\max_{\mathbf{w}} \sum_{i}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

But (by definition) we know that

$$P(y|\mathbf{w}, \mathbf{x}) = \sigma(y_i \mathbf{w}^T \mathbf{x}_i) = \frac{1}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)}$$

$$P(y|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathrm{y_i}\mathbf{w}^T\mathbf{x}_i)}$$

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}}\, P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i,\mathbf{w})$$

$$\max_{\mathbf{w}} \sum_{i}^{m} \log P(y_i|\mathbf{x}_i,\mathbf{w})$$

Equivalent to solving

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

$$P(y|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathrm{y_i}\mathbf{w}^T\mathbf{x}_i)}$$

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}}\, P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i,\mathbf{w})$$

The goal: Maximum likelihood training of a discriminative probabilistic classifier under the logistic model for the posterior distribution.

$$\max_{\mathbf{w}} \sum_{i}^{m} \log P(y_i|\mathbf{x}_i,\mathbf{w})$$

Equivalent to solving

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

$$P(y|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathrm{y_i}\mathbf{w}^T\mathbf{x}_i)}$$

Maximum likelihood estimation

$$\underset{\mathbf{w}}{\operatorname{argmax}}\, P(S|\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} P(y_i|\mathbf{x}_i,\mathbf{w})$$

The goal: Maximum likelihood training of a discriminative probabilistic classifier under the logistic model for the posterior distribution.

$$\max_{\mathbf{w}} \sum_{i}^{m} \log P(y_i|\mathbf{x}_i,\mathbf{w})$$

Equivalent to solving

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

Equivalent to: Training a linear classifier by minimizing the ***logistic loss***.

Maximum a posteriori estimation

We could also add a prior on the weights

Suppose each weight in the weight vector is drawn independently from the normal distribution with zero mean and standard deviation σ

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Let us work through this procedure again to see what changes

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Let us work through this procedure again to see what changes

What is the goal of MAP estimation? (In maximum likelihood, we maximized the likelihood of the data)

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

What is the goal of MAP estimation? (In maximum likelihood, we maximized the likelihood of the data)

To maximize the posterior probability of the model given the data (i.e. to find the most probable model, given the data)

$$P(\mathbf{w}|S) \propto P(S|\mathbf{w})P(\mathbf{w})$$

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\underset{\mathbf{w}}{\text{argmax}}\, P(\mathbf{w}|S) = \underset{\mathbf{w}}{\text{argmax}}\, P(S|\mathbf{w})P(\mathbf{w})$$

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

We have already expanded out the first term.

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

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

Expand the log prior

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

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

$$\max_{\mathbf{w}} \sum_{i}^{m} -\log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i) + \sum_{j=1}^{d} \frac{-w_i^2}{\sigma^2} + constants$$

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

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

MAP estimation for logistic regression

Maximum likelihood estimation

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

$$\max_{\mathbf{w}} \sum_{i=1}^{m} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

Equivalent to solving

$$\max_{\mathbf{w}} \sum_{i=1}^{m} -\log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

$$p(\mathbf{w}) = \prod_{j=1}^{d} p(w_i) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w_i^2}{\sigma^2}\right)$$

Learning by solving

$$\operatorname{argmax}_{\mathbf{w}} P(S|\mathbf{w})P(\mathbf{w})$$

Take log to simplify

$$\max_{\mathbf{w}} \log P(S|\mathbf{w}) + \log P(\mathbf{w})$$

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

Maximizing a negative function is the same as minimizing the function

Learning a logistic regression classifier

Learning a logistic regression classifier is equivalent to solving

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

Today's lecture

- GMM
- Bayesian Learning
- Maximum a posteriori and maximum likelihood estimation
- Naïve Bayes

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

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?

	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

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

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

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

← $P(x_j = 1 \mid y = 1) = a_j$

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

← $P(x_j = 1 \mid y = 0) = b_j$

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

   2. For each word w in vocabulary :

How often does a word occur with a label?

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

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

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

   2. For each word w in vocabulary :

Smoothing

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

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