CS 536 – Montana State University

Bayesian Learning
Chapter 6 of Tom Mitchell's Machine Learning Book

Sections 6.5-6.10

Neal Richter – March 27th 2006

Slides adapted from Mitchell's lecture notes and
Dr. Geehyuk Lee's Machine Learning class at ICU

Bayes Theorem *(Review from last week)*

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

$$\begin{aligned} P(h) &= \text{prior probability of hypothesis } h \\ P(D) &= \text{prior probability of training data } D \\ P(h|D) &= \text{probability of } h \text{ given } D \\ P(D|h) &= \text{probability of } D \text{ given } h \end{aligned}$$

- Note a symmetry in the equation:
  - The equation remains in the same form if you exchange h and D.
- Can you explain the meaning of P(D)?

Choosing Hypotheses *(Review from last week)*

- Maximum *a posteriori* (MAP) hypothesis:

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

- Maximum likelihood (ML) hypothesis:

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

- When do these two become the same?

Basic Formulas for Probabilities *(Review from last week)*

Product Rule:

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

Sum Rule:

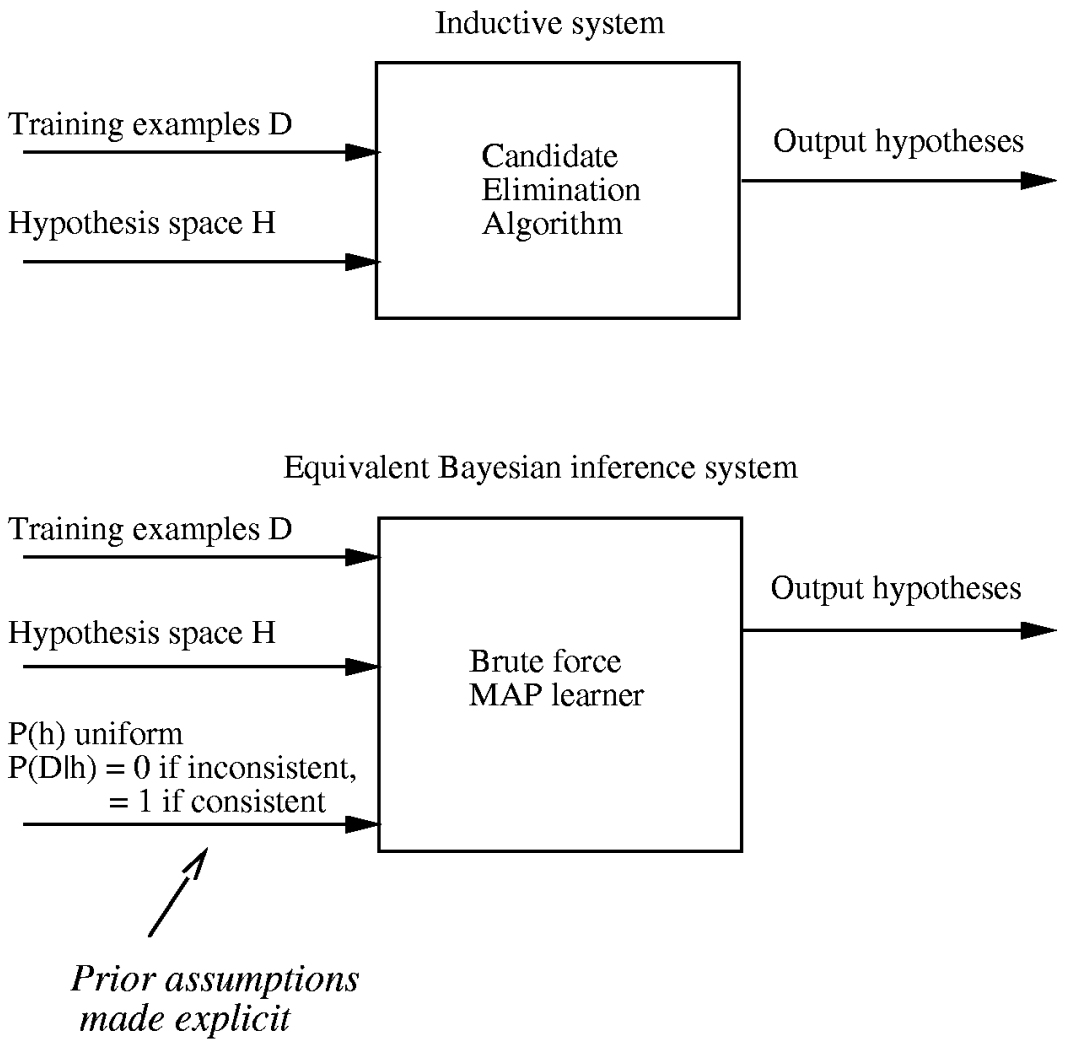
$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Theorem of total probability:

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

if $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$.

Characterizing Concept Learning by Equivalent MAP Learners *(Review from last week)*



Learning to Predict Probabilities

Consider predicting survival probability from patient data.

- (Stochastic) Examples: $\langle x_i, d_i \rangle$, $d_i \in \{0, 1\}$
- Need a non-deterministic classifier, but MLP is not.
- Instead, MLP that outputs a *probability* given x_i.
- Conditional probability:

$$P(D|h) = \prod_{i=1}^{m} P(x_i, d_i|h) = \prod_{i=1}^{m} P(d_i|h, x_i)P(x_i)$$

- Note $P(d_i|h, x_i)$ is either 0 or 1.

$$P(d_i|h, x_i) = h(x_i)^{d_i}(1 - h(x_i))^{1-d_i}$$

$$P(D|h) = \prod_{i=1}^{m} h(x_i)^{d_i}(1 - h(x_i))^{1-d_i} P(x_i)$$

Learning to Predict Probabilities (2)

- ML hypothesis:

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} h(x_i)^{d_i} (1 - h(x_i))^{1-d_i} P(x_i)$$

- Note that $P(x_i)$ is independent of h.

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} h(x_i)^{d_i} (1 - h(x_i))^{1-d_i}$$

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

- The quantity to be maximized look like an entropy.
- It is in fact $-(const) \times (crossentropy)$
- Cross entropy represents similarity between two probability distributions.

Gradient Search to Maximize Likelihood in NNs

- Let $G(h, D)$ the quantity to be maximized.

$$\begin{aligned}
\frac{\partial G(h,D)}{\partial w_{jk}} &= \sum_{i=1}^{m} \frac{\partial G(h,D)}{\partial h(x_i)} \frac{\partial h(x_i)}{\partial w_{jk}} \\
&= \sum_{i=1}^{m} \frac{\partial (d_i \ln h(x_i) + (1-d_i)\ln(1-h(x_i)))}{\partial h(x_i)} \frac{\partial h(x_i)}{\partial w_{jk}} \\
&= \sum_{i=1}^{m} \frac{d_i - h(x_i)}{h(x_i)(1-h(x_i))} \frac{\partial h(x_i)}{\partial w_{jk}}
\end{aligned}$$

- For a single-layer NN

$$\frac{\partial h(x_i)}{\partial w_{jk}} = \sigma'(x_i) x_{ijk} = h(x_i)(1-h(x_i)) x_{ijk}$$

$$\frac{\partial G(h,D)}{\partial w_{jk}} = \sum_{i=1}^{m} (d_i - h(x_i)) x_{ijk}$$

- This is going to take us to something similar to LMS learning rule...

Comparison to NN Sigmoid Update Rule

Consider predicting survival probability from patient data

Training examples $\langle x_i, d_i \rangle$, where d_i is 1 or 0

Want to train neural network to output a *probability* given x_i (not a 0 or 1)

In this case can show

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

Weight update rule for a sigmoid unit:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

where

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} h(x_i)(1 - h(x_i)\,(d_i - h(x_i))\ x_{ijk} \qquad \text{(With sigmoid derivative)}$$

Minimum Description Length Principle

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\
&= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\
&= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \quad (1)
\end{aligned}
$$

Interesting fact from information theory:

> The optimal (shortest expected coding length) code for an event with probability p is $-\log_2 p$ bits.

So interpret (1):

- $-\log_2 P(h)$ is length of h under optimal code
- $-\log_2 P(D|h)$ is length of D given h under optimal code

$\rightarrow$ prefer the hypothesis that minimizes

$$length(h) + length(misclassifications)$$

Minimum Description Length Principle (2)

- MDL: prefer the hypothesis h that minimizes

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

where $L_C(x)$ is the description length of x under encoding C.

- Occam's razor: prefer the shortest hypothesis

- Does this mean that a hypothesis chosen by the MDL principle will be the MAP hypothesis?

Most Probable Classification of New Instances

- So far we've sought the most probable *hypothesis* h_{MAP} given the data D.
- Given new instance x, what is its most probable *classification*?
- Is $h_{MAP}(x)$ the most probable classification?

Consider:

- Three possible hypotheses:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- Given new instance x,

$$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- What's most probable classification of x?

Bayes Optimal Classifier

Bayes optimal classification:

$$\arg\max_{v_j \in V} P(v_j|D) = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

In our example,

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = 0.4 + 0 + 0 = 0.4$$

$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = 0 + 0.3 + 0.3 = 0.6$$

Therefore,

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Gibbs Classifier/Sampler

Bayes optimal classifier provides best result, but can be expensive if many hypotheses.

Gibbs algorithm:

- Choose one hypothesis at random, according to $P(h|D)$
- Use this to classify new instance

Surprising fact: if target concepts are drawn at random from H according to priors on H, then

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

For uniform prior distribution over H,

- Pick any hypothesis from VS, with uniform probability
- Its expected error no worse than twice Bayes optimal

Naive Bayes Classifier

Along with decision trees, neural networks, nearest nbr, one of the most practical learning methods.

When to use

- Moderate or large training set available
- Attributes that describe instances are conditionally independent given classification

Successful applications:

- Diagnosis
- Classifying text documents

Naive Bayes Classifier (2)

Assume target function $f : X \rightarrow V$, where each instance x described by attributes $\langle a_1, a_2 \ldots a_n \rangle$.

Most probable value of $f(x)$:

$$
\begin{aligned}
v_{MAP} &= \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n) \\
v_{MAP} &= \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\
&= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j)
\end{aligned}
$$

Naive Bayes classifier:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

Naive Bayes Algorithm

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Naive Bayes: Example

Consider *PlayTennis* again, and new instance

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Want to compute:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$

$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

Naive Bayes: Subtleties

- Conditional independence assumption is often violated

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

...but it works surprisingly well anyway.

- Estimated posteriors $\hat{P}(v_j|x)$ need not be correct; need only that

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots, a_n | v_j)$$

  - See [Domingos & Pazzani, 1996] for analysis.

- Naive Bayes posteriors often unrealistically close to 1 or 0

Naive Bayes: Subtleties (2)

- What if none of the training instances with target value v_j have attribute value a_i? Then,

$$\hat{P}(a_i|v_j) = 0, \text{ and...} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

- Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

Learning to Classify Text

- Target concept:

$$Interesting? : Document \rightarrow \{+, -\}$$

1. Represent each document by vector of words.
   - One attribute per word position in document
2. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Learning to Classify Text

- Naive Bayes conditional independence assumption:

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

  - where $P(a_i = w_k|v_j)$ is probability that word in position i is w_k, given v_j

- One more assumption:

$$P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$$

  - What does this mean?
  - Is this a plausible assumption?

Learn_Naïve_Bayes_Text (*Examples, V*)

1. Collect all words and other tokens that occur in $Examples$
   - $Vocabulary \leftarrow$ all distinct words and other tokens in $Examples$
2. Calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms
   - For each target value v_j in V do
     - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
     - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
     - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
     - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
     - for each word w_k in $Vocabulary$
       - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
       - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Classify_Naïve_Bayes_Text (*Doc*)

Return the estimated target value for the document *Doc*.

- *positions* $\leftarrow$ all word positions in *Doc* that contain tokens found in $Vocabulary$
- Return v_{NB}, where

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Twenty Newsgroups (Joachims, 1996)

- 1000 training documents from each of 20 groups → 20,000
- Use two third of them in learning to classify new documents according to which newsgroup it came from.

- Newsgroups:
  - comp.graphics, misc.forsale, comp.os.ms-windows.misc, rec.autos, comp.sys.ibm.pc.hardware, rec.motorcycles, comp.sys.mac.hardware, rec.sport.baseball, comp.windows.x, rec.sport.hockey, alt.atheism, sci.space, soc.religion.christian, sci.crypt, talk.religion.misc, sci.electronics, talk.politics.mideast, sci.med, talk.politics.misc, talk.politics.guns

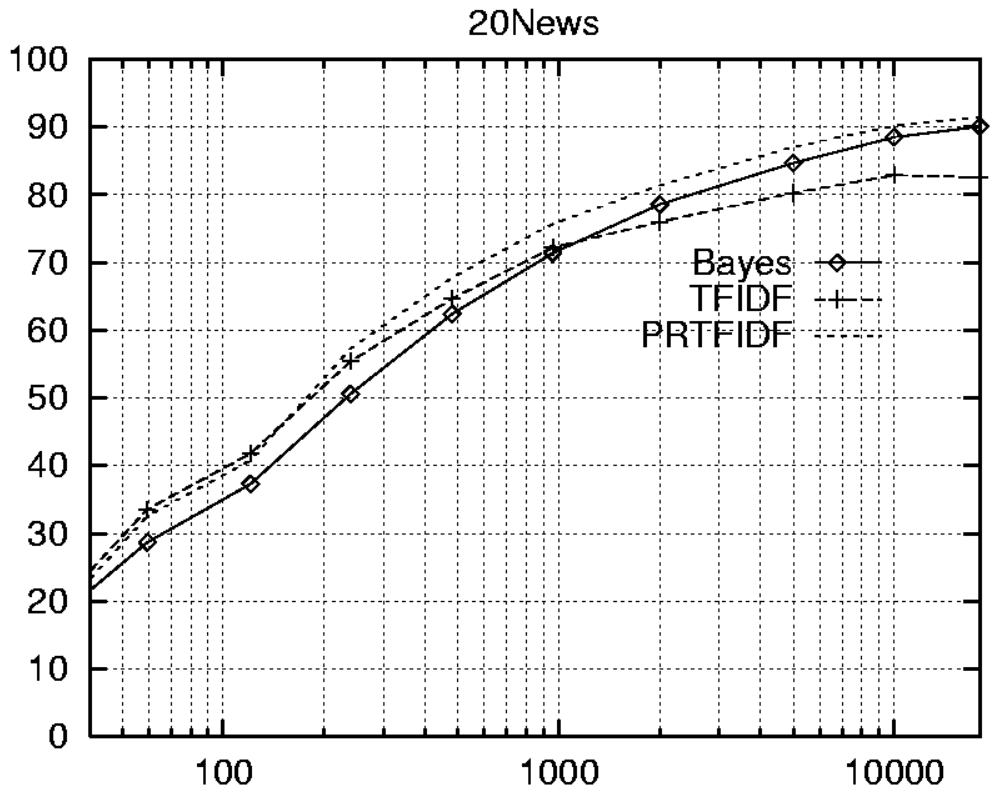
- Naive Bayes: 89% classification accuracy
- Random guess: ?

An article from rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!logicse!uwm.edu
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinion)...
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided …

Learning Curve for 20 Newsgroups



Accuracy vs. Training set size (1/3 withheld for test)

(Note that the x-axis in log scale)