Bayesian Belief Networks

- Characteristics of Bayesian Methods

. providing practical learning algorithms:
Naive Bayes learning
Bayesian belief network learning
. combining prior knowledge:
Causality relationship is represented by the graph.
Prior probabilities are obtained from the observed data.

- Bayes Theorem

. Let h be the hypothesis and D be the data
. The probability of h given D is determined by

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

where

$P(h)$ = the prior probability of hypothesis h
$P(D)$ = the prior probability of data D
$P(D|h)$ = the probability of D given h

- Choosing Hypotheses

. Generally we want the most probable hypothesis given the data, that is, $P(h|D)$.

. The maximum a posteriori (MAP) hypothesis h_{MAP} is selected by

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

. If we assume that $P(h_i) = P(h_j)$, we can further simplify and choose the maximum likelihood (ML) hypothesis using

$$h_{ML} = \text{argmax}_{h_i \in H} P(D|h_i)$$

- Example: Bayes Theorem

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct possible result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.8% of the entire population have this cancer.

Let h = cancer, $\neg h$ = not cancer, $+$ = positive result, and $-$ = negative result.

$$P(h) = 0.008,\ \ P(\neg h) = 0.992$$
$$P(+|h) = 0.98,\ \ P(-|h) = 0.02$$
$$P(+|\neg h) = 0.03,\ \ P(-|\neg h) = 0.97$$

$$h_{MAP} = \text{argmax}_{h \in H} P(h|+) = \text{argmax}_{h \in H} P(+|h)P(h)$$

$$P(+|h)P(h) = 0.98 \cdot 0.008 = 0.0078$$
$$P(+|\neg h)P(\neg h) = 0.03 \cdot 0.992 = 0.0298$$

Therefore, the selected hypothesis is $\neg h$, that is, not cancer.

- Basic Formulas for Probabilities

. Product Rule: probability $P(A \wedge B)$ of a conjunction of two events A and B:

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

. Sum Rule: probability of a disjunction of two events A and B:

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

. Theorem of total probability: if events $A_1, \cdots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

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

- Relation to Concept Learning

. Consider our usual concept learning task

(1) instance space X, hypothesis space H, training examples D

(2) consider the FindS learning algorithm (outputs most specific hypothesis from the version space VS_{HD})

. What would Bayes rule produce as the MAP hypothesis?

. Does FindS output a MAP hypothesis?

. Assume fixed set of instances $\langle x_1, \cdots, x_m \rangle$

. Assume D is the set of classifications $D = \langle c(x_1), \cdots, c(x_m) \rangle$

. Choose $P(D|h)$:

$$P(D|h) = \begin{cases} 1 & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

$$P(h) = \frac{1}{|H|} \quad \text{for all } h \in H.$$

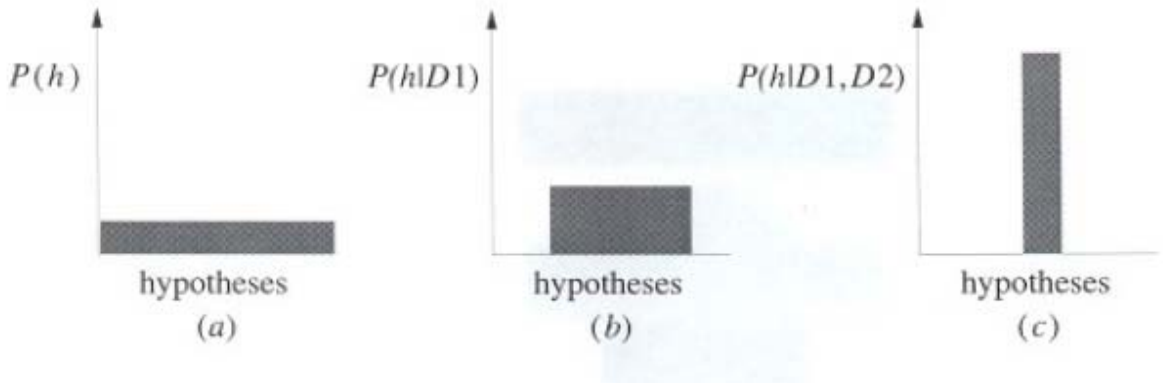
. Then,

$$P(D|h) = \begin{cases} 1/|VS_{HD}| & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

since

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{1 \cdot 1/|H|}{|VS_{HD}|/|H|} = \frac{1}{|VS_{HD}|}.$$

- Evolution of Posterior Probabilities



As we get more samples, the probability distribution is concentrated on certain hypotheses.

- Learning a Real Valued Function

. Consider any real-valued target function f

. Training examples $\langle x_i, d_i \rangle$ where d_i is noisy training value

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

where e_i is random variable (noise) drawn independently for each x_i according to some Gaussian distribution with mean zero.

. Then, the maximum likelihood hypothesis h_{ML} is the one that minimizes the sum of square errors, that is,

$$h_{ML} = \text{argmax}_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2.$$

Proof:

$$
\begin{aligned}
h_{ML} &= \operatorname{argmax}_{h \in H} p(D|h) \\
&= \operatorname{argmax}_{h \in H} \prod_{i=1}^{m} p(d_i|h) \\
&= \operatorname{argmax}_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2\right)
\end{aligned}
$$

Maximize the natural log of this.

$$
\begin{aligned}
h_{ML} &= \operatorname{argmax}_{h \in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi}\,\sigma} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \operatorname{argmax}_{h \in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \operatorname{argmax}_{h \in H} \sum_{i=1}^{m} -(d_i - h(x_i))^2 \\
&= \operatorname{argmin}_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2
\end{aligned}
$$

. Minimizing the mean square error can be interpreted as the maximum likelihood estimate of h when the additive noise has normal distribution.

- Learning to Predict Probabilities

. 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.

. In this case, we can show that

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

Proof:

The training data are given by

$D = \{(x_1, d_1), \cdots, (x_m, d_m)\}$ where $d_i \in \{0,1\}$.

Assuming that each training example is drawn independently,

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

since x_i is independent of h.

$$P(d_i|h, x_i) = \begin{cases} h(x_i) & \text{if } d_i = 1 \\ 1 - h(x_i) & \text{if } d_i = 0 \end{cases}$$

This term can be rewritten as

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

This implies that

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

Assuming $P(x_i) = P(x_j)$ for $i,j = 1,\cdots,m$,

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

- Minimum Description Length (MDL) Principle

. Occam's razor: prefer the shortest hypothesis

. MDL: prefer the hypothesis h that minimizes

$$h_{MDL} = \text{argmin}_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

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

. Example: Let

H = decision tree and D = training data labels.

Then,

$L_{C_1}(h)$ = the number of bits to describe tree h and

$L_{C_2}(D|h)$ = the number of bits to describe D given h.

Note that $L_{C_2}(D|h)=0$ if examples classified perfectly by h, that is, it needs only to describe exceptions.

Hence, h_{MDL} trades off tree size for training errors.

. Let us consider the MAP hypothesis:

$$\begin{aligned} h_{MAP} &= \mathrm{argmax}_{h \in H} P(D|h)P(h) \\ &= \mathrm{argmax}_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \mathrm{argmin}_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \end{aligned}$$

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 h_{MAP} as

(1) $-\log_2 P(h)$ is the length of h and

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

. We can say that MDL hypothesis is the hypothesis that minimizes $length(h)+length(misclassifications)$.

- Most Probable Classification of New Instances

. So far we've sought the most probable hypothesis given the data D, that is, h_{MAP}.

. Given a new instance x, what is its most probable classification?

. $h_{MAP}(h)$ is not the most probable classification!

. Consider three possible hypotheses:

$P(h_1|D) = 0.4$, $P(h_2|D) = 0.3$, and $P(h_3|D) = 0.3$.

Given a new instance x, $h_1(x) = +$, $h_2(x) = -$, and $h_3(x) = -$.

What is the most probable classification of x?

+ according to the MAP hypothesis.

- Bayes Optimal Classifier

. Let $V = \{+,-\}$. Then, for $v_j \in V$

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D).$$

. Bayes optimal classification:

$$\text{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

That is, the most probable classification of new instance is obtained by combining the prediction of all hypotheses weighted by their posterior probabilities.

. Example:

$$P(h_1|D) = 0.4,\ P(-|h_1) = 0,\ P(+|h_1) = 1$$

$$P(h_2|D) = 0.3,\ P(-|h_2) = 1,\ P(+|h_2) = 0$$

$$P(h_3|D) = 0.3,\ P(-|h_3) = 1,\ P(+|h_3) = 0$$

therefore

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

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

and

$$\mathrm{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -.$$

- Gibbs Classifier

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

. Gibbs algorithm:

Step 1. Choose one hypothesis at random according to $P(h|D)$.

Step 2. Use this hypothesis to classify a new instance.

Surprising fact: Assume target concepts are drawn at random from H according to prors on H. Then,

$$E[error_{Gibbs}] \leqq 2E[error_{BayesOptimal}].$$

Suppose uniform prior distribution over H.

Let us pick any hypothesis from VS with uniform probability to classify a new instance.
Then, its expected error is no worse than twice Bayes optimal.

- Naive Bayes Classifier

. When to use:
(1) moderate or large training set available
(2) attributes that describe instances are
conditionally independent given classification
. successful applications: diagnosis, classifying text documents, ...

- Naive Bayes Classifier

. Assume target function $f : X \rightarrow V$ where each instance x described by attributes $\langle a_1, a_2, \cdots, a_n \rangle$. Then, the most probable value of $f(x)$ is

$$v_{MAP} = \text{argmax}_{v_j \in V} P(v_j | a_1, a_2, \cdots, a_n)$$

$$v_{MAP} = \text{argmax}_{v_j \in V} \frac{P(a_1, a_2, \cdots, a_n | v_j) P(v_j)}{P(a_1, a_2, \cdots, a_n)}$$

$$= \text{argmax}_{v_j \in V} P(a_1, a_2, \cdots, a_n | v_j) P(v_j)$$

Here, the naive Bayes assumption is given by

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

This implies that the naive Bayes classifier is

$$v_{NB} = \text{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j).$$

- Naive Bayes Algorithm

For each target value v_j

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

For each attribute value a_i of each attribute a

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

Classify_New_Instance(x): $v_{NB} = \text{argmax}_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j).$

- Naive Bayes: Example of PlayTennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

a new instance:

$\langle Outlook = Sunny,\ Temp = Cool,\ Humidity = High,\ Wind = Strong \rangle$

What is the conclusion using v_{NB}?

We need to compute $v_{NB} = \text{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$.

From the example, we get

$P(Yes) = 9/14,\ P(No) = 5/14$

$P(Sunny|Yes) = 2/9,\ P(Sunny|No) = 3/5$

$P(Cool|Yes) = 3/9,\ P(Cool|No) = 1/5$

$P(High|Yes) = 3/9,\ P(High|No) = 4/5$

$P(Strong|Yes) = 2/9,\ P(Strong|No) = 3/5$

That is,

$P(Yes)P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes) = 0.005$

$P(No)P(Sunny|No)P(Cool|No)P(High|No)P(Strong|No) = 0.021$

Therefore,

$v_{NB} = No$

- Naive Bayes: Subtleties

1. Conditional independence assumption is often violated

$$P(a_1, \cdots, a_n|v_j) = \prod_{i=1}^{n} P(a_i|v_j)$$

but it works surprisingly well any way.

Note that we don't need estimated posteriors $\hat{P}(v_j|x)$ to be correct; need only that

$$\text{argmax}_{v_j \in V} \hat{P}(v_j) \prod_{i=1}^{n} \hat{P}(a_i|v_j) = \text{argmax}_{v_j \in V} P(v_j) P(a_1, \cdots, a_n|v_j)$$

Naive Bayes posteriors often unrealistically close to 1 or 0.

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

$\hat{P}(a_i|v_j) = 0$ and

$$\hat{P}(v_j) \prod_{i=1}^{n} \hat{P}(a_i|v_j) = 0.$$

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

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

where

n = the number of training examples for which $v = v_j$

n_c = the number of examples for which $v = v_j$ and $a = a_i$

p = the prior estimate for $\hat{P}(a_i|v_j)$

m = the weight given to prior, that is, the number of virtual examples

- Learning to Classify Text

. Problems:
Learn which news articles are of interest
Learn to classify web pages by topic
. Naive Bayes is among most effective algorithm
. What attributes shall we use to represent text documents?

. target concept Interesting?: Document -> {+, -}
(1) represent each document by vector of words
(one attribute per word position in document)
(2) learning: use training examples to estimate
$P(+)$, $P(-)$, $P(doc|+)$, $P(doc|-)$

. 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 the probability that word in position i is w_k, given v_j

one more assumption:

$P(a_i = w_k|v_j) = P(a_m = w_k|v_j)$ for all i and m.

Learning_Naive_Bayes_Text (Examples, V)

Step 1. Collect all words and other tokens occur in Examples

$Vocabulary \leftarrow$ all distinct words and other tokens in Examples

Step 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 $docs_j$

- $n \leftarrow$ the total number of words in $Text_j$ (counting duplicate words multiple times)
- for each word w_k in $Vocabulary$

  $n_k \leftarrow$ the number of times word w_k occurs in $Text_j$

  $P(w_k|v_j) \leftarrow \frac{n_k + 1}{n + |Vocbulary|}$

Classify_Naive_Bayes_Text (Doc)

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

$$v_{NB} = \text{argmax}_{v_j \in V} \prod_{i \in positions} P(a_i|v_j)$$

Given 1000 training documents from each group, learn to classify new documents according to which news group it came from

comp.graphics
comp.os.ms-winodows
...
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
...

Naive Bayes: 89% classification accuracy

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvar
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opi
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 tha
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 an
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 o
his own, but because some thugs in Toronto deci
```