

## Bayesian Learning

**Dr Dimitris C. Dracopoulos**

*email:* d.dracopoulos@westminster.ac.uk

Course web page:

<http://users.wmin.ac.uk/~dracopd/DOCUM/courses/2ait608/ait608.html>

1

### Bayes Theorem

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

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

3

### Two Roles for Bayesian Methods

Provides practical learning algorithms:

- Naive Bayes learning
- Bayesian belief network learning
- Combine prior knowledge (prior probabilities) with observed data
- Requires prior probabilities

Provides useful conceptual framework

- Provides a way for evaluating other learning algorithms

2

### Choosing Hypotheses

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

Generally want the most probable hypothesis given the training data

*Maximum a posteriori* hypothesis  $h_{MAP}$ :

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

If assume  $P(h_i) = P(h_j)$  then can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

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

4

## Example

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

$$\begin{aligned}P(\text{cancer}) &= 0.008 & P(\neg\text{cancer}) &= 0.992 \\P(+|\text{cancer}) &= 0.98 & P(-|\text{cancer}) &= 0.02 \\P(+|\neg\text{cancer}) &= 0.03 & P(-|\neg\text{cancer}) &= 0.97\end{aligned}$$

5

## 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  $A_1, \dots, A_n$  are mutually exclusive with  $\sum_{i=1}^n P(A_i) = 1$ , then

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

7

$$P(+|\text{cancer}) \cdot P(\text{cancer}) = 0.98 \cdot 0.008 = 0.0078$$

and

$$P(+|\neg\text{cancer}) \cdot P(\neg\text{cancer}) = 0.03 \cdot 0.992 = 0.0298$$

So:

$$h_{MAP} = \max(0.0078, 0.0298) = \neg\text{cancer}$$

6

## Brute Force MAP Hypothesis Learner

- For each hypothesis  $h$  in  $H$ , calculate the posterior probability

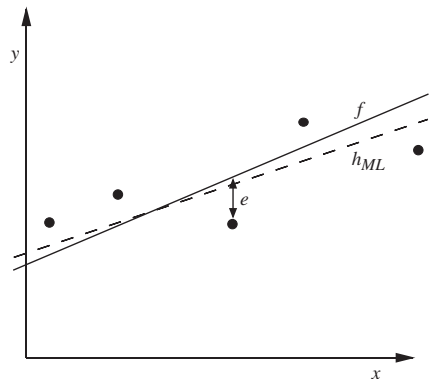
$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- Output the hypothesis  $h_{MAP}$  with the highest posterior probability

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

8

## Learning A Real Valued Function



according to some Gaussian distribution with mean=0

Then the maximum likelihood hypothesis  $h_{ML}$  is the one that minimizes the sum of squared errors:

$$h_{ML} = \arg \min_{h \in H} \sum_{i=1}^m (d_i - h(x_i))^2$$

Consider any real-valued target function  $f$

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

- $d_i = f(x_i) + e_i$
- $e_i$  is random variable (noise) drawn independently for each  $x_i$

10

## Learning A Real Valued Function

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

Maximize natural log of this instead...

$$h_{ML} = \arg \max_{h \in H} \sum_{i=1}^m \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2} \left( \frac{d_i - h(x_i)}{\sigma} \right)^2$$

## 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  $x_i$  (not a 0 or 1)

In this case can show

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

Weight update rule for a sigmoid unit:

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

where

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

and

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

## Bayes Optimal Classifier

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

Example:

$$P(h_1 | D) = .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1$$

$$P(h_2 | D) = .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0$$

$$P(h_3 | D) = .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0$$

therefore

$$\sum_{h_i \in H} P(+|h_i) P(h_i | D) = .4$$

$$\sum_{h_i \in H} P(-|h_i) P(h_i | D) = .6$$

## Gibbs Classifier

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: Assume target concepts are drawn at random from  $H$  according to priors on  $H$ . Then:

$$E[\text{error}_{Gibbs}] \leq 2E[\text{error}_{BayesOptimal}]$$

Suppose correct, uniform prior distribution over  $H$ , then

- 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

17

## 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( $x$ )

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

19

## Naive Bayes Classifier

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

Most probable value of  $f(x)$  is:

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

Naive Bayes assumption:

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

which gives

$$\text{Naive Bayes classifier: } v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

## Example

Consider *PlayTennis* again, and new instance

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

Want to compute:

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

(based on Table 3.2 of textbook)

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

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

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

20

## Bayesian Belief Networks

Interesting because:

- Naive Bayes assumption of conditional independence too restrictive
- But it's intractable without some such assumptions...
- Bayesian Belief networks describe conditional independence among *subsets* of variables

→ allows combining prior knowledge about (in)dependencies among variables with observed training data

(also called Bayesian Networks)

21

## Conditional Independence

**Definition:**  $X$  is *conditionally independent* of  $Y$  given  $Z$  if the probability distribution governing  $X$  is independent of the value of  $Y$  given the value of  $Z$ ; that is, if

$$(\forall x_i, y_j, z_k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

more compactly, we write

$$P(X|Y, Z) = P(X|Z)$$

Example: *Thunder* is conditionally independent of *Rain*, given *Lightning*

$$P(\text{Thunder} | \text{Rain}, \text{Lightning}) = P(\text{Thunder} | \text{Lightning})$$

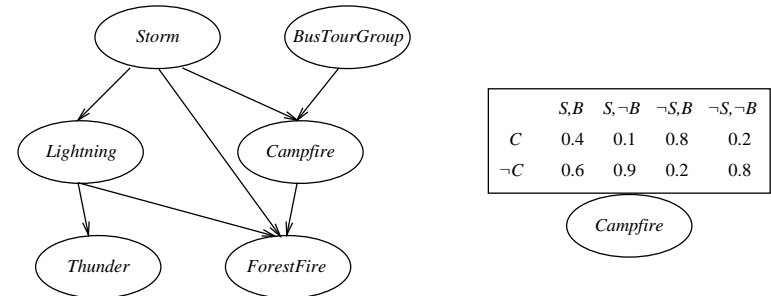
22

Naive Bayes uses cond. indep. to justify

$$\begin{aligned} P(X, Y | Z) &= P(X | Y, Z) P(Y | Z) \\ &= P(X | Z) P(Y | Z) \end{aligned}$$

23

## An Example of a Bayesian Network

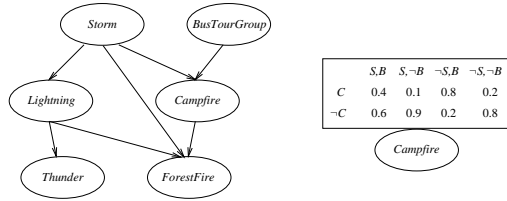


Network represents a set of conditional independence assertions:

- Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors.
- Directed acyclic graph

24

## Bayesian Networks (cont'd)



Represents joint probability distribution over all variables

- e.g.,  $P(\text{Storm}, \text{BusTourGroup}, \dots, \text{ForestFire})$
- in general,

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | \text{Parents}(Y_i))$$

where  $\text{Parents}(Y_i)$  denotes immediate predecessors of  $Y_i$  in graph

- so, joint distribution is fully defined by graph, plus the  $P(y_i | \text{Parents}(Y_i))$

25

## Learning of Bayesian Networks

Several variants of this learning task

- Network structure might be *known* or *unknown*
- Training examples might provide values of *all* network variables, or just *some*

If structure known and observe all variables

- Then it's easy as training a Naive Bayes classifier

## Inference in Bayesian Networks

How can one infer the (probabilities of) values of one or more network variables, given observed values of others?

- Bayes net contains all information needed for this inference
- If only one variable with unknown value, easy to infer it
- In general case, problem is NP hard

In practice, can succeed in many cases

- Exact inference methods work well for some network structures
- Monte Carlo methods “simulate” the network randomly to calculate approximate solutions

26

## Learning Bayesian Networks (cont'd)

Suppose structure known, variables partially observable

e.g., observe *ForestFire*, *Storm*, *BusTourGroup*, *Thunder*, but not *Lightning*, *Campfire*...

- Similar to training neural network with hidden units
- In fact, can learn network conditional probability tables using gradient ascent!
- Converge to network  $h$  that (locally) maximizes  $P(D|h)$

## Gradient Ascent for Bayesian Networks

Let  $w_{ijk}$  denote one entry in the conditional probability table for variable  $Y_i$  in the network

$$w_{ijk} = P(Y_i = y_{ij} | \text{Parents}(Y_i) = \text{the list } u_{ik} \text{ of values})$$

e.g., if  $Y_i = \text{Campfire}$ , then  $u_{ik}$  might be  $\langle \text{Storm} = T, \text{BusTourGroup} = F \rangle$

Perform gradient ascent by repeatedly

1. update all  $w_{ijk}$  using training data  $D$

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik} | d)}{w_{ijk}}$$

2. then, renormalize the  $w_{ijk}$  to assure

- $\sum_j w_{ijk} = 1$
- $0 \leq w_{ijk} \leq 1$