Statistical Learning

Bùi Tiến Lên

01/09/2019

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Contents

Notation

symbol	meaning
$a, b, c, N \ldots$	scalar number
$\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \ldots$	column vector
$\boldsymbol{X}, \boldsymbol{Y} \ldots$	matrix
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{N}$	set of natural numbers
$\mathbb{R}^D$	set of vectors
$\mathcal{X}, \mathcal{Y}, \ldots$	set
$\mathcal{A}$	algorithm

operator	meaning
$\boldsymbol{w}^\mathsf{T}$	transpose
$\boldsymbol{X}\boldsymbol{Y}$	matrix multiplication
$\boldsymbol{X}^{-1}$	inverse

Probability And Statistics

Bayes Theorem

$$P(h \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid h)P(h)}{P(\mathcal{D})} \tag{1}$$

- $P(h)$ = **prior probability** of hypothesis h
- $P(\mathcal{D})$ = prior probability of training data $\mathcal{D}$
- $P(h \mid \mathcal{D})$ = probability of h given $\mathcal{D}$ (called **posterior probability**)
- $P(\mathcal{D} \mid h)$ = probability of $\mathcal{D}$ given h (called **likelihood**)

Basic Formulas for Probabilities

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

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

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

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

Example 1

- We have the joint distribution of three random variables $P(Toothache, Cavity, Catch)$

	$toothache$		$\neg\ toothache$	
	$catch$	$\neg\ catch$	$catch$	$\neg\ catch$
$cavity$	**.108**	**.012**	**.072**	**.008**
$\neg\ cavity$	**.016**	**.064**	**.144**	**.576**

Example 1 (cont.)

	toothache		$\neg$ *toothache*	
	catch	$\neg$ *catch*	*catch*	$\neg$ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
$\neg$ *cavity*	**.016**	**.064**	**.144**	**.576**

$$
\begin{aligned}
P(toothache) &= 0.108 + 0.012 + 0.016 + 0.064 \\
&= 0.2
\end{aligned}
$$

Example 1 (cont.)

	toothache		$\neg$ *toothache*	
	catch	$\neg$ *catch*	*catch*	$\neg$ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
$\neg$ *cavity*	**.016**	**.064**	**.144**	**.576**

$$
\begin{aligned}
P(cavity \lor toothache) &= 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 \\
&= 0.28
\end{aligned}
$$

Example 1 (cont.)

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
¬ *cavity*	**.016**	**.064**	**.144**	**.576**

$$\begin{aligned} P(\neg cavity \mid toothache) &= \frac{P(\neg cavity \wedge toothache)}{P(toothache)} \\ &= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} \\ &= 0.4 \end{aligned}$$

Example 1 (cont.)

Problem

Let $\boldsymbol{x}$ be all the variables. We want the posterior joint distribution of the **query variables** $\boldsymbol{y}$ given specific values $\boldsymbol{v}_e$ for the **evidence variables** $\boldsymbol{e}$

Solution

- General idea: compute distribution on query variable by fixing **evidence variables** and summing over **hidden variables**
- Let the **hidden variables** be $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{y} - \boldsymbol{e}$ and denominator can be viewed as a **normalization constant** α

$$P(\boldsymbol{y} \mid \boldsymbol{e} = \boldsymbol{v}_e) = \alpha\, P(\boldsymbol{y}, \boldsymbol{e} = \boldsymbol{v}_e) = \alpha \sum_{\boldsymbol{h}} P(\boldsymbol{y}, \boldsymbol{e} = \boldsymbol{v}_e, \boldsymbol{h} = \boldsymbol{v}_h)$$

■

Example 1 (cont.)

	toothache		$\neg$ *toothache*	
	catch	$\neg$ *catch*	*catch*	$\neg$ *catch*
cavity	$\boxed{.108}$	$\boxed{.012}$	.072	.008
$\neg$ *cavity*	$\boxed{.016}$	$\boxed{.064}$	.144	.576

$$
\begin{aligned}
& P(Cavity \mid toothache) \\
= \; & \alpha\, P(Cavity, toothache) \\
= \; & \alpha\, [P(Cavity, toothache, catch) + P(Cavity, toothache, \neg catch)] \\
= \; & \alpha\, [\langle 0.108, 0.016 \rangle + \langle 0.012, 0.064 \rangle] \\
= \; & \alpha\, \langle 0.12, 0.08 \rangle \\
= \; & \langle 0.6, 0.4 \rangle
\end{aligned}
$$

Example 2

- 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."*

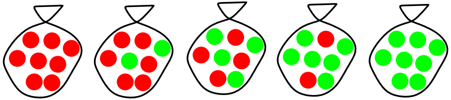
Solution

$$
\begin{aligned}
P(cancer) &= \underline{\hspace{3cm}} & P(\neg cancer) &= \underline{\hspace{3cm}} \\
P(\oplus \mid cancer) &= \underline{\hspace{3cm}} & P(\ominus \mid cancer) &= \underline{\hspace{3cm}} \\
P(\oplus \mid \neg cancer) &= \underline{\hspace{3cm}} & P(\ominus \mid \neg cancer) &= \underline{\hspace{3cm}}
\end{aligned}
$$

■

Example 3

- Suppose there are five kinds of bags of candies:
  - 10% are h_1: 100% cherry candies
  - 20% are h_2: 75% cherry candies + 25% lime candies
  - 40% are h_3: 50% cherry candies + 50% lime candies
  - 20% are h_4: 25% cherry candies + 75% lime candies
  - 10% are h_5: 100% lime candies



Experiment

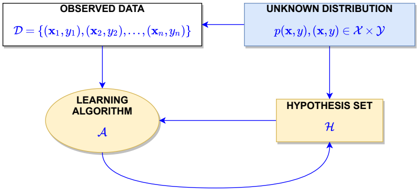
- Select one bag
- Candies drawn from the bag: $\mathcal{D} = \{$● ● ● ● ● ● ● ● ● ●$\}$

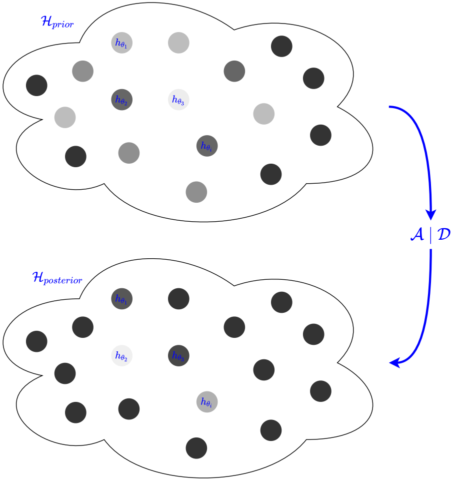
Question

1. What kind of bag is it?
2. What flavour will the next candy be?

Statiscal Learning

Components of Learning





Probabilistic Approach

Concept 1

Learning is an estimation of joint *probability density* functions given observed data $\mathcal{D}$.

- **Classification** and **Regression**: conditional density estimation

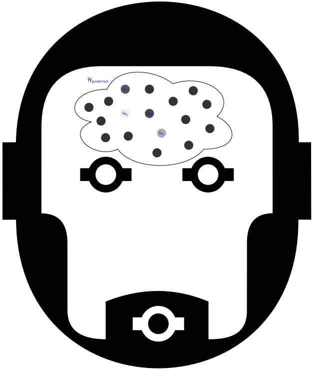
$$p(y \mid \boldsymbol{x}) \tag{2}$$

- **Unsupervised Learning**: density estimation

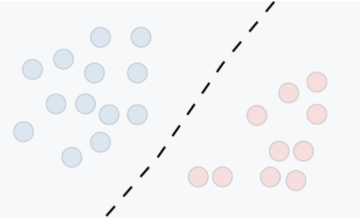
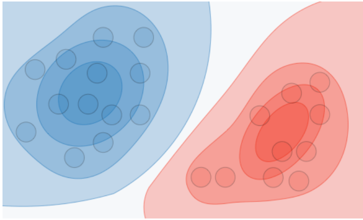
$$p(\boldsymbol{x}) \tag{3}$$

- **Inductive bias** is expressed as prior assumptions about these joint distributions.

Probabilistic Approach (cont.)



Type of Supervised Model

	Discriminative model	**Generative model**
Goal	• Directly estimate $P(y \mid \boldsymbol{x})$	• Estimate $P(\boldsymbol{x} \mid y)$ to then deduce $P(y \mid \boldsymbol{x})$
What's learned	• Decision boundary	• Probability distributions of the data
		

Bayesian Learning Fundamentals

Concept 2

Bayesian learning is a process that updates of a probability distribution over the **hypothesis space** $\mathcal{H} = \{h_1, h_2, \ldots\}$ given samples $\mathcal{D}$.

- Prior probability of each hypothesis h_i

$$P(h_i) \tag{4}$$

- Given the data $\mathcal{D}$, each hypothesis has a posterior probability (update)

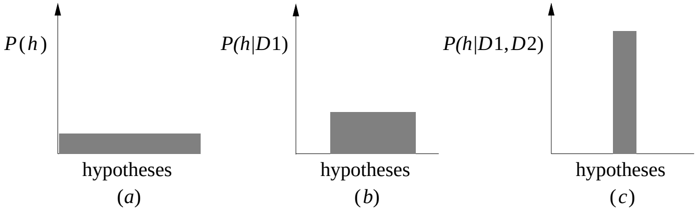
$$P(h_i \mid \mathcal{D}) = \alpha P(\mathcal{D} \mid h_i) P(h_i) \tag{5}$$

- Predictions use a average over the hypotheses

$$P(y) = \sum_i P(y \mid h_i) P(h_i) \tag{6}$$

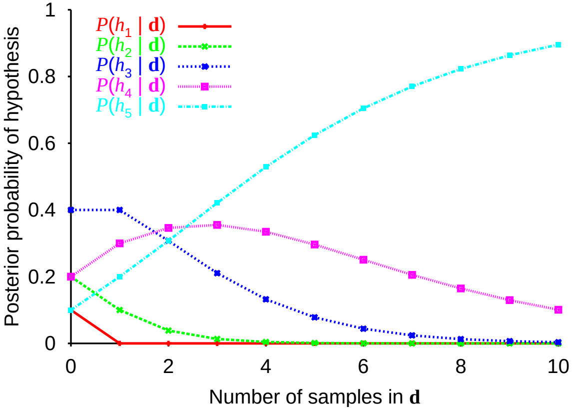
Evolution of Posterior Probabilities

- Changes of a probability distribution $P(h)$ after observing the data D_1 and D_2



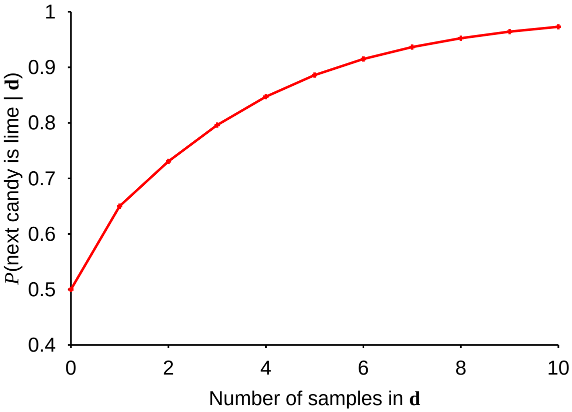
Example 3 Revisited

1. What kind of bag is it?
   Posterior probability of hypotheses



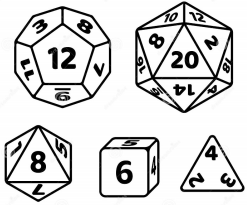
Example 3 Revisited (cont.)

2. What flavour will the next candy be?
   Prediction probability



Exercise

- Suppose we have a box of dice that contains a 4-sided die, a 6-sided die, an 8-sided die, a 12-sided die, and a 20-sided die



Experiment

- We select one die
- We roll the die a few more times and get $\mathcal{D} = \{6, 8, 7, 7, 5, 4\}$

Question

1. What die is selected?

Learning Strategy

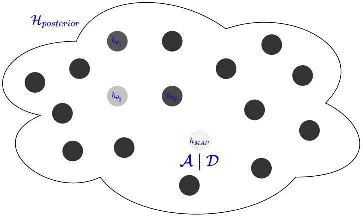
- **Updating** or **summing** over the hypothesis space is often intractable (e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)
- Alternative strategies:
  - **Maximum a posteriori** (MAP) learning
  - **Maximum likelihood** (ML) learning

Learning Strategy (cont.)

Maximum a posteriori learning

Given data $\mathcal{D}$

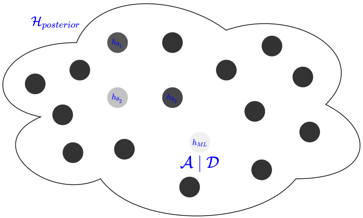
- choose hypothesis h (called h_{MAP}) maximizing $P(h \mid \mathcal{D})$
- i.e., maximize $P(\mathcal{D} \mid h)P(h)$ or $\log P(\mathcal{D} \mid h) + \log P(h)$



Learning Strategy (cont.)

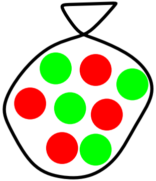
For large data sets, prior becomes weak or irrelevant, **maximum likelihood** learning

> Given data $\mathcal{D}$
>
> - choose hypothesis h (called h_{ML}) maximizing $P(\mathcal{D} \mid h)$ or $\log P(\mathcal{D} \mid h)$



Example 4

- A bag has a fraction θ of cherry candies?



Experiment

- Candies drawn from the bag: $\mathcal{D} = \{$  $\}$

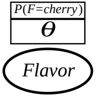
Question

1. What θ is it?

Example 4 (cont.)

ML learning solution

- Bayes net with one parameter θ



- Any θ is possible: continuum of hypotheses h_θ
- θ is a **parameter** for this simple (**binomial**) family of models
- Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes. These are **i.i.d.** (independent, identically distributed) observations, so

$$P(\mathcal{D} \mid h_\theta) = \prod_{j=1}^{N} P(d_j \mid h_\theta) = \theta^c \cdot (1-\theta)^\ell$$

Example 4 (cont.)

- Maximize this w.r.t. θ—which is easier for the **log-likelihood**:

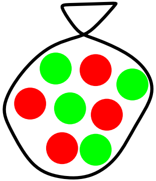
$$
\begin{aligned}
L(\mathcal{D} \mid h_\theta) &= \log P(\mathcal{D} \mid h_\theta) \\
&= \sum_{j=1}^{N} \log P(d_j \mid h_\theta) \\
&= c \log \theta + \ell \log(1-\theta) \\
\frac{dL(\mathcal{D} \mid h_\theta)}{d\theta} &= \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \\
\Longrightarrow \theta &= \frac{c}{c+\ell} = \frac{c}{N}
\end{aligned}
$$

- Seems sensible, but causes problems with 0 counts!

■

Example 5

- A bag has a fraction θ of cherry candies, red/green wrapper depends probabilistically θ_1, θ_2 on flavor?



Experiment

- Candies drawn from some bag: $\mathcal{D} =\{$  $\}$

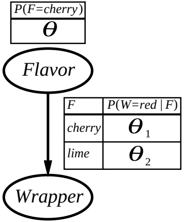
Question

1. What $\theta, \theta_1, \theta_2$ are they?

Example 5 (cont.)

ML learning solution

- Bayes net with one parameter $\theta, \theta_1, \theta_2$



Example 5 (cont.)

- Likelihood for, e.g., cherry candy in green wrapper:

$$
\begin{aligned}
& P(F = cherry, W = green \mid h_{\theta,\theta_1,\theta_2}) \\
&= P(F = cherry \mid h_{\theta,\theta_1,\theta_2}) P(W = green \mid F = cherry, h_{\theta,\theta_1,\theta_2}) \\
&= \theta \cdot (1 - \theta_1)
\end{aligned}
$$

- N candies, r_c red-wrapped cherry candies, etc.:

$$
P(\mathcal{D} \mid h_{\theta,\theta_1,\theta_2}) = \theta^c (1-\theta)^\ell \cdot \theta_1^{r_c} (1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell} (1-\theta_2)^{g_\ell}
$$

$$
\begin{aligned}
L = & [c \log \theta + \ell \log(1-\theta)] \\
& + [r_c \log \theta_1 + g_c \log(1-\theta_1)] \\
& + [r_\ell \log \theta_2 + g_\ell \log(1-\theta_2)]
\end{aligned}
$$

Example 5 (cont.)

- Derivatives of L contain only the relevant parameter:

$$\frac{\partial L}{\partial \theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \implies \theta = \frac{c}{c+\ell}$$

$$\frac{\partial L}{\partial \theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \implies \theta_1 = \frac{r_c}{r_c+g_c}$$

$$\frac{\partial L}{\partial \theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \implies \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}$$

- With **complete data**, *parameters can be learned separately*

■

Linear Regression Revisited

Linear Regression

- Unknown function f is modeled by the hypothesis $h_{\boldsymbol{w}}$

$$y = h_{\boldsymbol{w}}(x) = \boldsymbol{w}^{\top}\boldsymbol{x} + e \tag{7}$$

  where y is noisy target value, e is random variable (noise) drawn independently according to a Gaussian distribution with mean equal to 0 and variance equal to σ ($\mathcal{N}(0, \sigma^2)$)

- Probability language

$$p(y \mid \boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y \mid \boldsymbol{w}^{\top}\boldsymbol{x}, \sigma^2) \tag{8}$$

- Given data $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, the likelihood of $\mathcal{D}$ given $h_{\boldsymbol{w}}$ and noise

$$\begin{aligned} p(\mathcal{D} \mid h_{\boldsymbol{w}}, e) &= \prod_{i=1}^{N} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) \\ &= \prod_{i=1}^{N} \mathcal{N}(y_i \mid \boldsymbol{w}^{\top}\boldsymbol{x}, \sigma^2) \end{aligned}$$

ML Learning

Choose hypothesis h maximizing the likelihood

$$\begin{aligned}
& \arg\max_{\boldsymbol{w},\sigma} p(\mathcal{D} \mid h_{\boldsymbol{w}}, e) \\
\Leftrightarrow\quad & \arg\max_{\boldsymbol{w},\sigma} \prod_{i=1}^{N} \mathcal{N}(y_i \mid \boldsymbol{w}^\intercal \boldsymbol{x}_i, \sigma^2) \\
\Leftrightarrow\quad & \arg\max_{\boldsymbol{w},\sigma} \log\left(\prod_{i=1}^{N} \mathcal{N}(y_i \mid \boldsymbol{w}^\intercal \boldsymbol{x}_i, \sigma^2)\right) \\
\Leftrightarrow\quad & \arg\min_{\boldsymbol{w},\sigma} \frac{1}{2\sigma^2} MSE + \frac{N}{2}\log(\sigma^2) + \frac{N}{2}\log(2\pi)
\end{aligned} \tag{9}$$

Solving (9), we have

$$\boldsymbol{w}_{ML} = (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y}$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{w}_{ML}^\intercal \boldsymbol{x}_i - y_i)^2$$

Example

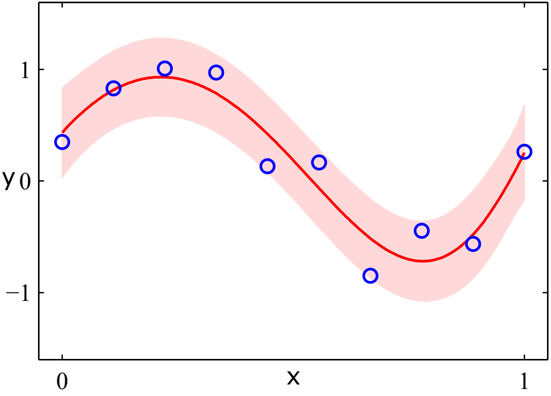


Figure 1: The red curve denotes the regression curve and the red region corresponds to $\pm\sigma$ standard deviation

Naive Bayes

When to use

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

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

Successful applications

- Diagnosis
- Classifying text documents

Naive Bayes

- Assume target function for classification problem

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

where each instance $\boldsymbol{x}$ described by attributes $(x_1, x_2 \dots x_D)$ and y is a corresponding class.

- **Naive Bayes assumption**

$$P(x_1, x_2 \dots x_D \mid y) = \prod_{i=1}^{D} P(x_i \mid y) \tag{10}$$

- **Naive Bayes classifier**

$$y_{NB} = \arg\max_{y} \hat{P}(y) \prod_{i=1}^{D} \hat{P}(x_i \mid y) \tag{11}$$

Naive Bayes classifiers

The form of the class-conditional density depends on the type of each feature.

1. In the case of real-valued features, we can use the **Gaussian distribution**:

$$p(\boldsymbol{x} \mid y = c) \sim \prod_{j=1}^{D} \mathcal{N}(x_j \mid \mu_{jc}, \sigma_{jc}^2)$$

   where μ_{jc} is the mean of feature j in objects of class c, and σ_{jc}^2 is its variance.

2. In the case of binary features, $x_j \in \{0, 1\}$, we can use the **Bernoulli distribution**:

$$p(\boldsymbol{x} \mid y = c) \sim \prod_{j=1}^{D} \mathrm{Ber}(x_j \mid \mu_{jc})$$

   where μ_{jc} is the probability that feature j occurs in class c.

Naive Bayes classifiers (cont.)

3. In the case of categorical features, $x_j \in \{v_{j_1}, v_{j_2}, \cdots, v_{j_K}\}$, we can use the **multinoulli distribution**:

$$p(\boldsymbol{x} \mid y = c) \sim \prod_{j=1}^{D} \mathrm{Cat}(x_j \mid \boldsymbol{\mu}_{jc})$$

where $\boldsymbol{\mu}_{jc}$ is a histogram over the K possible values for x_j in class c.

Naive Bayes Algorithm

```
NaiveBayesLearn(D)
  For each target value (class) y_j
    P̂(y_j) ← estimate P(y_j) given data D
    For each attribute x_i
      P̂(x_i | y_j) ← estimate P(x_i | y_j) given data D
```

```
ClassifyNewInstance(x)
  y = argmax_y P̂(y) ∏_{i=1}^{D} P̂(x_i | y)
  return y
```

Naive Bayes Algorithm (cont.)

Maximum likelihood learning for $\hat{P}(y = c)$ and $\hat{P}(x_i = a \mid y = c)$

$$\hat{P}(y = c) \leftarrow \frac{n_c}{n} \tag{12}$$

$$\hat{P}(x_i = a \mid y = c) \leftarrow \frac{n_a}{n_c} \tag{13}$$

where

- n is number of training examples
- n_c is number of training examples for which $y = c$
- n_a is number of examples for which $y = c$ and $x_i = a$

Example

- Consider *PlayTennis* again

$\mathcal{D} =$

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

Example (cont.)

$\hat{P}(PlayTennis)$

Yes	9/14
No	5/14

$\hat{P}(Outlook \mid PlayTennis)$

		Outlook		
		Overcast	Rain	Sunny
PlayTennis	Yes	4/9	3/9	2/9
	No	0/5	2/5	3/5

$\hat{P}(Temperature \mid PlayTennis)$

		Temperature		
		Cool	Hot	Mild
PlayTennis	Yes	3/9	2/9	4/9
	No	1/5	2/5	2/5

$\hat{P}(Humidity \mid PlayTennis)$

		Humidity	
		High	Normal
PlayTennis	Yes	3/9	6/9
	No	4/5	1/5

$\hat{P}(Wind \mid PlayTennis)$

		Wind	
		Strong	Weak
PlayTennis	Yes	3/9	6/9
	No	3/5	2/5

Example (cont.)

- Get new instance

$$\boldsymbol{x} = (Outlk = sun, Temp = cool, Humid = high, Wind = strong)$$

- Compute

$$\hat{P}(Yes) \times \hat{P}(sun \mid Yes) \times \hat{P}(cool \mid Yes) \times \hat{P}(high \mid Yes) \times \hat{P}(strong \mid Yes) = .005$$
$$\hat{P}(No) \times \hat{P}(sun \mid No) \times \hat{P}(cool \mid No) \times \hat{P}(high \mid No) \times \hat{P}(strong \mid No) = .021$$

- Make decison

$$y = No$$

Word Example

- Find Naive Bayes classifier given the following training datasets

#	Vị	Màu	Vỏ	Độc tính
1	Ngọt	Đỏ	Nhẵn	Không
2	Cay	Đỏ	Nhẵn	Không
3	Chua	Vàng	Có gai	Không
4	Cay	Vàng	Có gai	Có
5	Ngọt	Tím	Có gai	Không
6	Chua	Vàng	Nhẵn	Không
7	Ngọt	Tím	Nhẵn	Không
8	Cay	Tím	Có gai	Có
9	Cay	Vàng	Có gai	Không

Avoiding the zero-probability problem

1. Conditional independence assumption is often violated but it works surprisingly well anyway
2. Suppose that none of the training instances with target value y have attribute value $x_i = v$? then $\hat{P}(x_i = v \mid y) = 0$, and $\hat{P}(y) \ldots \hat{P}(x_i = v \mid y) \cdots = 0$ (not good in probability language)

Avoiding the zero-probability problem (cont.)

Typical solution is **Bayesian estimate** for $\hat{P}(y = c)$ and $\hat{P}(x_i = a \mid y = c)$

$$\hat{P}(y = c) \leftarrow \frac{n_c + 1}{n + C} \tag{14}$$

$$\hat{P}(x_i = a \mid y = c) \leftarrow \frac{n_a + 1}{n_c + r} \tag{15}$$

where

- n is number of training examples
- n_c is number of training examples for which $y = c$
- C is the number of classes
- n_a is number of examples for which $y = c$ and $x_i = a$
- r is the number of values of attribute x_i

Bayesian Networks

Bayesian networks

Concept 3

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax

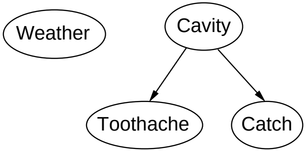
- A set of nodes, one per variable
- A directed, acyclic graph (DAG) (link $\approx$ "directly influences")
- A conditional distribution for each node given its parents:

$$\boldsymbol{P}(X_i \mid parents(X_i))$$

- In the simplest case, conditional distribution represented as a **conditional probability table** (CPT) giving the distribution over X_i for each combination of parent values

Example

- Topology of network encodes conditional independence assertions:



  - *Weather* is independent of the other variables
  - *Toothache* and *Catch* are conditionally independent given *Cavity*

Example (cont.)

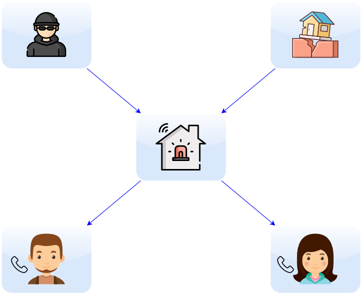
Problem

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

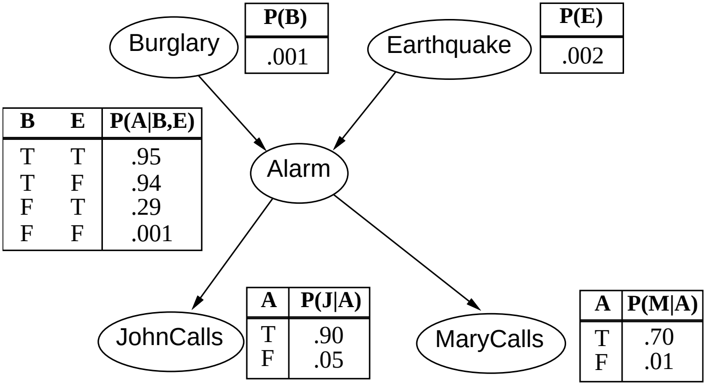
- Variables: *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call



Graphical Model



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

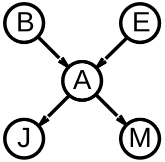
Type of Variables

$$X \rightarrow Y$$

- X, Y are random or not
- X, Y are observed or hidden
- X, Y are continuous or discrete (boolean, category)

CPT Representation

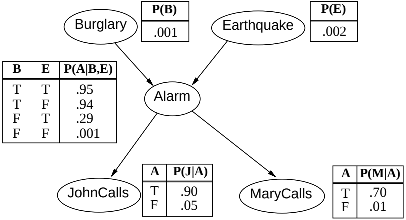
- A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values
- Each row requires one number p for $X_i = true$ (the number for $X_i = false$ is just $1 - p$)
- If each variable has no more than k parents, the complete network requires $O(n \times 2^k)$ numbers
  - I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution
- For **burglary net**, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)



Global semantics and Inference

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$



- For example,

$$\begin{aligned} P(j \land m \land a \land \neg b \land \neg e) &= P(j \mid a)P(m \mid a)P(a \mid \neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.00063 \end{aligned}$$

The Learning Problem

	Known Structure	**Unknown Structure**
Complete Data	Statistical parametric estimation (closed-form)	Discrete optimization over structures (discrete search)
Incomplete Data	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models...)

Learning problem includes

- **Parameter learning**
- **Structure learning**

Known Structure and Complete Data

- Given a training data $\mathcal{D}$, find the best parameter θs for multinomial variables

$$P_\theta(X_i \mid pa_i) \tag{16}$$

where $pa_i = parents(X_i)$ (pa_i can be $\emptyset$)

- Estimate parameter
Maximum likelihood

$$\hat{\theta}_{ML} = \frac{count(x_i, pa_i)}{count(pa_i)} \tag{17}$$

Maximum a posteriori

$$\hat{\theta}_{MAP} = \frac{\alpha(x_i, pa_i) + count(x_i, pa_i)}{\alpha(pa_i) + count(pa_i)} \tag{18}$$

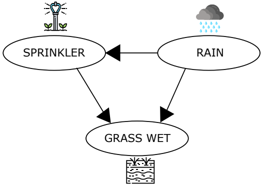
where $Count(.)$ is the number of instances

Example

- Find the best parameter θ given the following training data $\mathcal{D}$

#	Rain	Sprinkler	Grass Wet
1	T	T	T
2	T	T	F
3	T	F	T
4	T	F	F
5	F	T	T
6	F	T	F
7	F	F	T
8	F	F	F
9	T	T	T
10	F	T	T
11	T	F	T
12	F	F	T
13	F	T	T
14	T	T	T

RAIN	SPRINKLER T	SPRINKLER F
F	?	?
T	?	?



RAIN T	RAIN F
?	?

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	?	?
F	T	?	?
T	F	?	?
T	T	?	?

Unknown Structure and Complete Data

- Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that

$$\mathbf{P}(X_i \mid parents(X_i)) = \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$$

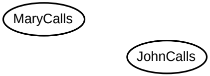
This choice of parents guarantees the global semantics:

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i \mid parents(X_i)) \quad \text{(by construction)}\end{aligned}$$

Example: Burglary alarm

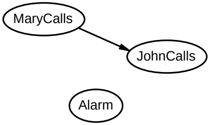
- Suppose we choose the ordering M, J, A, B, E

$$P(J \mid M) = P(J)?$$



Example: Burglary alarm (cont.)

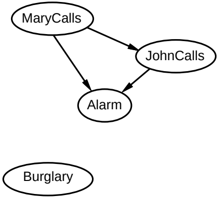
- Suppose we choose the ordering M, J, A, B, E



$$P(J \mid M) = P(J)? \text{ No}$$
$$P(A \mid J, M) = P(A \mid J)? \; P(A \mid J, M) = P(A)?$$

Example: Burglary alarm (cont.)

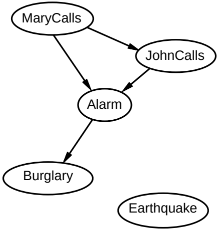
- Suppose we choose the ordering M, J, A, B, E



$P(J \mid M) = P(J)$? No
$P(A \mid J, M) = P(A \mid J)$? $P(A \mid J, M) = P(A)$? No
$P(B \mid A, J, M) = P(B \mid A)$?
$P(B \mid A, J, M) = P(B)$?

Example: Burglary alarm (cont.)

- Suppose we choose the ordering M, J, A, B, E



$P(J \mid M) = P(J)$? No

$P(A \mid J, M) = P(A \mid J)$? $P(A \mid J, M) = P(A)$? No

$P(B \mid A, J, M) = P(B \mid A)$? Yes

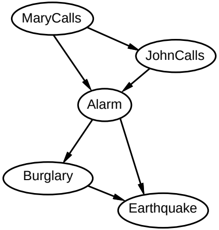
$P(B \mid A, J, M) = P(B)$? No

$P(E \mid B, A, J, M) = P(E \mid A)$?

$P(E \mid B, A, J, M) = P(E \mid A, B)$?

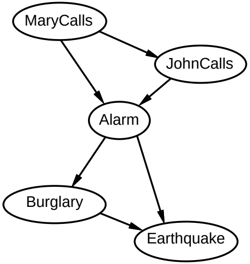
Example: Burglary alarm (cont.)

- Suppose we choose the ordering M, J, A, B, E



$P(J \mid M) = P(J)$? No
$P(A \mid J, M) = P(A \mid J)$? $P(A \mid J, M) = P(A)$? No
$P(B \mid A, J, M) = P(B \mid A)$? Yes
$P(B \mid A, J, M) = P(B)$? No
$P(E \mid B, A, J, M) = P(E \mid A)$? No
$P(E \mid B, A, J, M) = P(E \mid A, B)$? Yes

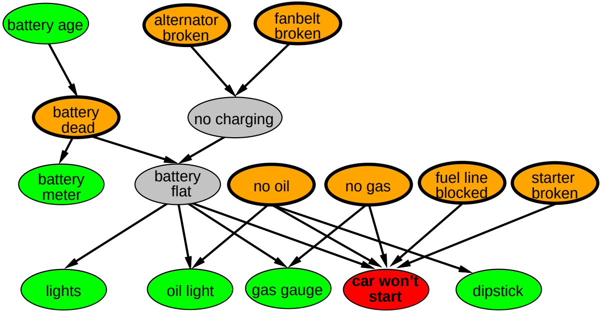
Example: Burglary alarm (cont.)



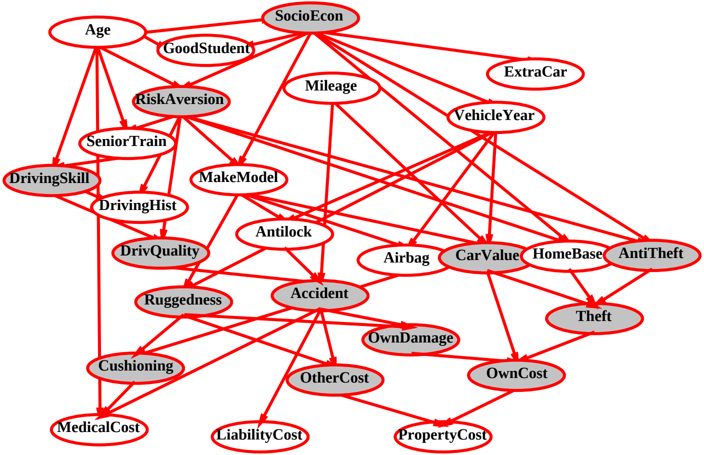
- Deciding conditional independence is hard in noncausal directions (Causal models and conditional independence seem hardwired for humans!)
- Assessing conditional probabilities is hard in noncausal directions
- Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

Example: Car diagnosis

- Initial evidence (red): car won't start
- Testable variables (green), "broken, so fix it" variables (orange)
- Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car insurance



Compact conditional distributions

Problem

- CPT grows exponentially with number of parents
- CPT becomes infinite with continuous-valued parent or child

Solution: **canonical** distributions that are defined compactly

- **Deterministic** nodes are the simplest case:

$$X = f(parents(X)) \text{ for some function } f$$

  - Boolean functions

$$NorthAmerican \equiv Canadian \vee US \vee Mexican$$

  - Numerical relationships among continuous variables

$$\frac{\partial Level}{\partial t} = \text{ inflow } + \text{ precipitation - outflow - evaporation}$$

Compact conditional distributions (cont.)

- **Noisy-OR** distributions model multiple noninteracting causes
  1. Parents $U_1 \ldots U_k$ include all causes (can add **leak node**)
  2. Independent failure probability q_i for each cause alone

$$P(X \mid U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$$

Number of parameters *linear* in number of parents

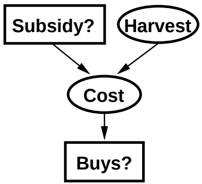
Compact conditional distributions (cont.)

$$
\begin{aligned}
q_{cold} &= P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6 \\
q_{flu} &= P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2 \\
q_{malaria} &= P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1
\end{aligned}
$$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
F	F	F	**0.0**	1.0
F	F	T	0.9	**0.1**
F	T	F	0.8	**0.2**
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Bayesian nets with continuous variables

Discrete (*Subsidy*? and *Buys*?); continuous (*Harvest* and *Cost*)



- Option 1: discretization – possibly large errors, large CPTs
- Option 2: finitely parameterized canonical families
  1. Continuous variable, discrete+continuous parents (e.g., *Cost*)
  2. Discrete variable, continuous parents (e.g., *Buys*?)

Continuous child variables

- Need one **conditional density** function for child variable given continuous parents, for each possible assignment to discrete parents
- Most common is the **linear Gaussian** (LG) model, e.g.,:

$$P(Cost = c \mid Harvest = h, Subsidy? = true)$$
$$= N(a_t h + b_t, \sigma_t)(c) = \frac{1}{\sigma_t\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{c-(a_t h + b_t)}{\sigma_t}\right)^2\right)$$

- Mean *Cost* varies linearly with *Harvest*, variance is fixed
- Linear variation is unreasonable over the full range but works OK if the *likely* range of *Harvest* is narrow

Continuous child variables (cont.)

- All-continuous network with LG distributions $\Longrightarrow$ full joint distribution is a multivariate Gaussian
- Discrete+continuous LG network is a **conditional Gaussian** network i.e., a multivariate Gaussian over all continuous variables for each combination of discrete variable values

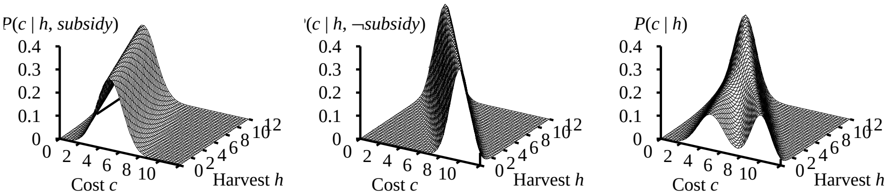
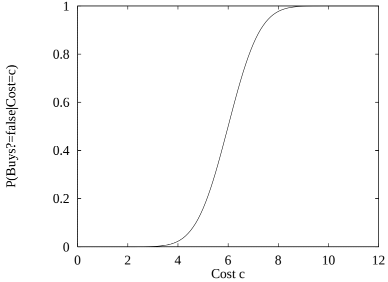


Figure 2: The graphs in (1) and (2) show the probability distribution over *Cost* as a function of *Harvest* size, with *Subsidy* true and false, respectively. Graph (3) shows the distribution $P(Cost \mid Harvest)$, obtained by summing over the two subsidy cases.

Discrete variable given continuous parents

- Probability of *Buys*? given *Cost* should be a "soft" threshold:



- **Probit** distribution uses integral of Gaussian:

$$\Phi(x) = \int_{-\infty}^{x} N(0,1)(x)dx$$

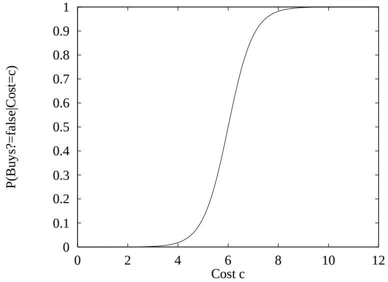
$$P(Buys? = true \mid Cost = c) = \Phi((-c+\mu)/\sigma)$$

Discrete variable given continuous parents

- **Sigmoid** (or **logit**) distribution also used in neural networks:

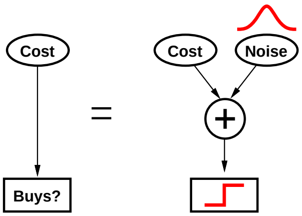
$$P(Buys? = true \mid Cost = c) = \frac{1}{1 + exp(-2\frac{-c+\mu}{\sigma})}$$

- Sigmoid has similar shape to probit but much longer tails:



Why the probit?

1. It's sort of the right shape
2. Can view as hard threshold whose location is subject to noise



References

Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

Lê, B. and Tô, V. (2014).
Cở sở trí tuệ nhân tạo.
Nhà xuất bản Khoa học và Kỹ thuật.

Nguyen, T. (2018).
Artificial intelligence slides.
Technical report, HCMC University of Sciences.

Russell, S. and Norvig, P. (2016).
Artificial intelligence: a modern approach.
Pearson Education Limited.