Statistical Learning

Bùi Tiến Lên

2023

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
$\mathcal{D}, \mathcal{X}, \mathcal{Y}, \ldots$	set
$\mathcal{A}$	algorithm

symbol	meaning
$X, Y \ldots$	random variable
$\boldsymbol{X}, \boldsymbol{Y} \ldots$	multivariate random variable
$x, y \ldots$	value
$\boldsymbol{x}, \boldsymbol{y} \ldots$	vector
p, pr, P, Pr	probability

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 observed 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

Given two random variables X and Y, we say that

- The quantity $p(X, Y)$ is a *joint probability*
- The quantity $p(Y \mid X)$ is a *conditional probability*
- The quantity $p(X)$ is a *marginal probability*

- **Sum rule**:

$$p(X) = \sum_{y} p(X, Y) \tag{2}$$

or

$$p(X) = \int p(X, Y)dy \tag{3}$$

- **Product rule**:

$$p(X, Y) = p(Y \mid X)p(X) \tag{4}$$

Example 1

- We have the joint distribution of three random variables $P(Headache, Hurt, Cold)$

Probabilistic database

Headache	*Hurt*	*Cold*	probability
$headache$	$hurt$	$cold$	.108
$headache$	$hurt$	$\neg cold$	.012
$headache$	$\neg hurt$	$cold$	.016
$headache$	$\neg hurt$	$\neg cold$	.064
$\neg headache$	$hurt$	$cold$	.072
$\neg headache$	$hurt$	$\neg cold$	.008
$\neg headache$	$\neg hurt$	$cold$	.144
$\neg headache$	$\neg hurt$	$\neg cold$	.576

query

users

Example 1 (cont.)

Headache	*Hurt*	*Cold*	probability
headache	*hurt*	*cold*	.108
headache	*hurt*	$\neg$*cold*	.012
headache	$\neg$*hurt*	*cold*	.016
headache	$\neg$*hurt*	$\neg$*cold*	.064
$\neg$*headache*	*hurt*	*cold*	.072
$\neg$*headache*	*hurt*	$\neg$*cold*	.008
$\neg$*headache*	$\neg$*hurt*	*cold*	.144
$\neg$*headache*	$\neg$*hurt*	$\neg$*cold*	.576

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

Example 1 (cont.)

Headache	*Hurt*	*Cold*	probability
headache	*hurt*	*cold*	.108
headache	*hurt*	$\neg$*cold*	.012
headache	$\neg$*hurt*	*cold*	.016
headache	$\neg$*hurt*	$\neg$*cold*	.064
$\neg$*headache*	*hurt*	*cold*	.072
$\neg$*headache*	*hurt*	$\neg$*cold*	.008
$\neg$*headache*	$\neg$*hurt*	*cold*	.144
$\neg$*headache*	$\neg$*hurt*	$\neg$*cold*	.576

$$
\begin{aligned}
P(\mathit{hurt} \vee \mathit{headache}) &= 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 \\
&= 0.28
\end{aligned}
$$

Example 1 (cont.)

Headache	*Hurt*	*Cold*	probability
headache	*hurt*	*cold*	.108
headache	*hurt*	¬*cold*	.012
headache	¬*hurt*	*cold*	.016
headache	¬*hurt*	¬*cold*	.064
¬*headache*	*hurt*	*cold*	.072
¬*headache*	*hurt*	¬*cold*	.008
¬*headache*	¬*hurt*	*cold*	.144
¬*headache*	¬*hurt*	¬*cold*	.576

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

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 $\boldsymbol{e}$ for the **evidence variables** $\boldsymbol{E}$

Summing 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

$$P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e}) = \alpha\, P(\boldsymbol{Y}, \boldsymbol{E} = \boldsymbol{e}) = \alpha \sum_{\boldsymbol{h}} P(\boldsymbol{Y}, \boldsymbol{E} = \boldsymbol{e}, \boldsymbol{H} = \boldsymbol{h}) \tag{5}$$

■

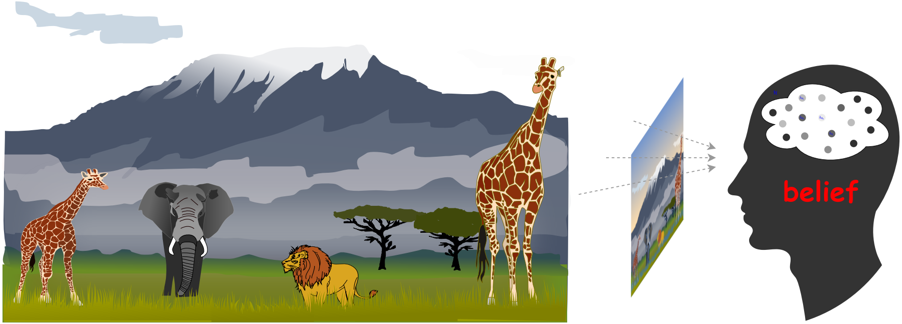
Example 1 (cont.)

Headache	*Hurt*	*Cold*	probability
headache	*hurt*	*cold*	.108
headache	*hurt*	$\neg$*cold*	.012
headache	$\neg$*hurt*	*cold*	.016
headache	$\neg$*hurt*	$\neg$*cold*	.064
$\neg$*headache*	*hurt*	*cold*	.072
$\neg$*headache*	*hurt*	$\neg$*cold*	.008
$\neg$*headache*	$\neg$*hurt*	*cold*	.144
$\neg$*headache*	$\neg$*hurt*	$\neg$*cold*	.576

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

Statistical Learning

Statistical Model



The world ⟶ The image ⟶ Model

- All models are wrong, but some are useful (**statistical model**) - George Box

Probabilistic Approach

Concept 1

Learning is an estimation of *joint probability density* function $p(\boldsymbol{x}, y)$ given observed data $\mathcal{D}$. **Inductive bias** is expressed as prior assumptions about these joint distributions.

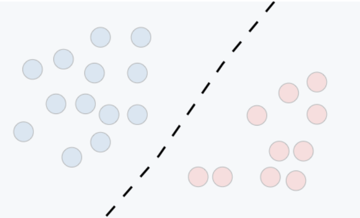
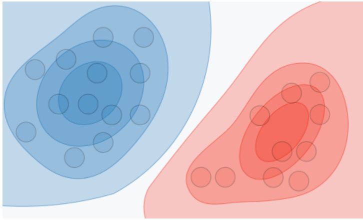
- **Classification** and **Regression**: conditional density estimation

$$p(y \mid \boldsymbol{x}) \tag{6}$$

- **Unsupervised Learning**: density estimation

$$p(\boldsymbol{x}) \tag{7}$$

Type of Supervised Model

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

Bayesian Learning

Concept 2

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

- Prior probability of each hypothesis h_i

$$P(h_i) \tag{8}$$

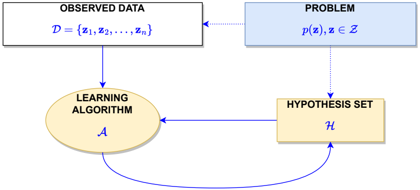
- Given the data $\mathcal{D}$, each hypothesis has a posterior probability (update)

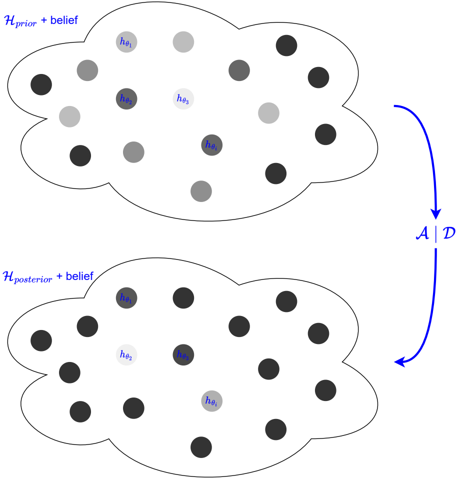
$$P(h_i \mid \mathcal{D}) = \alpha P(\mathcal{D} \mid h_i) P(h_i) \tag{9}$$

- Predictions use **an average** over the hypotheses

$$P(d) = \sum_i P(d \mid h_i) P(h_i) \tag{10}$$

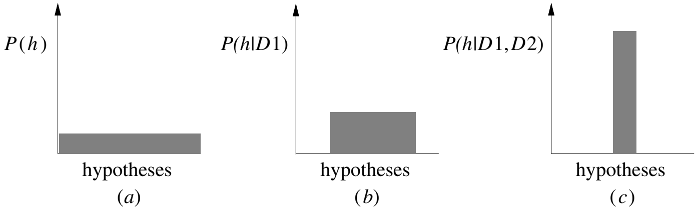
Components of Learning





Evolution of Posterior Probabilities

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



Example 1

- Does patient have COVID 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 COVID."*

Solution

prior

$$P(COVID) = \underline{0.008} \qquad P(\neg COVID) = \underline{0.992}$$

likelihood

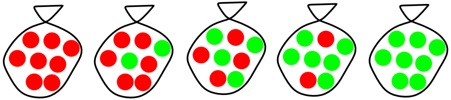
$$P(\oplus \mid COVID) = \underline{0.98} \qquad P(\ominus \mid COVID) = \underline{0.03}$$

$$P(\oplus \mid \neg COVID) = \underline{0.02} \qquad P(\ominus \mid \neg COVID) = \underline{0.97}$$

■

Example 2

- 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

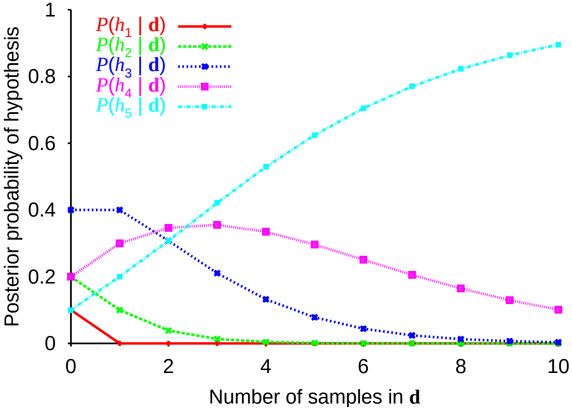
- Select one bag
- Candies drawn from the bag: $\mathcal{D} = \{$ $\}$

Question

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

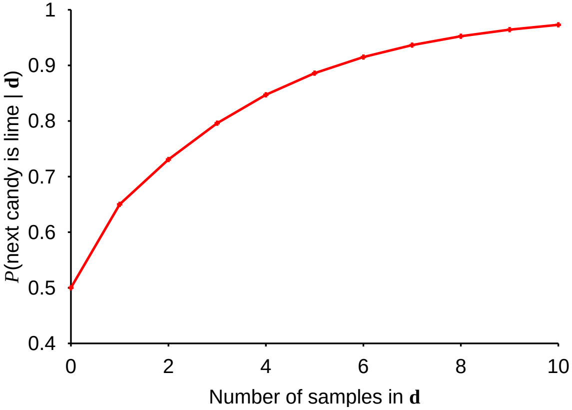
Example 2 (cont.)

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



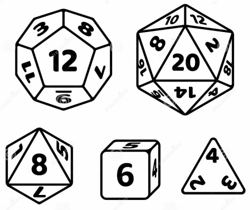
Example 2 (cont.)

1. 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?

Fitting Models

- One categorical variable
- Two categorical variables
- One continuous variable
- Joint Probability Distributions

Learning Strategy

Concept 3

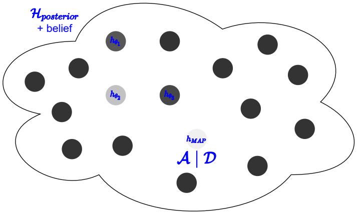
Fitting probability models to data $\mathcal{D} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_n\}$ is referred to as *learning* because we learn about the parameters θ of the model

- **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

Maximum a posteriori

Given data $\mathcal{D}$

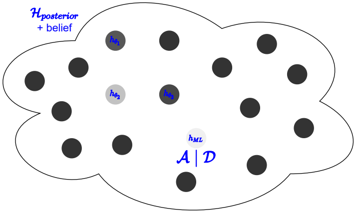
- choose hypothesis h (called h_{MAP}) maximizing $P(h \mid \mathcal{D})$
- i.e., maximize $P(\mathcal{D} \mid h)P(h)$ or $\log P(\mathcal{D} \mid h) + \log P(h)$



Maximum likelihood

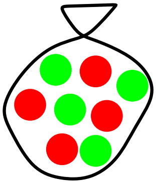
- For large data sets, prior becomes weak or irrelevant, **maximum likelihood learning**

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



Example 1

- A bag containing two kinds of candies, lime and cherry, has a fraction θ of cherry candies?



Experiment

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

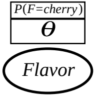
Question

1. What the value of θ is?

Example 1 (cont.)

ML learning solution

- The model (Bayes net) has one variable *flavor* with one parameter θ



- Any $\theta \in [0, 1]$ 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 1 (cont.)

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

$$
\begin{aligned}
L(\mathcal{D} \mid h_\theta) &= \log p(\mathcal{D} \mid h_\theta) \\
&= \sum_{j=1}^{N} \log p(d_j \mid h_\theta) \\
&= c \log \theta + \ell \log(1-\theta) \\
\frac{dL(\mathcal{D} \mid h_\theta)}{d\theta} &= \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \\
\Longrightarrow \theta_{ML} &= \frac{c}{c+\ell} = \frac{c}{N}
\end{aligned}
\tag{11}
$$

■

- ML seems sensible, but causes problems with **0 counts**!

Example 1 (cont.)

MAP learning solution

- Prior for h_θ

$$p(h_\theta) = Beta(a, b) \text{ where } a, b > 0,$$

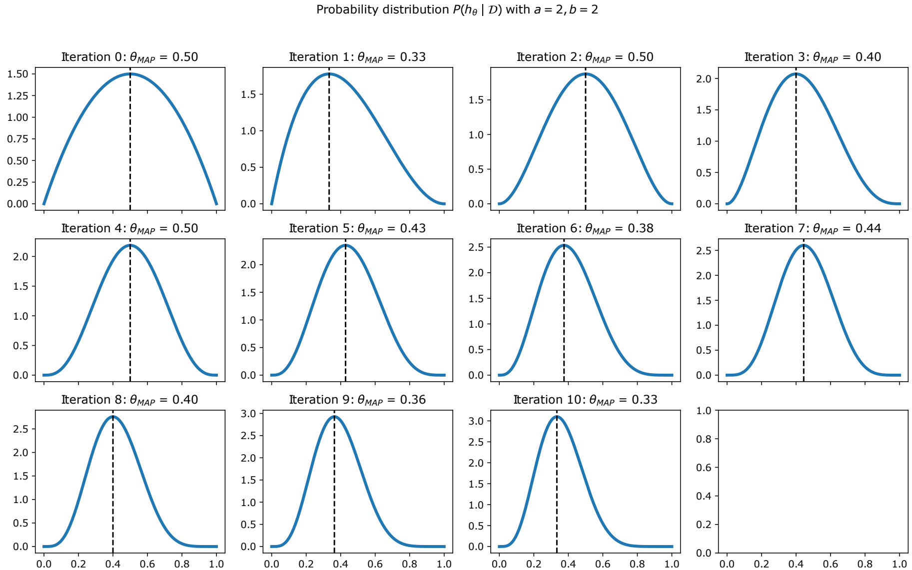
- Posteriori for h_θ given $\mathcal{D}$

$$\begin{aligned} p(h_\theta \mid \mathcal{D}) &\propto p(\mathcal{D} \mid h_\theta) p(h_\theta) \\ &= Beta(a + c, b + \ell) \end{aligned}$$

- MAP estimate

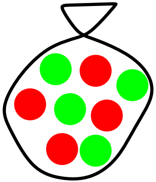
$$\theta_{MAP} = \frac{c + a - 1}{c + \ell + a + b - 2} \tag{12}$$

Example 1 (cont.)



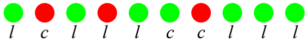
Example 2

- 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 $\mathcal{D}$



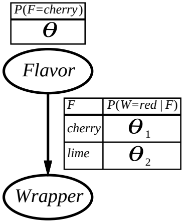
Question

1. What the values of $\theta, \theta_1, \theta_2$ are?

Example 2 (cont.)

ML learning solution

- The model (Bayes net) has two variables *flavor* and *wrapper* with three parameters $\theta, \theta_1, \theta_2$



Example 2 (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.:

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

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

Example 2 (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*

■

Example 3

- Suppose that the distribution of male height is a normal distribution $\mathcal{N}(\mu, \sigma^2)$. Given the following data $\mathcal{D}$

#	height (m)	#	height (m)
1	1.72	6	1.63
2	1.65	7	1.74
3	1.60	8	1.82
4	1.73	9	1.75
5	1.80	10	1.64

Question: what are the values of μ, σ?

Example 3 (cont.)

ML learning solution

- The model has one variable x (*height*) with two parameters μ, σ^2

$$p(x \mid \mu, \sigma^2) = \mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{13}$$

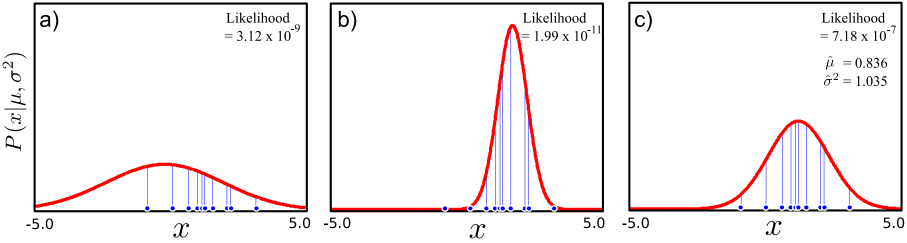
- The likelihood of model with the parameters $\{\mu, \sigma^2\}$ for observed i.d.d. data $\mathcal{D} = \{x_1, ..., x_n\}$ is

$$\begin{aligned} p(\mathcal{D} \mid \mu, \sigma^2) &= \prod_{i=1}^{n} \mathcal{N}(x_i \mid \mu, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{(x_i-\mu)^2}{\sigma^2}\right] \end{aligned} \tag{14}$$

Example 3 (cont.)

- The maximum likelihood solution μ_{ML}, σ^2_{ML}

$$\mu_{ML}, \sigma^2_{ML} = \operatorname{argmax}_{\mu,\sigma^2}[p(\mathcal{D} \mid \mu, \sigma^2)] \tag{15}$$



Example 3 (cont.)

- Since the logarithm is a monotonic function, the position of the maximum in the transformed function $L = \log p(\mathcal{D} \mid \mu, \sigma^2)$ remains the same as $p(\mathcal{D} \mid \mu, \sigma^2)$
- The maximum *log-likelihood* solution

$$\begin{aligned} \mu_{ML}, \sigma^2_{ML} &= \text{argmax}_{\mu,\sigma^2} [\log p(\mathcal{D} \mid \mu, \sigma^2)] \\ &= \text{argmax}_{\mu,\sigma^2} \left[-\frac{1}{2} n \log[2\pi] - \frac{1}{2} \log \sigma^2 - \frac{1}{2} \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^2} \right] \end{aligned} \tag{16}$$

Example 3 (cont.)

- To maximize, we differentiate this *log-likelihood* L with respect to μ and equate the result to zero

$$\begin{aligned}\frac{\partial L}{\partial \mu} &= \sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2} \\ &= \frac{\sum_{i=1}^{n} x_i}{\sigma^2} - \frac{n\mu}{\sigma^2} = 0 \\ \mu_{ML} &= \frac{1}{n}\sum_{i=1}^{n} x_i \end{aligned} \tag{17}$$

Example 3 (cont.)

- By a similar process, the expression for the variance can be shown to be

$$\sigma^2_{ML} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{ML})^2. \tag{18}$$

■

Joint Probability Distributions

- The joint probability distribution is central to probabilistic inference, because once we know the joint distribution we can answer every possible probabilistic question that can be asked about these variables.

Gender	HoursWorked	Wealth	probability
female	< 40.5	poor	0.2531
female	< 40.5	rich	0.0246
female	≥ 40.5	poor	0.0422
female	≥ 40.5	rich	0.0116
male	< 40.5	poor	0.3313
male	< 40.5	rich	0.0972
male	≥ 40.5	poor	0.1341
male	≥ 40.5	rich	0.1059

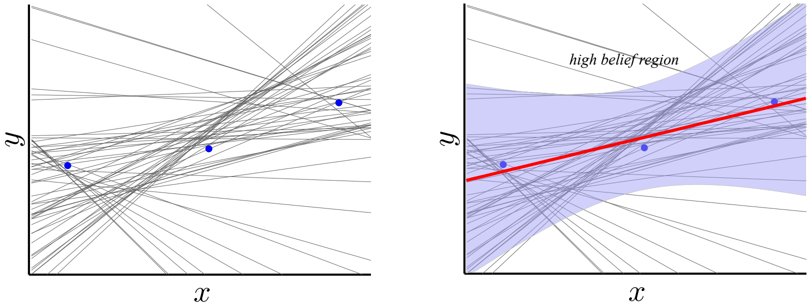
Discussion

- How can we learn joint distributions from observed training data $\mathcal{D}$?

Linear Regression Revisited

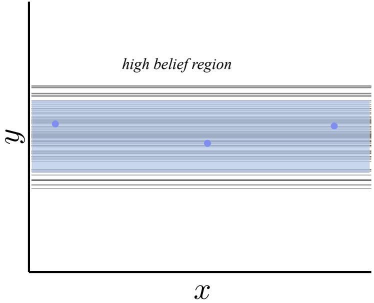
Linear Regression

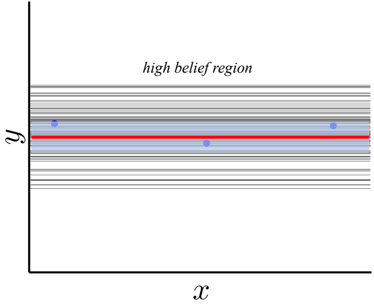
- Linear regression without prior belief



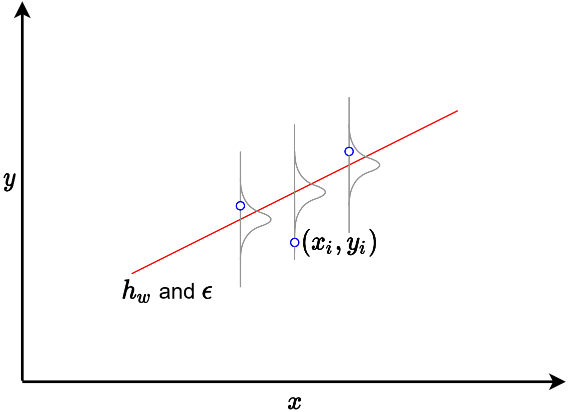
Linear Regression (cont.)

- Linear regression with prior belief





Linear Regression Revisited



Linear Regression Revisited (cont.)

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

$$y = h_{\mathbf{w}}(x) = \mathbf{w}^\intercal \mathbf{x} + \epsilon \tag{19}$$

where y is noisy target value, ϵ 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

$$p(y \mid \mathbf{x}, \mathbf{w}) = \mathcal{N}(y \mid \mathbf{w}^\intercal \mathbf{x}, \sigma^2) \tag{20}$$

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

$$\begin{aligned} p(\mathcal{D} \mid h_{\mathbf{w}}, \epsilon) &= \textstyle\prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \textstyle\prod_{i=1}^N \mathcal{N}(y_i \mid \mathbf{w}^\intercal \mathbf{x}, \sigma^2) \end{aligned} \tag{21}$$

ML Learning

Choose hypothesis h maximizing the likelihood

$$\begin{aligned} & \arg\max_{\mathbf{w},\sigma} p(\mathcal{D} \mid h_{\mathbf{w}}, e) \\ \Leftrightarrow \quad & \arg\max_{\mathbf{w},\sigma} \prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{w}^{\mathsf{T}}\mathbf{x}_i, \sigma^2) \\ \Leftrightarrow \quad & \arg\max_{\mathbf{w},\sigma} \log\left(\prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{w}^{\mathsf{T}}\mathbf{x}_i, \sigma^2)\right) \\ \Leftrightarrow \quad & \arg\min_{\mathbf{w},\sigma} \frac{1}{2\sigma^2} MSE + \frac{N}{2}\log(\sigma^2) + \frac{N}{2}\log(2\pi) \end{aligned} \tag{22}$$

Solving (20), we have

$$\mathbf{w}_{ML} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$$

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

Example

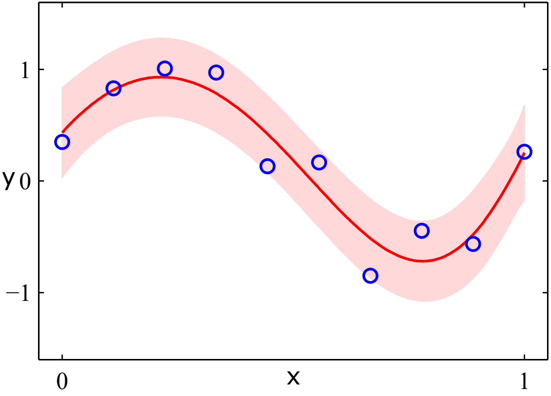
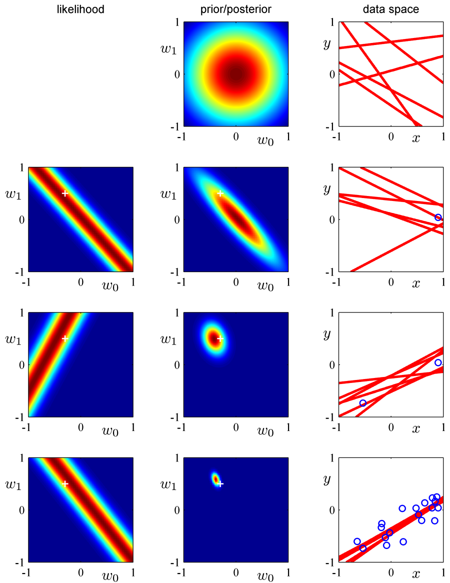


Figure 1: The red curve denotes the regression curve and the red region corresponds to $\pm\sigma$ standard deviation

Bayesian Learning



- The figure shows the update process of Bayesian learning where ***w*** are introduced as hypothesis parameters

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 assumption

- Assume target function for classification problem

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

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

- **Naive Bayes assumption**

$$P(x_1, x_2 \ldots x_D \mid y) = \prod_{i=1}^{D} P(x_i \mid y) \tag{23}$$

Naive Bayes classifiers

- **Naive Bayes classifier**

$$y_{NB} = \arg\max_{y} \hat{P}(y) \prod_{i=1}^{D} \hat{P}(x_i \mid y) \tag{24}$$

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.

Naive Bayes classifiers (cont.)

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

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

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

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**:

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

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

Naive Bayes Algorithm

$\textsc{LearnNaiveBayes}(\mathcal{D})$

for each target value (class) y_j

$\hat{P}(y_j) \leftarrow$ estimate $P(y_j)$ given data $\mathcal{D}$

for each attribute x_i

$\hat{P}(x_i \mid y_j) \leftarrow$ estimate $P(x_i \mid y_j)$ given data $\mathcal{D}$

$\textsc{ClassifyNewInstance}(x)$

$y = \arg\max_y \hat{P}(y) \prod_{i=1}^{D} \hat{P}(x_i \mid y)$

return y

Naive Bayes Algorithm (cont.)

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

$$\hat{P}(y = c) \leftarrow \frac{n_c}{n} \tag{25}$$

$$\hat{P}(x_i = a \mid y = c) \leftarrow \frac{n_a}{n_c} \tag{26}$$

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 1

- 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 1 (cont.)

- Naive Bayes model

$\hat{P}(PlayTennis)$

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 1 (cont.)

- Get new instance
  $\boldsymbol{x} = (Outlk = sun, Temp = cool, Humid = high, Wind = strong)$
- Compute
  $$\begin{cases} \hat{P}(yes) \times \hat{P}(sun \mid yes) \times \hat{P}(cool \mid yes) \times \hat{P}(high \mid yes) \times \hat{P}(strong \mid yes) & = .005 \\ \hat{P}(no) \times \hat{P}(sun \mid no) \times \hat{P}(cool \mid no) \times \hat{P}(high \mid no) \times \hat{P}(strong \mid no) & = .021 \end{cases}$$
- Make decison
  $y = no$

Example 2

- Find Naive Bayes classifier given the following training dataset

#	Vị	Màu	Vỏ	Độc tính
1	ngọt	đỏ	nhẵn	không
2	cay	đỏ	nhẵn	có
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	tím	có gai	không
10	cay	tím	có gai	có
11	cay	vàng	có gai	có

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 $\hat{P}(x_i = a \mid y = c)$

$$\hat{P}(y=c) \leftarrow \frac{n_c + 1}{n + C} \tag{27}$$

$$\hat{P}(x_i = a \mid y = c) \leftarrow \frac{n_a + 1}{n_c + r} \tag{28}$$

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

Example 3

- Find Naive Bayes classifier given the following training dataset

#	Height (m)	Hair	Gender
1	1.72	brown	male
2	1.65	black	female
3	1.60	black	female
4	1.73	black	female
5	1.80	brown	male
6	1.63	black	male
7	1.74	black	female
8	1.82	brown	male
9	1.75	black	male
10	1.64	brown	female

Statistical Language Model

Language Models

Concept 4

A **language model** is a function p_{LM} that takes an English (or any language) sentence and returns the probability that it was produced by an English speaker

Language Models (cont.)

Language models

- Answer the question: How likely is a string of English (or any language) words good English?
- Help with reordering

$$p_{\mathrm{LM}}(\text{the house is small}) > p_{\mathrm{LM}}(\text{small the is house})$$

- Help with word choice

$$p_{\mathrm{LM}}(\text{I am going home}) > p_{\mathrm{LM}}(\text{I am going house})$$

Language Models (cont.)

- **Given** a string of English words $W = w_1, w_2, w_3, ..., w_n$. **Question**: what is $p(W)$?
- Decomposing $p(W)$ using the chain rule:

$$p(w_1, w_2, w_3, ..., w_n) = p(w_1)\ p(w_2|w_1)\ p(w_3|w_1, w_2)...p(w_n|w_1, w_2, ...w_{n-1}) \tag{29}$$

- The language model probability $p(w_1, w_2, w_3, ..., w_n)$ is a product of word history probabilities given a **history** of preceding words.

Markov Assumption

Concept 5

Markov assumption states that only a limited number of previous words affect the probability of the next word.

- limited memory: only last k words are included in history (older words less relevant) $\rightarrow$ kth order Markov model

N-Gram Language Models

- Unigram (1-gram) model

$$p(w_1, w_2, w_3, \ldots, w_n) = p(w_1)\ p(w_2)\ p(w_3) \ldots p(w_n) \tag{30}$$

- Bigram (2-gram) model

$$p(w_1, w_2, w_3, \ldots, w_n) = p(w_1)\ p(w_2|w_1)\ p(w_3|w_2) \ldots p(w_n|w_{n-1}) \tag{31}$$

- Trigram (3-gram) model

$$p(w_1, w_2, w_3, \ldots, w_n) = p(w_1)\ p(w_2|w_1)\ p(w_3|w_1, w_2) \ldots p(w_n|w_{n-2}, w_{n-1}) \tag{32}$$

Estimating N-Gram Probabilities

- Maximum likelihood estimation for 1-gram

$$p_{\mathrm{LM}}(w_1) = \frac{\text{count}(w_1)}{\text{the total number of words}} \tag{33}$$

- Maximum likelihood estimation for 2-gram

$$p_{\mathrm{LM}}(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)} \tag{34}$$

- Maximum likelihood estimation for 3-gram

$$p_{\mathrm{LM}}(w_3|w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)} \tag{35}$$

Example 1

Given a corpus

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like ham </s>
```

Some of the bigram probabilities from this corpus

$$p_{\mathrm{LM}}(w_2|w_1) = \frac{\mathsf{count}(w_1, w_2)}{\mathsf{count}(w_1)}$$

$$p_{\mathrm{LM}}(\texttt{I}|\texttt{<s>}) = \frac{2}{3} = 0.67$$
$$p_{\mathrm{LM}}(\texttt{Sam}|\texttt{<s>}) = \frac{1}{3} = 0.33$$
$$p_{\mathrm{LM}}(\texttt{am}|\texttt{I}) = \frac{2}{3} = 0.67$$
$$p_{\mathrm{LM}}(\texttt{do}|\texttt{I}) = \frac{1}{3} = 0.33$$
$$p_{\mathrm{LM}}(\texttt{Sam}|\texttt{am}) = \frac{1}{2} = 0.5$$
$$p_{\mathrm{LM}}(\texttt{</s>}|\texttt{Sam}) = \frac{1}{2} = 0.5$$

Exercise 1

Given a corpus

`<s> Tôi là nam </s>`

`<s> Bạn tôi là nữ </s>`

`<s> Tôi thích trà </s>`

Compute all the bigram probabilities from this corpus

Example 2

- Bigram probabilities for eight words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

- And a few other useful probabilities

$$p_{\mathrm{LM}}(\texttt{i}|\texttt{<s>}) = 0.25 \qquad p_{\mathrm{LM}}(\texttt{english}|\texttt{want}) = 0.0011$$
$$p_{\mathrm{LM}}(\texttt{food}|\texttt{english}) = 0.5 \quad p_{\mathrm{LM}}(\texttt{</s>}|\texttt{food}) = 0.68$$

- Compute the probability of sentences like `I want English food` or `I want Chinese food`

Example 2

- The probability of sentence I want English food
  $\rightarrow$ <s> i want english food </s>

#	$p(w_2 \mid w_1)$	value
1	$p_{\text{LM}}(\texttt{i} \mid \texttt{<s>})$	0.25
2	$p_{\text{LM}}(\texttt{want} \mid \texttt{i})$	0.33
3	$p_{\text{LM}}(\texttt{english} \mid \texttt{want})$	0.0011
4	$p_{\text{LM}}(\texttt{food} \mid \texttt{english})$	0.5
5	$p_{\text{LM}}(\texttt{</s>} \mid \texttt{food})$	0.68
	total	0.000031

Practical Issues

We do everything in log space

- Avoid underflow
- Adding is faster than multiplying

$$\log(p_1 \times p_2 \times \ldots \times p_n) = \log(p_1) + \log(p_2) + \ldots + \log(p_n) \tag{36}$$

#	$(w_2 \mid w_1)$	$p_{\mathrm{LM}}(w_2 \mid w_1)$	$\log_2 p_{\mathrm{LM}}(w_2 \mid w_1)$
1	(i|<s>)	0.25	-2.0
2	(want|i)	0.33	-1.6
3	(english|want)	0.0011	-9.8
4	(food|english)	0.5	-1.0
5	(</s>|food)	0.68	-0.6
	total	0.000031	-15.0

Evaluation and Perplexity

- A good model assigns a string of words $W = w_1, w_2, w_3, ..., w_n$ a high probability
- There are various ways to measure this

$$\begin{aligned} L &= p(w_1, w_2, w_3, ..., w_n) && \text{(likelihood)} \\ H &= -\frac{1}{n}\log_2 L && \text{(per-word cross-entropy)} \\ PP &= 2^H && \text{(perplexity)} \end{aligned} \tag{37}$$

- Lower perplexity = better model

Text Generation

- Three sentences randomly generated from three *n*-gram models computed from 40 million words of the Wall Street Journal, lower-casing all characters and treating punctuation as words.

Unigram

Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

Bigram

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Trigram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Text Generation (cont.)

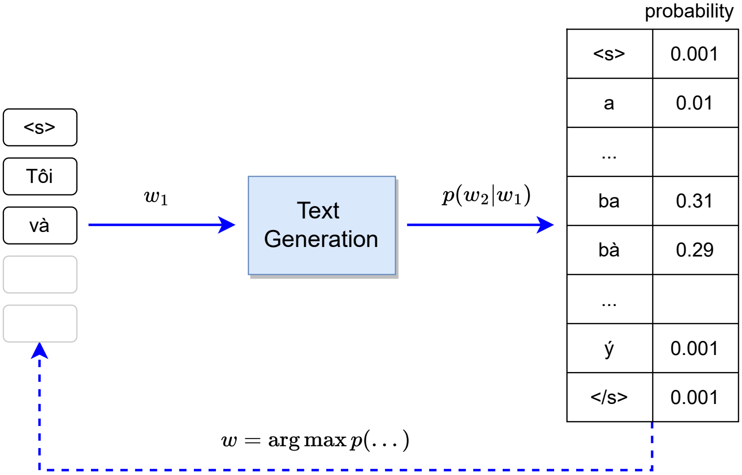
- The Shannon Game: How well can we predict the next word?

I always order pizza with cheese and ____?____

	probability
mushrooms	0.1
pepperoni	0.1
anchovies	0.01
...	
fried rice	0.0001
...	
and	1^{-100}
...	

Text Generation (cont.)

- Text generation model



Discussion

- But there are many more unseen *n*-grams than seen *n*-grams
- Example, Europarl 2-grams:
  - $86,700$ distinct words
  - $86,700^2 = 7,516,890,000$ possible 2-grams
  - but only about $30,000,000$ words (and 2-grams) in corpus
- Example, Vietnamese 3-grams:
  - $6,814$ distinct syllables
  - $6,814^3 = 316,378,081,144$ possible 3-grams
  - but only about $1,500,000$ 3-grams in corpus

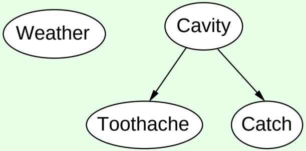
Bayesian Networks

- Introduction
- Representation
- Learning
- Examples
- More Representation

Bayesian networks

Concept 6

A **Bayesian network**, or **causal probabilistic network**, is a *directed acyclic graph* (DAG) for conditional independence assertions and hence for compact specification of full joint distributions



Bayesian networks (cont.)

Syntax

- Nodes represent **variables**
- Links represent **dependency relations** between variables and quantified by *conditional distributions*
- A **conditional distribution** for each node given its parents:

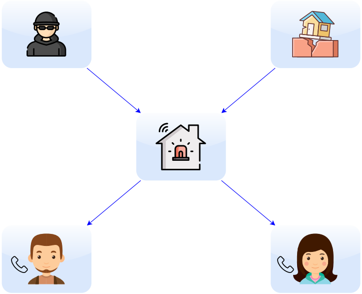
$$P(X_i \mid parents(X_i))$$

- 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

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: $\boldsymbol{B}urglar$, $\boldsymbol{E}arthquake$, $\boldsymbol{A}larm$, $\boldsymbol{J}ohnCalls$, $\boldsymbol{M}aryCalls$
- 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

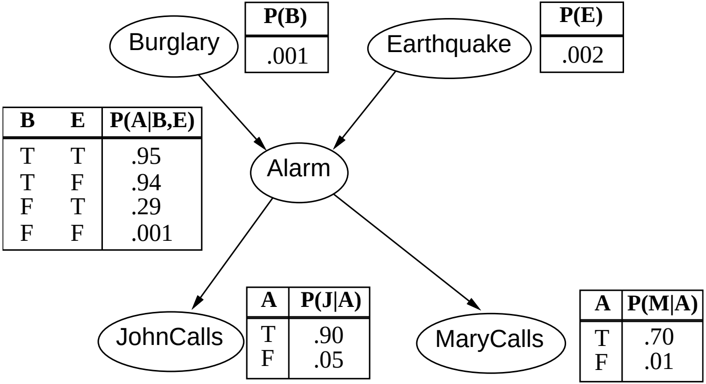


Queries

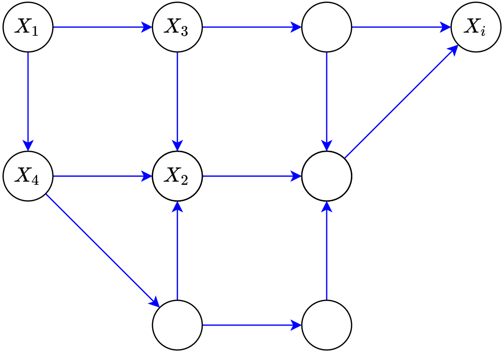
$P(Query \mid Evidence) = ?$

- **Diagnostic** (from effects to causes)
  $P(B \mid J) \rightarrow$ "*If John calls, how likely is the house burglarized?*"
- **Causal** (from causes to effects):
  $P(J \mid E) \rightarrow$ "*If earthquake happens, how likely will John make a call?*"
- **Intercausal** (between causes of a common effect):
  $P(B \mid A, E) \rightarrow$ "*If earthquake happens and alarm is on, how likely is the house burglarized??*"
- **Mixed**:
  $P(A \mid J, E) \rightarrow$ "..."
  $P(B \mid J, E) \rightarrow$ "..."

Graphical Model



Type of Variables

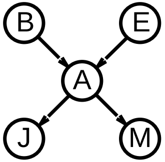


Variables $\{X_i\}$

- **random** or **deterministic**
- **observed** or **hidden**
- **continuous** or **discrete** (boolean, category)

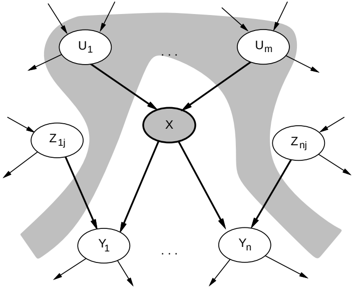
CPT Representation

- 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. $2^5 - 1 = 31$)

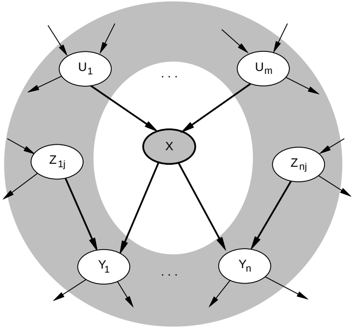


Conditional Independence

- Each node is conditionally independent of its *nondescendants* given its *parents*



- Each node is conditionally independent of all others given its **Markov blanket** (*parents* + *children* + *children's parents*)

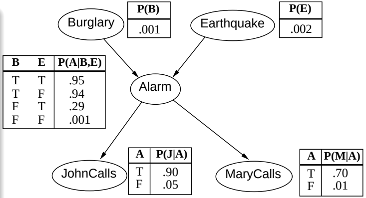


Global semantics and Inference

Concept 7

The **full joint distribution** is defined 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,

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

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

$$P_\theta(X_i \mid pa_i) \tag{38}$$

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

- Estimate parameter
  Maximum likelihood

$$\hat{\theta}_{ML} = \frac{count(x_i, pa_i)}{count(pa_i)} \tag{39}$$

  Maximum a posteriori

$$\hat{\theta}_{MAP} = \frac{\alpha(x_i, pa_i) + count(x_i, pa_i)}{\alpha(pa_i) + count(pa_i)} \tag{40}$$

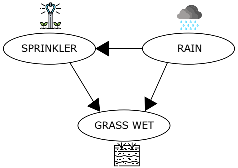
  where $count(.)$ is the number of instances and $\alpha(.)$ is the prior parameters.

Example 4

- Find the best parameter θs 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

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

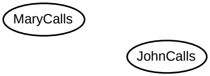
   This choice of parents guarantees the global semantics:

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

Example: Burglary alarm

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

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

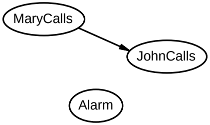


Example: Burglary alarm (cont.)

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

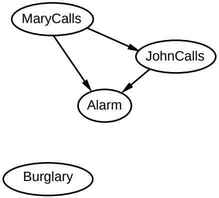
$P(J \mid M) = P(J)$? No

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



Example: Burglary alarm (cont.)

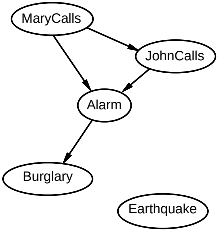
- Suppose we choose the ordering M, J, A, B, E



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

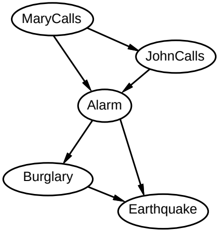
$P(B \mid A, J, M) = P(B)$? No

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

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

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

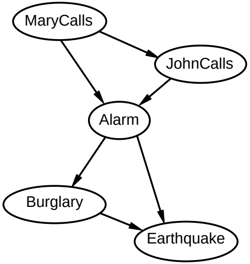
$P(B \mid A, J, M) = P(B \mid A)$? Yes

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

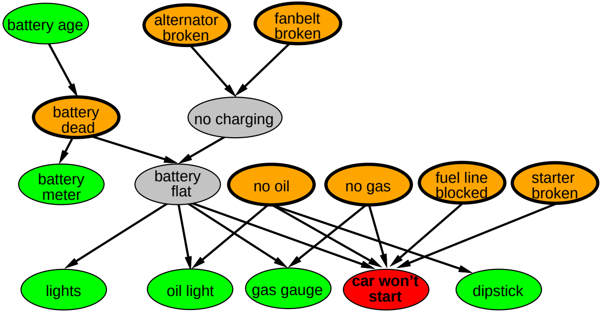
Example: Burglary alarm (cont.)



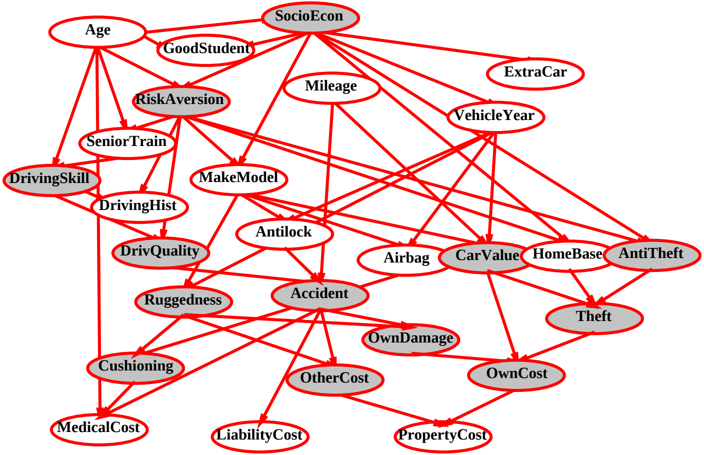
- Deciding conditional independence is hard in noncausal directions (Causal models and conditional independence seem hardwired for humans!)
- Assessing conditional probabilities is hard in noncausal directions
- Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

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



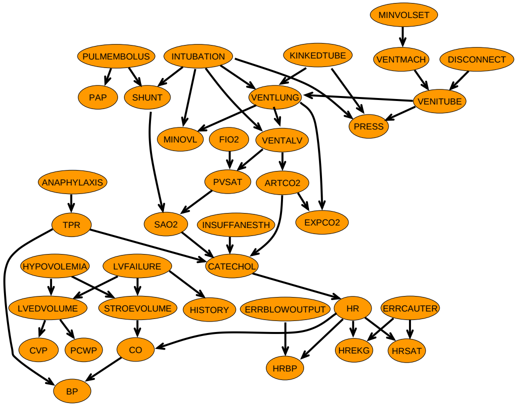
Example: Car insurance



Example: “ICU Alarm” network

Domain: Monitoring Intensive-Care Patients

- 37 variables
- 509 parameters



Compact conditional distributions

Problem

- CPT grows exponentially with number of parents
- CPT does not work with continuous-valued parent or child

- **Deterministic nodes** X:

$$X = f(parents(X)) \text{ for some function } f \tag{41}$$

  - Boolean functions

$$NorthAmerican = Canadian \vee US \vee Mexican$$

  - 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
  - Parents $U_1 \ldots U_k$ include all causes (can add **leak node**)
  - Independent failure probability q_i for each cause alone

$$P(X \mid U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i \tag{42}$$

Number of parameters *linear* in number of parents

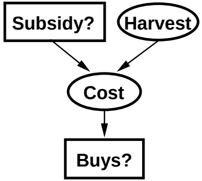
Compact conditional distributions (cont.)

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

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

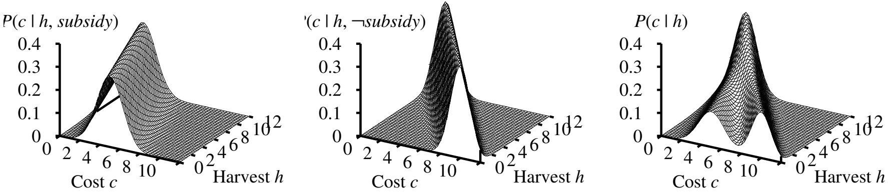
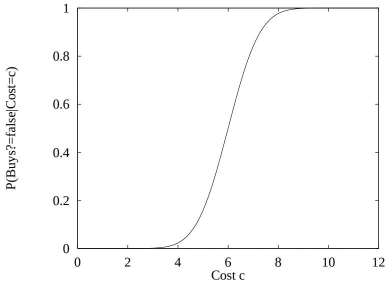


Figure 2: The graphs in (1) and (2) show the probability distribution over *Cost* as a function of *Harvest* size, with *Subsidy* true and false, respectively. Graph (3) shows the distribution $P(Cost \mid Harvest)$, obtained by summing over the two subsidy cases.

Discrete variable given continuous parents

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



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

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

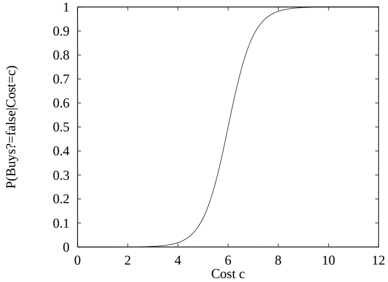
$$P(Buys? = true \mid Cost = c) = \Phi\left(\frac{-c+\mu}{\sigma}\right)$$

Discrete variable given continuous parents (cont.)

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

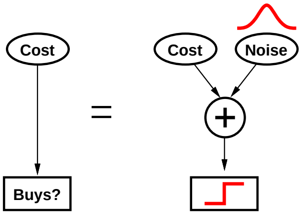
$$P(Buys? = true \mid Cost = c) = \frac{1}{1 + \exp(-2\frac{-c+\mu}{\sigma})}$$

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



Why the probit/logit?

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.

Russell, S. and Norvig, P. (2021).
Artificial intelligence: a modern approach.
Pearson Education Limited.