Chapter 6
Probability-based Learning
Part B

Prof. Chang-Chieh Cheng

Dept. Computer Science

National Chiao Tung University, Taiwan

Continuous Features

- Categorical feature ➔ Discrete random variable
  - $X = \{X_1, X_2, \dots, X_m\}$
  - $P(X_1) + P(X_2) + \cdots + P(X_m) = 1.0$
- Continuous feature ➔ Continuous random variable
  - $X \in \mathbf{R}$

$$P(a \le X \le b) = \int_a^b f(x)\, dx \le 1.0$$

$$P(X) = \int_{-\infty}^{\infty} f(x)\, dx = 1.0$$

Continuous Features

- **Probability density function** (PDF)
- If f is a PDF

$$\int_{-\infty}^{\infty} f(x)\, dx = 1.0$$

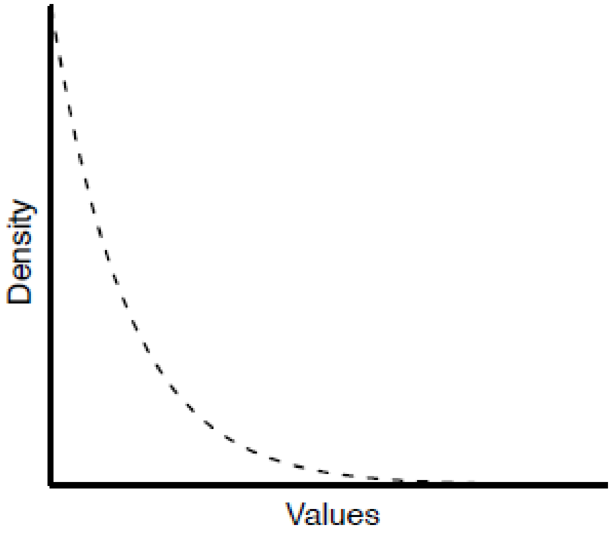
- A PDF can be used to represent the probability distribution of a continuous random variable.
- Using a PDF to fit a probability distribution
- Five standard PDFs
  - Exponential
  - Normal
  - Student-t
  - Mixture Gaussians
  - Gamma

Standard PDF

- Exponential

$$E(x, \lambda) = \lambda e^{-\lambda x} \text{ if } x > 0\text{, otherwise} = 0$$

$$x \in \mathbf{R}$$
$$\lambda \in \mathbf{R} \text{ and } \lambda > 0$$

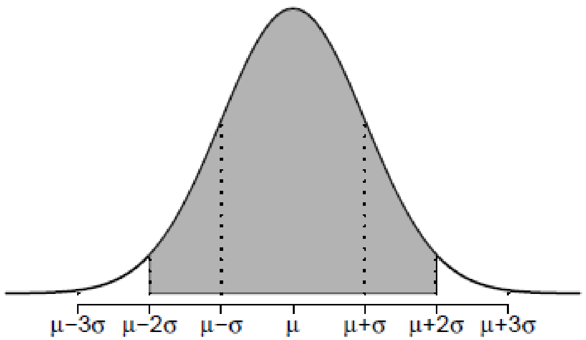


Standard PDF

- Normal distribution
  - Gaussian function

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$x \in \mathbf{R}$$
$$\mu \in \mathbf{R}$$
$$\sigma \in \mathbf{R} \text{ and } \sigma > 0$$

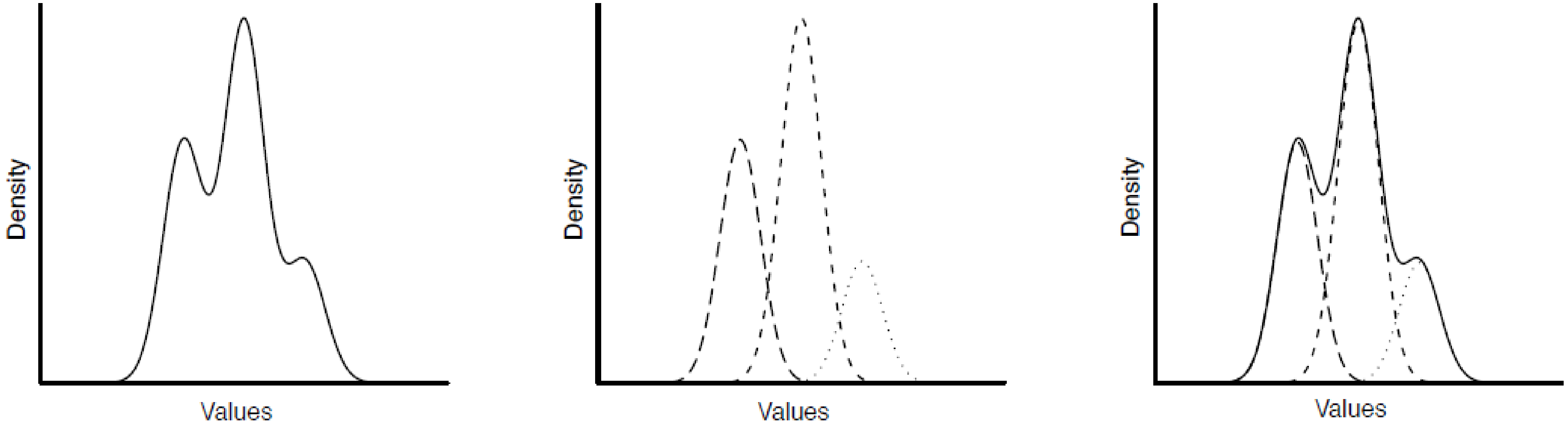


Standard PDF

- Mixture Gaussians

$$N(x, \mathbf{u}, \boldsymbol{\sigma}, \mathbf{w}) = \sum_{i=1}^{n} \frac{w_i}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2{\sigma_i}^2}}$$

$$x \in \mathbf{R}$$
$$\mathbf{u} = \{\mu_1, \mu_2, \dots, \mu_n | \mu_i \in \mathbf{R}\}$$
$$\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \dots, \sigma_n | \sigma_i \in \mathbf{R} > 0\}$$
$$\mathbf{w} = \{w_1, w_2, \dots, w_n | \, w_i \in \mathbf{R} > 0\}$$



Standard PDF

- Student-t

$$\tau(x,k) = \frac{\Gamma(\frac{k+1}{2})}{\sqrt{k\pi}\,\Gamma(\frac{k}{2})}\left(1+\frac{x^2}{k}\right)^{-\frac{k+1}{2}}$$

$x \in \mathbf{R}$

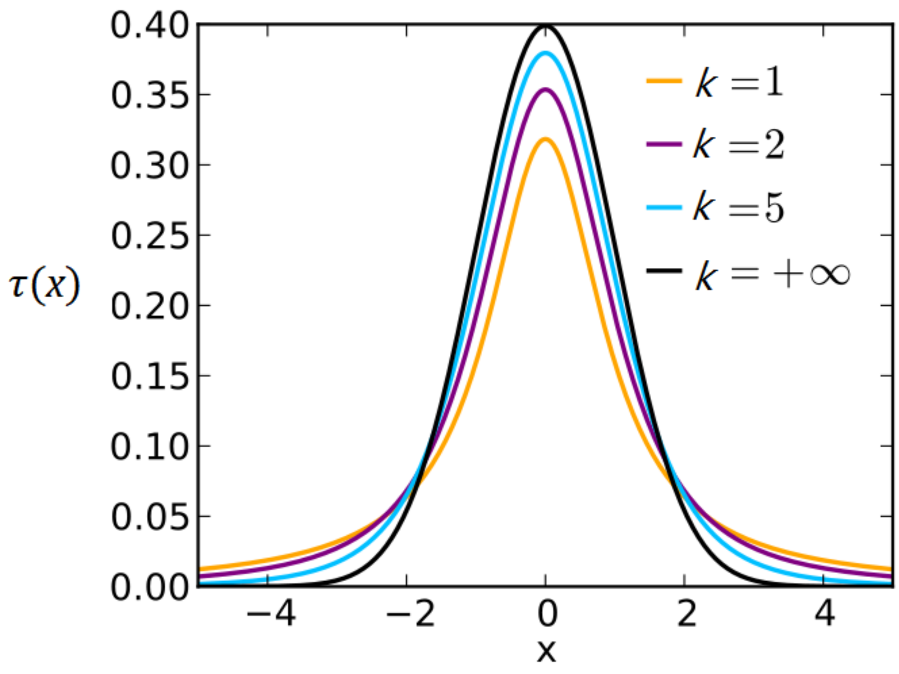
$k \in \mathbf{N}$ and $k > 0$

$$\Gamma(n) = (n-1)!$$

where $n \in \mathbf{N} > \mathbf{0}$

$$\Gamma(z) = \int_0^{\infty} x^{z-1}\, e^{-x} dx$$

where $z \in \mathbf{C} > \mathbf{and\ real(z)} > \mathbf{0}$



Standard PDF

- Student-t
  - if k is even

$$\frac{\Gamma(\frac{k+1}{2})}{\sqrt{k\pi}\Gamma(\frac{k}{2})} = \frac{(k-1)(k-3)\ldots 5\cdot 3}{2\sqrt{k}(k-2)(k-4)\ldots 4\cdot 2}$$

  - Otherwise

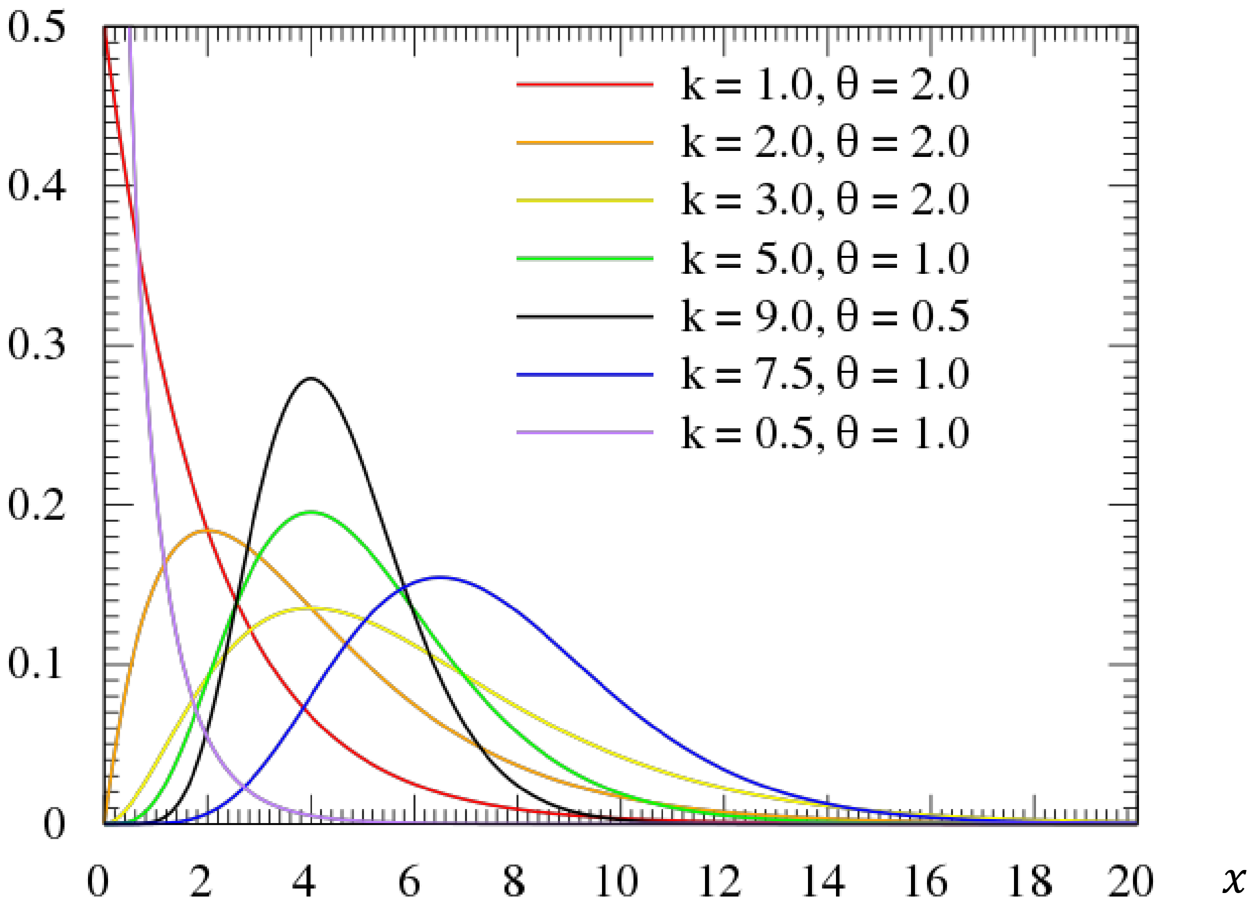
$$\frac{\Gamma(\frac{k+1}{2})}{\sqrt{k\pi}\Gamma(\frac{k}{2})} = \frac{(k-1)(k-3)\ldots 4\cdot 2}{\pi\sqrt{k}(k-2)(k-4)\ldots 5\cdot 3}$$

$$\begin{aligned}
\Gamma\left(-\tfrac{3}{2}\right) &= \tfrac{4}{3}\sqrt{\pi} \\
\Gamma\left(-\tfrac{1}{2}\right) &= -2\sqrt{\pi} \\
\Gamma\left(\tfrac{1}{2}\right) &= \sqrt{\pi} \\
\Gamma\left(\tfrac{3}{2}\right) &= \tfrac{1}{2}\sqrt{\pi} \\
\Gamma\left(\tfrac{5}{2}\right) &= \tfrac{3}{4}\sqrt{\pi} \\
\Gamma\left(\tfrac{7}{2}\right) &= \tfrac{15}{8}\sqrt{\pi}
\end{aligned}$$

Standard PDF

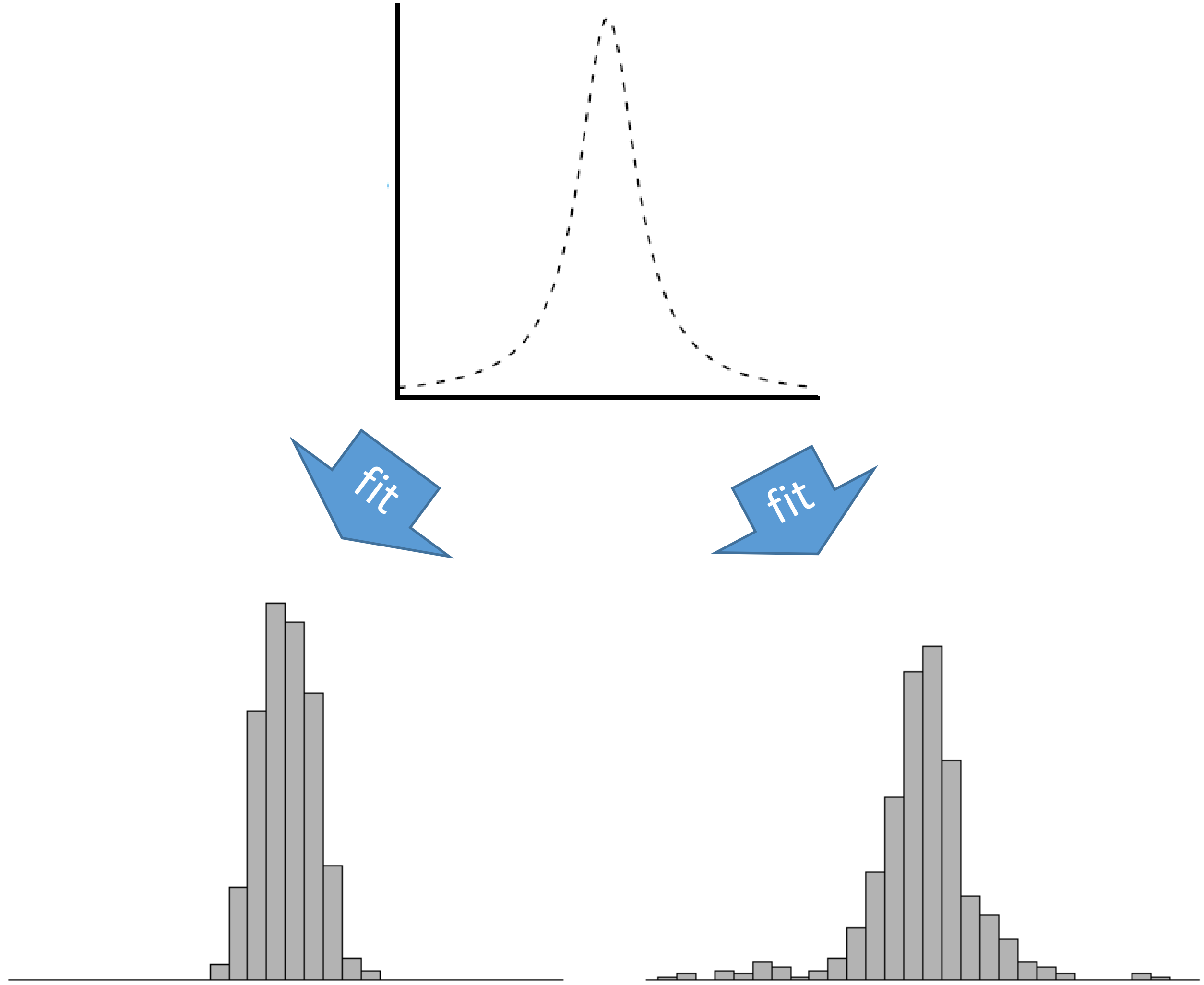
- Gamma distribution

$$G(x,k,\theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}$$



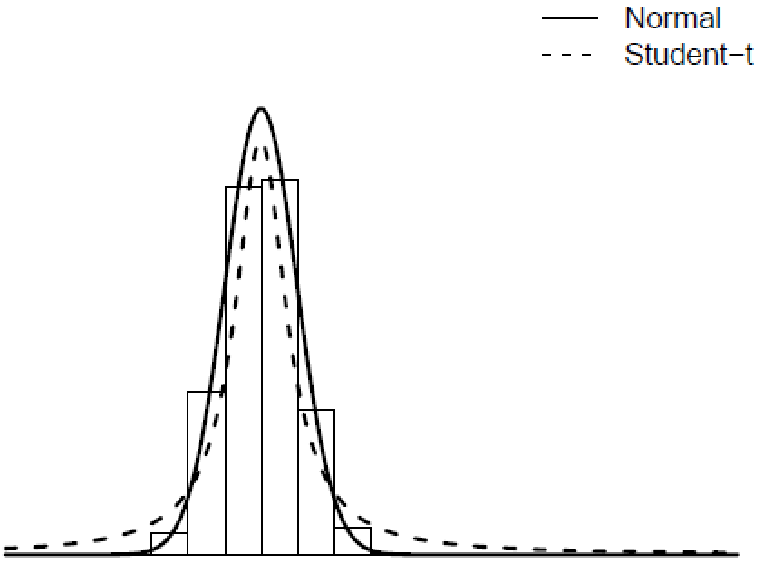
PDF Fitting

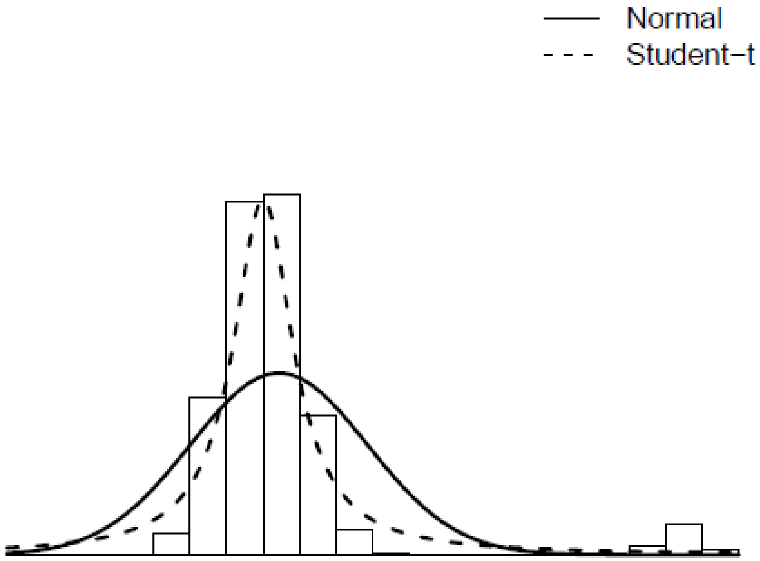
- Fitting a PDF to different histograms



PDF Fitting

- Fitting different PDFs to a histogram

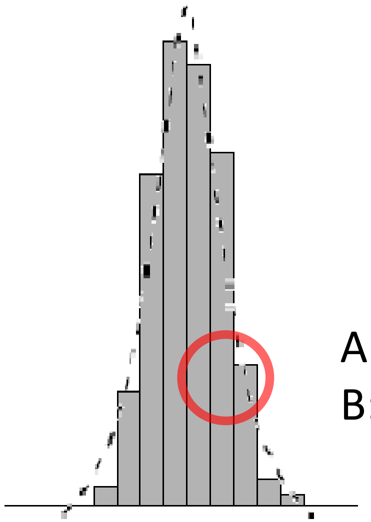




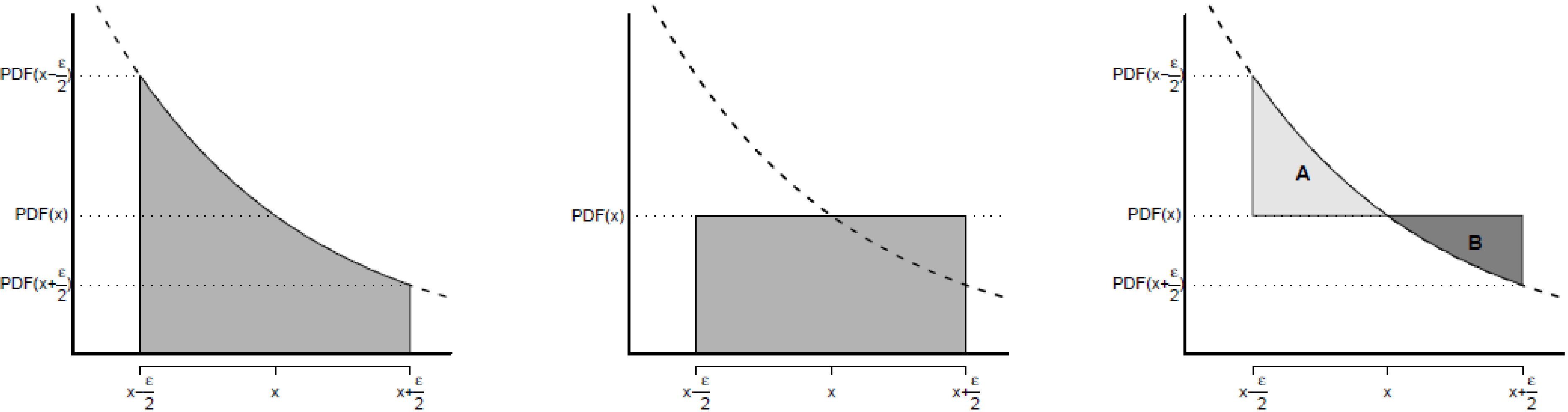
the same dataset

PDF Fitting

- Interval error
  - Errors produced by the interval size
  - There is no hard and fast rule for deciding on interval size
  - By case



A: + error
B: - error



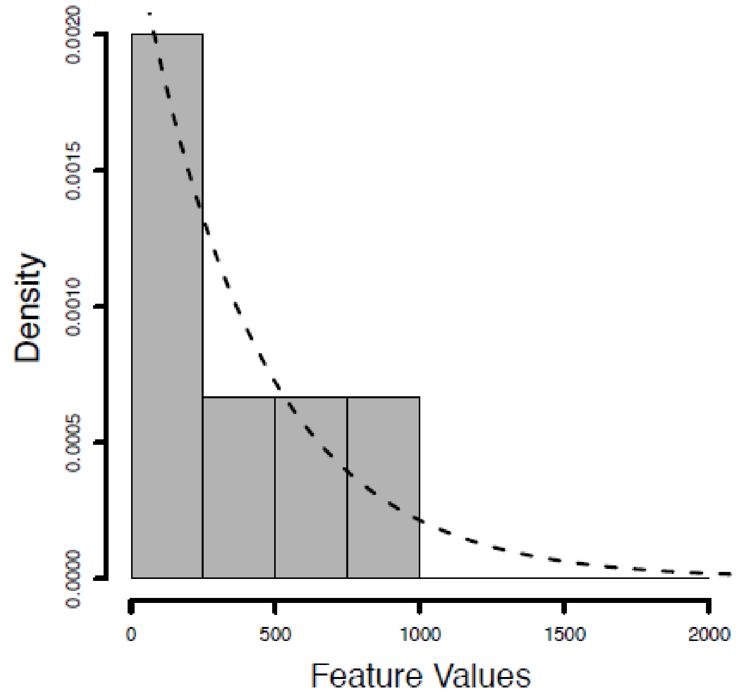
PDF & Naive Bayes' Classifier

- An example of loan application fraud detection with **account balance (*AB*)**

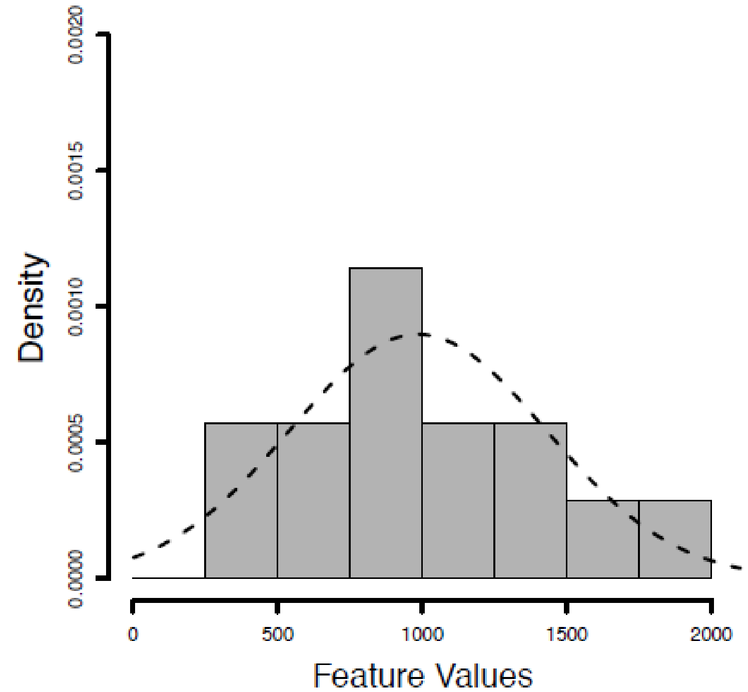
ID	Credit History	Guarantor/ CoApplicant	Accommodation	Account Balance	Fraud
1	current	none	own	56.75	true
2	current	none	own	1,800.11	false
3	current	none	own	1,341.03	false
4	paid	guarantor	rent	749.50	true
5	arrears	none	own	1,150.00	false
6	arrears	none	own	928.30	true
7	current	none	own	250.90	false
8	arrears	none	own	806.15	false
9	current	none	rent	1,209.02	false
10	none	none	own	405.72	true
11	current	coapplicant	own	550.00	false
12	current	none	free	223.89	true
13	current	none	rent	103.23	true
14	paid	none	own	758.22	false
15	arrears	none	own	430.79	false
16	current	none	own	675.11	false
17	arrears	coapplicant	rent	1,657.20	false
18	arrears	none	free	1,405.18	false
19	arrears	none	own	760.51	false
20	current	none	own	985.41	false

PDF & Naive Bayes' Classifier

- Binning for continuous data ➔Histogram
- Choose a PDF to fit each histogram



$P(AB = x|fr)$



$P\left(AB = x\middle|\overline{fr}\right)$

Bin size: 250

PDF & Naive Bayes' Classifier

- A simple method to fit the exponential distribution
  - Compute the sample mean, μ, of the ACCOUNT BALANCE where FRAUDULENT = 'True'
  - Let $\lambda = \frac{1}{\mu}$
  - Then,

$$E(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}$$

PDF & Naive Bayes' Classifier

- A simple method to fit the normal distribution
  - Compute the sample mean, μ, and standard deviation, σ, of the ACCOUNT BALANCE where FRAUDULENT = 'False'
  - Then,

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

PDF & Naive Bayes' Classifier

- To implement a probability-based learning model, you have to do that
  - applying the Laplace smoothing for each categorical feature, and
  - fitting a PDF for each continuous feature

PDF & Naive Bayes' Classifier

- For example, how about that *FRAUDULENT (FR) = ?* if
  - *CREDIT HISTORY (CH) = paid*
  - *GUARANTOR/COAPPLICANT (GC) = guarantor*
  - *ACCOMODATION (ACC) = free*
  - *ACCOUNT BALANCE* (*AB*) = 759.07

$$
\begin{array}{rcl@{\qquad}rcl}
P(fr) & = & 0.3 & P(\neg fr) & = & 0.7 \\
P(CH = paid \mid fr) & = & 0.2222 & P(CH = paid \mid \neg fr) & = & 0.2692 \\
P(GC = guarantor \mid fr) & = & 0.2667 & P(GC = guarantor \mid \neg fr) & = & 0.1304 \\
P(ACC = free \mid fr) & = & 0.2 & P(ACC = free \mid \neg fr) & = & 0.1739 \\
P(AB = 759.07 \mid fr) & & & P(AB = 759.07 \mid \neg fr) & & \\
\approx E\left(\begin{array}{c} 759.07, \\ \lambda = 0.0024 \end{array}\right) & = & 0.00039 & \approx N\left(\begin{array}{c} 759.07, \\ \mu = 984.26, \\ \sigma = 460.94 \end{array}\right) & = & 0.00077
\end{array}
$$

$$\left(\prod_{k=1}^{m} P(\mathbf{q}[k] \mid fr)\right) \times P(fr) = 0.0000014$$

$$\left(\prod_{k=1}^{m} P(\mathbf{q}[k] \mid \neg fr)\right) \times P(\neg fr) = 0.0000033$$

Binning & Naive Bayes' Classifier

- The loan application fraud detection with a second continuous descriptive feature added: LOAN AMOUNT (LA)

ID	Credit History	Guarantor/ CoApplicant	Accommodation	Account Balance	Loan Amount	Fraud
1	current	none	own	56.75	900	true
2	current	none	own	1 800.11	150 000	false
3	current	none	own	1 341.03	48 000	false
4	paid	guarantor	rent	749.50	10 000	true
5	arrears	none	own	1 150.00	32 000	false
6	arrears	none	own	928.30	250 000	true
7	current	none	own	250.90	25 000	false
8	arrears	none	own	806.15	18 500	false
9	current	none	rent	1 209.02	20 000	false
10	none	none	own	405.72	9 500	true
11	current	coapplicant	own	550.00	16 750	false
12	current	none	free	223.89	9 850	true
13	current	none	rent	103.23	95 500	true
14	paid	none	own	758.22	65 000	false
15	arrears	none	own	430.79	500	false
16	current	none	own	675.11	16 000	false
17	arrears	coapplicant	rent	1 657.20	15 450	false
18	arrears	none	free	1 405.18	50 000	false
19	arrears	none	own	760.51	500	false
20	current	none	own	985.41	35 000	false

Binning & Naive Bayes' Classifier

- Bin size

Bin Thresholds		
	Bin1	$\leq 9,925$
$9,925 <$	Bin2	$\leq 19,250$
$19,225 <$	Bin3	$\leq 49,000$
$49,000 <$	Bin4	

ID	Loan Amount	Binned Loan Amount	Fraud
15	500	bin1	false
19	500	bin1	false
1	900	bin1	true
10	9,500	bin1	true
12	9,850	bin1	true
4	10,000	bin2	true
17	15,450	bin2	false
16	16,000	bin2	false
11	16,750	bin2	false
8	18,500	bin2	false

ID	Loan Amount	Binned Loan Amount	Fraud
9	20,000	bin3	false
7	25,000	bin3	false
5	32,000	bin3	false
20	35,000	bin3	false
3	48,000	bin3	false
18	50,000	bin4	false
14	65,000	bin4	false
13	95,500	bin4	true
2	150,000	bin4	false
6	250,000	bin4	true

Binning & Naive Bayes' Classifier

- *FRAUDULENT (FR) = ?* if
    - *CREDIT HISTORY (CH) = paid*
    - *GUARANTOR/COAPPLICANT (GC) = guarantor*
    - *ACCOMODATION (ACC) = free*
    - *ACCOUNT BALANCE (AB)* = 759.07
    - LOAN AMOUNT(LA) = 8000

$$
\begin{array}{rclrcl}
P(fr) & = & 0.3 & P(\neg fr) & = & 0.7 \\
P(CH = paid \mid fr) & = & 0.2222 & P(CH = paid \mid \neg fr) & = & 0.2692 \\
P(GC = guarantor \mid fr) & = & 0.2667 & P(GC = guarantor \mid \neg fr) & = & 0.1304 \\
P(ACC = free \mid fr) & = & 0.2 & P(ACC = free \mid \neg fr) & = & 0.1739 \\
P(AB = 759.07 \mid fr) & & & P(AB = 759.07 \mid \neg fr) & & \\
\approx E\left(\begin{array}{c} 759.07, \\ \lambda = 0.0024 \end{array}\right) & = & 0.00039 & \approx N\left(\begin{array}{c} 759.07, \\ \mu = 984.26, \\ \sigma = 460.94 \end{array}\right) & = & 0.00077 \\
P(BLA = bin1 \mid fr) & = & 0.3333 & P(BLA = bin1 \mid \neg fr) & = & 0.1923
\end{array}
$$

$$\left(\prod_{k=1}^{m} P(\mathbf{q}[k] \mid fr)\right) \times P(fr) = 0.000000462$$

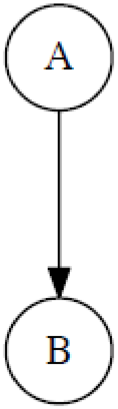
$$\left(\prod_{k=1}^{n} P(\mathbf{q}[k] \mid \neg fr)\right) \times P(\neg fr) = 0.000000633$$

Bayesian Networks

- A graph-based representation to encode the structural relationships
- It use a directed acyclic graph that is composed of thee basic elements:
  - Nodes
  - Edges
  - Conditional probability tables

P(A=T)	P(A=F)
0.4	0.6

A	P(B=T|A)	P(B=F|A)
T	0.3	0.7
F	0.4	0.6



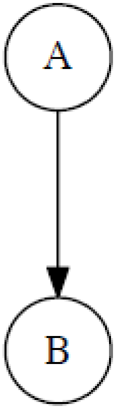
Bayesian Networks

- Recall the chain rule, the joint probability can be computed as follows

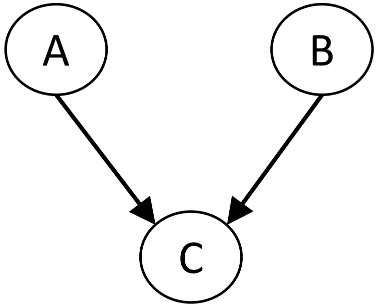
$$P(X_1, X_2, \dots, X_m) = P(X_1) \prod_{i=2}^{m} P(X_i | X_{i-1}, \dots, X_2, X_1)$$

- In a Bayesian network

$$P(X_1, X_2, \dots, X_m) = \prod_{i=1}^{m} P(X_i | Parents(X_i))$$

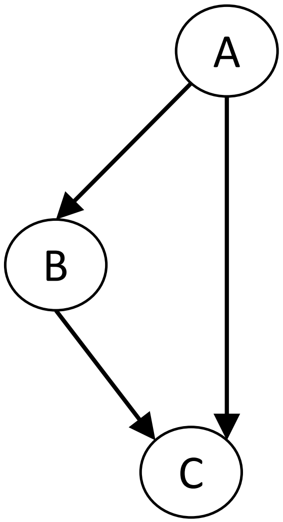


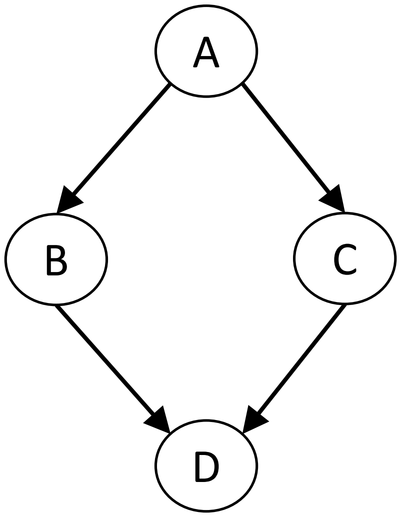
$$P(A, B) = P(A)P(B \mid A)$$



$$P(A, B, C) = P(A)P(B)P(C \mid A, B)$$

Bayesian Networks





$P(A, B) = P(A)P(B|A)$

$P(A, B, C) = P(A)P(B|A)P(C \mid A, B)$

$P(A, B, \mathrm{C}) = P(A)P(B|A)P(C|A)$
$P(A, B, \mathrm{C}, D) = P(A)P(B|A)P(C|A)P(D|B, \mathrm{C})$

Bayesian Networks

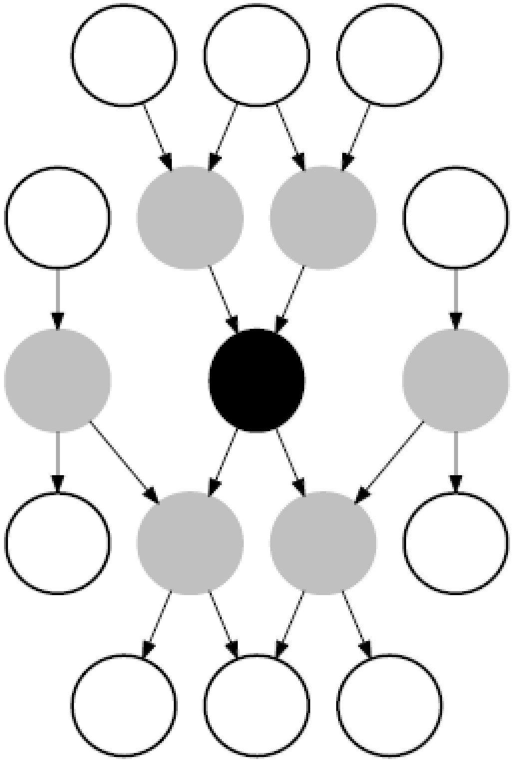
- Constructing a Bayesian network for *m* **ordered** variables, $\{X_1, X_2, \ldots, X_m\}$
  - For i = 1 to n
    - add X_i to the network
    - select parent from $\{X_1, X_2, \ldots, X_{i-1}\}$, the selected parent must guarantees

$$P(X_1, X_2, \ldots, X_m) = P(X_1) \prod_{i=2}^{m} P(X_i | X_{i-1}, \ldots, X_2, X_1)$$

$$= \prod_{i=1}^{m} P(X_i | Parents(X_i))$$

Bayesian Networks

- Markov blanket
  - **The Markov blanket of a node** is the set of nodes consisting of <u>its parents</u>, <u>its children</u>, and <u>any other parents of its children</u>.

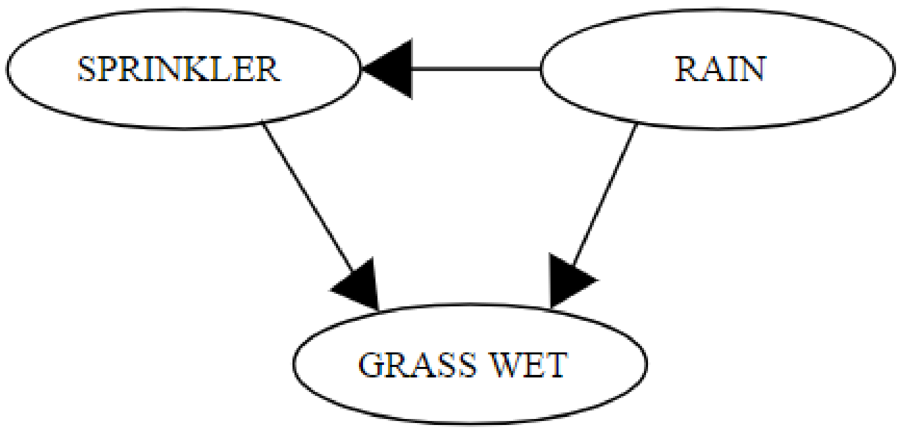


The black node is **conditionally independence** of the white nodes

Bayesian Networks

- Example
  - There are two events which could cause **grass to be wet (G)**: either the **sprinkler (S)** is on or it's **raining (R)**.
  - Suppose that the rain has a direct effect on the use of the sprinkler

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Bayesian Networks

- What is the probability that it is raining, given the grass is wet?
  - $P(R = true \mid G = true) = P(r|g) =?$

$$P(r \mid g) = \frac{P(g,r)}{P(g)}$$

$$= \frac{P(g,s,r) + P(g,\bar{s},r)}{P(g,\bar{s},\bar{r}) + P(g,\bar{s},r) + P(g,s,\bar{r}) + P(g,s,r)}$$

$$P(G,S,R) =?$$

Bayesian Networks

- The joint probability in Bayesian network:

$$P(G,S,R) = P(R)\,P(S|R)P(G|S,R)$$

- Check the Bayesian network, we have:
  - $P(r) = 0.2$
  - $P(s|r) = 0.01$
  - $P(g|s,r) = 0.99$
- Then,

$$P(g,s,r) = 0.2 \times 0.01 \times 0.99 = 0.00198$$

- Other joint probabilities
  - $P(g,\bar{s},\bar{r}) = P(\bar{r})P(\bar{s},|\bar{r})P(g|\bar{s},\bar{r}) = 0.8 \times 0.6 \times 0.0 = 0.0$
  - $P(g,\bar{s},r) = P(r)P(\bar{s},|r)P(g|\bar{s},r) = 0.2 \times 0.99 \times 0.8 = 0.1584$
  - $P(g,s,\bar{r}) = P(\bar{r})P(s,|\bar{r})P(g|s,\bar{r}) = 0.8 \times 0.4 \times 0.9 = 0.288$

Bayesian Networks

- Therefore,

$$P(r \mid g) = \frac{0.00198 + 0.1584}{0.00198 + 0.288 + 0.1584 + 0.0} = 0.3577$$

Bayesian Networks

- Example
  - What is the probability of burglary if John and Mary call to report the alarm

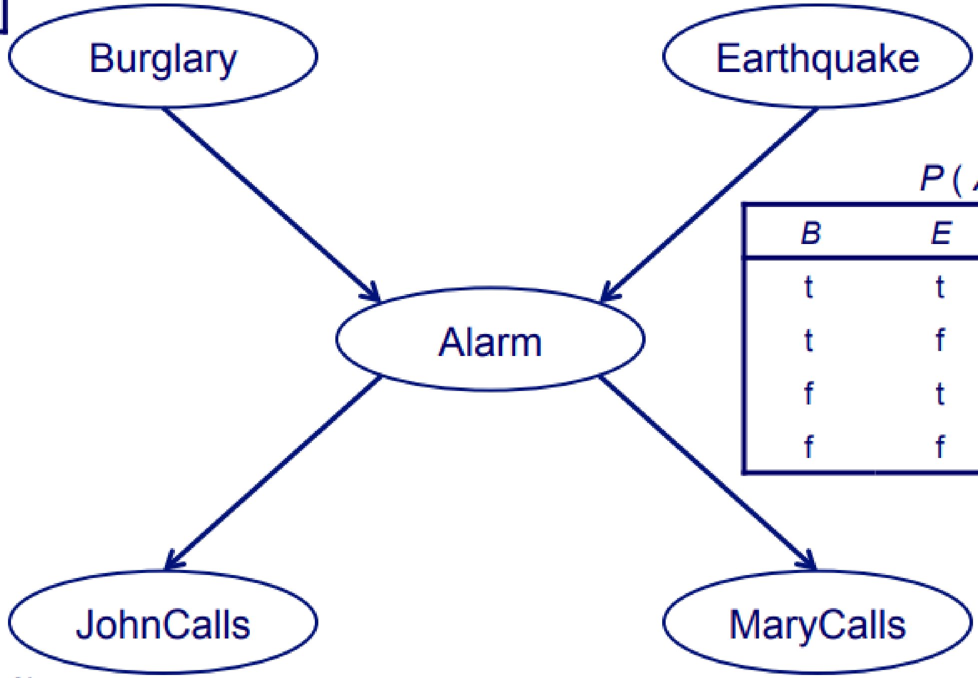
$$P(b \mid j, m) = ?$$

P (B)

t	f
0.001	0.999

P (E)

t	f
0.001	0.999



P (A | B, E)

B	E	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (J | A)

A	t	f
t	0.9	0.1
f	0.05	0.95

P (M | A)

A	t	f
t	0.7	0.3
f	0.01	0.99

Bayesian Networks

- Because

$$P(b \mid j,m) = \frac{P(b,j,m)}{P(j,m)} = \frac{P(b,j,m)}{P(b,j,m) + P(\bar{b},j,m)}$$

- And

$$P(b,e,a,j,m) = p(b)p(e)p(a|b,e)p(j|a)p(m|a)$$

- Therefore,

$$P(b,e,j,m) = \sum_{A\in\{a,\bar{a}\}} P(b,e,\mathrm{A},j,m) = \sum_{A\in\{a,\bar{a}\}} p(b)p(e)p(A|b,e)p(j|A)p(m|A)$$

$$P(b,j,m) = \sum_{E\in\{e,\bar{e}\}} P(b,E,j,m) = \sum_{E\in\{e,\bar{e}\}} \sum_{A\in\{a,\bar{a}\}} p(b)p(E)p(A|b,E)p(j|A)p(m|A)$$

$$P\left(\bar{b},j,m\right) = \sum_{E\in\{e,\bar{e}\}} P(b,E,j,m) = \sum_{E\in\{e,\bar{e}\}} \sum_{A\in\{a,\bar{a}\}} p(b)p(E)p(A|b,E)p(j|A)p(m|A)$$

Bayesian Networks

- Learning model using Bayesian network
- Given a query **q** with *m* features
  - $\mathbf{q} = \{X_1, X_2, \dots, X_m\}$
- And there are *n* target levels
  - $\mathbf{T} = \{Y_1, Y_2, \dots, Y_n\}$
- Then,
  - $M(\mathbf{q}) = \underset{Y \in \mathbf{T}}{\operatorname{argmax}} P(Y \mid X_1, X_2, \dots, X_m)$
- Example:
  - $M(\mathbf{q}) = \underset{B \in \{b, \bar{b}\}}{\operatorname{argmax}} P(B \mid j, m)$

Markov Chain

- The sequence of random variables such a process moves through.
- The next state of the process only depends on the previous state and not the sequence of states.
- Andrey Markov
  - 1856 - 1922
  - Russian mathematician

Markov Chain

- Discrete-time Markov chain
  - a sequence of random variables $X_1, X_2, \dots, X_n$ with the Markov property

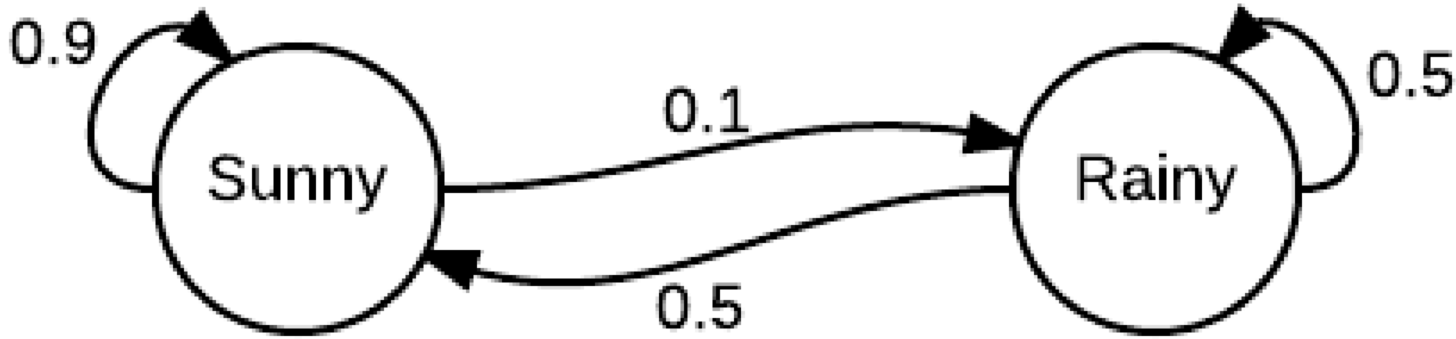
$$P(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_1 = x_1)$$

$$= P(X_n = x_n | X_{n-1} = x_{n-1})$$

if $P(X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_1 = x_1) > 0$

Markov Chain

- A simple example: The probabilities of weather conditions
  - A sunny day is 90% likely to be followed by another sunny day.
  - A rainy day is 50% likely to be followed by another rainy day.



Trasition matrix: $P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$

Initial state: $\mathbf{x}^0 = \begin{bmatrix} p^0_{sunny} & p^0_{rainy} \end{bmatrix}$

$$\mathbf{x}^i = \mathbf{x}^{i-1}P = \mathbf{x}^0 p^i$$

Markov Chain

- A simple example: The probabilities of weather conditions

Given $\mathbf{x}^0 = [1.0 \quad 0.0]$

The 1st day:

$$\mathbf{x}^1 = \mathbf{x}^0 P = [0.9 \quad 0.1]$$

The 2nd day:

$$\mathbf{x}^2 = \mathbf{x}^1 P = [0.86 \quad 0.14]$$

How about

$$\mathbf{q} = \lim_{i \to \infty} \mathbf{x}^i$$

Markov Chain

- A state t has **period** k if any return to state t must occur in multiples of k time steps.
- If $k = 1$, then the state is said to be **aperiodic**.
- A Markov chain is **irreducible** if its state space is a single communicating class; in other words,
  - if it is possible to get to any state from any state
  - all the states communicate with each other
  - all states are aperiodic.
- If the Markov chain is irreducible and aperiodic, then there is a unique stationary distribution **q**.

$$\mathbf{q} = \lim_{i \to \infty} \mathbf{x}^i$$
$$\mathbf{q}P = \mathbf{q}$$
$$\mathbf{q}(P - I) = 0$$

Markov Chain

- A simple example: The probabilities of weather conditions

$$P - I = \begin{bmatrix} -0.1 & 0.1 \\ 0.5 & -0.5 \end{bmatrix}$$

$$\mathbf{q}(P - I) = 0$$

$-0.1q_1 + 0.5q_2 = 0$
and $q_1 + q_2 = 1.0$ (sum of probabilities)

➔$\mathbf{q} = [0.833 \quad 0.166667]$

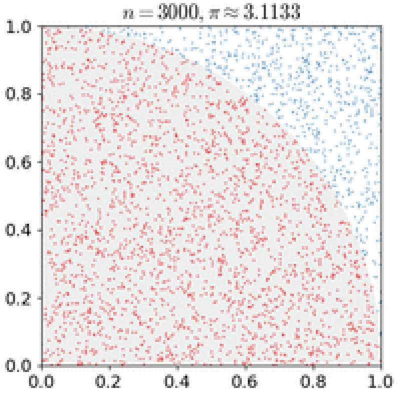
- In conclusion, in the long term, about 83.3% of days are sunny.

Markov Chain

- A big question of applying Markov chain to machine learning
  - The number of data instances in our training is pretty large.
  - The number of features is also large.
  - There are many choices to create a Markov chain for our training data.
  - How to create the best Markov chain for our training data?

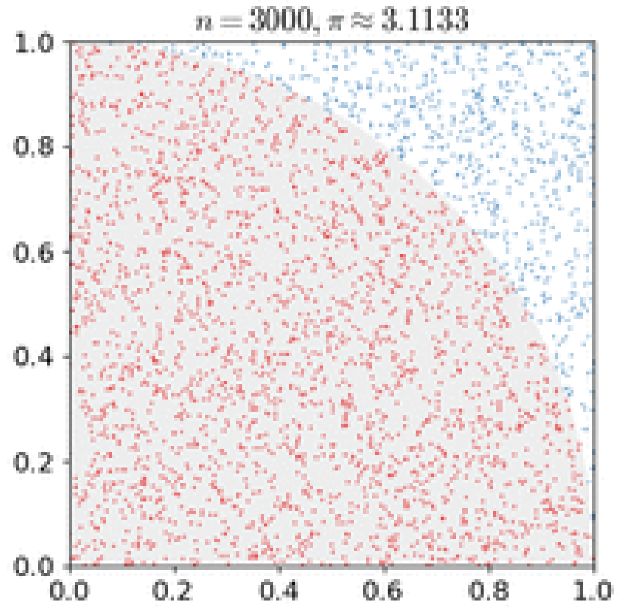
Monte Carlo Method

- Repeatedly, evenly, and randomly sample from a domain to obtain numerical results.
  1. Define a domain of possible inputs
  2. Generate inputs randomly from a probability distribution over the domain
  3. Perform a deterministic computation on the inputs
  4. Aggregate the results



Monte Carlo Method

- Example: π = ?
  1. Draw a square, then inscribe a circle within it
  2. Uniformly scatter objects of uniform size over the square
  3. Count the number of objects inside the circle and the total number of objects
  4. The ratio of the inside-count and the total-sample-count is an estimate of the ratio of the two areas, which is π /4. Multiply the result by 4 to estimate π



Markov Chain Monte Carlo, MCMC

- **MCMC** methods are **a class of algorithms** for sampling from a probability distribution based on **constructing a Markov chain** that has the desired distribution as its equilibrium distribution.
- Finding a good state transition
- Two MCMC methods are commonly used in machine learning
  - **Metropolis-Hastings method**
  - **Gibbs sampling**

Markov Chain Monte Carlo, MCMC

- **Metropolis-Hastings method**
  1. Initialize x_0
  2. For $i = 0$ to $n - 1$
  3. Randomly generate a candidate state $x' \sim q(x'|x_i)$
  4. Generate a uniform random number $u \sim U[0,1]$
  5. If $u < A(x_i, x') = \min(1, \frac{p(x')\cancel{q(x_i|x')}}{p(x_i)\cancel{q(x'|x_i)}})$
  6. $x_{i+1} = x'$
  7. else
  8. $x_{i+1} = x_i$

- where q is called **proposal density,** which is an arbitrary probability density
  - q must satisfy $q(\boldsymbol{x}|\boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x})$
  - Gaussian distribution is commonly used be q

Markov Chain Monte Carlo, MCMC

- **Gibbs sampling**

1. Initialize $\mathbf{x}^0 = [x_1^0, x_2^0, \dots, x_m^0]$
2. For $i = 0$ to $n - 1$
3. Crate the next sample $\mathbf{x}^{i+1} = [x_1^{i+1}, x_2^{i+1}, \dots, x_m^{i+1}]$
4. sample $x_1^{i+1} \sim p\left(x_1 | x_2^i, x_3^i, \dots x_m^i\right)$
5. sample $x_2^{i+1} \sim p\left(x_2 | x_1^{i+1}, x_3^i, \dots x_m^i\right)$

   …
6. sample $x_j^{i+1} \sim p\left(x_j | x_1^{i+1}, \dots, x_{j-1}^{i+1}, x_{j+1}^i, \dots x_m^i\right)$

   …
7. sample $x_m^{i+1} \sim p\left(x_m | x_1^{i+1}, x_2^{i+1}, \dots, x_{m-1}^{i+1}\right)$
8. Repeat step 2 - 7 *k* times ($\mathbf{x}^0 = \mathbf{x}^n$).

Markov Chain Monte Carlo, MCMC

- An example of Gibbs sampling
  - Darren Wilkinson, "MCMC programming in R, Python, Java, and C", 2016
- Two variables: x and y
  - $p(x|y) = \text{Gamma PDF with } k = 3, \theta = y^2 + 4$

$$G(x, k, \theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}$$

  - $p(y|x) = \text{Gaussian PDF with } \mu = \frac{1}{x+1}, \sigma = \frac{1}{\sqrt{2(x+1)}}$

$$N(y, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

1. $\mathrm{x} = 0, \mathrm{y} = 0$
2. $\text{for i} = 1 \text{ to N}$
3. $\quad\text{for j} = 1 \text{ to M}$
4. $\qquad \mathrm{x} \sim G(x, k, \theta)$
5. $\qquad \mathrm{y} \sim N(y, \mu, \sigma)$
6. $\quad \text{Output[i]} = (\mathrm{x}, \mathrm{y})$

Markov Chain Monte Carlo, MCMC

- Advanced reading
  - C. Andrieu, et al. "**An Introduction to MCMC for Machine Learning**," *Kluwer Academic*, 2003
  - Paolo, et al. "**Bayesian Function Learning Using MCMC Methods**," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, Dec. 1998.