Pattern Classification (IV)

杜俊

jundu@ustc.edu.cn

Outline

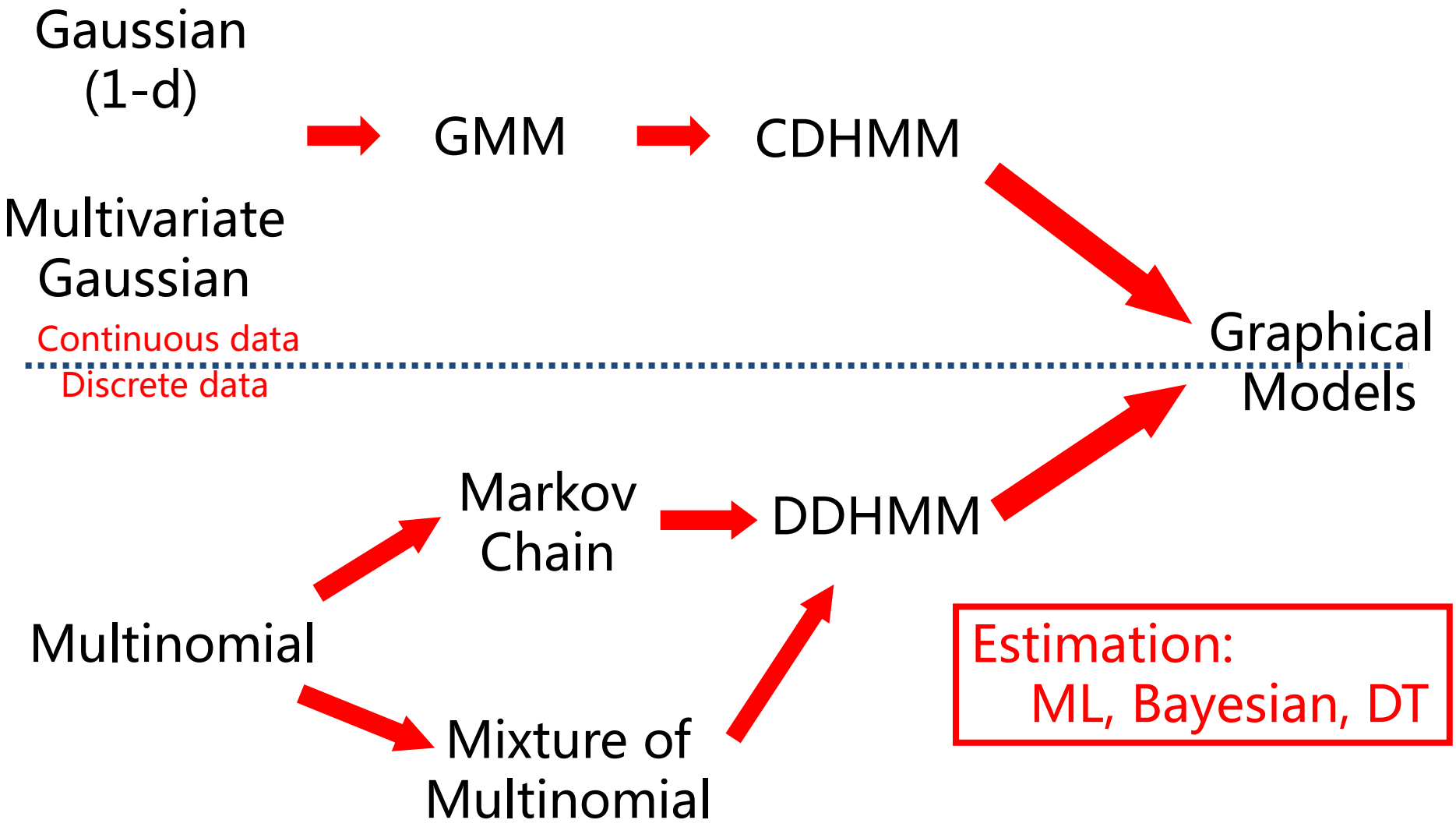
- Bayesian Decision Theory
  - How to make the optimal decision?
  - Maximum *a posterior* (MAP) decision rule
- Generative Models
  - Joint distribution of observation and label sequences
  - Model estimation: MLE, Bayesian learning, discriminative training
- Discriminative Models
  - Model the posterior probability directly (discriminant function)
  - Logistic regression, support vector machine, neural network

Plug-in MAP Decision Rule

$$\mathrm{C}_P = \arg\max_{\mathrm{C}_i} p(\mathrm{C}_i \mid X) = \arg\max_{\mathrm{C}_i} P(\mathrm{C}_i) \cdot p(X \mid \mathrm{C}_i)$$

$$\approx \arg\max_{\mathrm{C}_i} \overline{P}_{\Gamma_i}(\mathrm{C}_i) \cdot \overline{p}_{\Lambda_i}(X \mid \mathrm{C}_i)$$

Statistical Models: Roadmap



Model Parameter Estimation (I)

- Maximum Likelihood (ML) Estimation:
  - Objective function: likelihood function of all observed data
  - ML method: most popular model estimation; simplest
  - EM (Expected-Maximization) algorithm
  - Examples:
    - Univariate Gaussian distribution
    - Multivariate Gaussian distribution
    - Multinomial distribution
    - Gaussian mixture model (GMM)
    - Markov chain model: n-gram for language modeling
    - Hidden Markov model (HMM)
- Bayesian Model Estimation
  - The MAP (maximum *a posteriori*) estimation (point estimation)
  - General Bayesian theory for parameter estimation
  - Recursive Bayes learning (Sequential Bayesian learning)

Model Parameter Estimation (II)

- Discriminative Training:
  - Maximum mutual information (MMI) estimation
    - The model is viewed as a noisy data generation channel
      Class id C_i ➔ observation feature X
    - Maximize mutual information between C_i and X
  - Minimum classification error (MCE)
    - Minimize empirical classification error
      - error rate in training data set

- Minimum Discrimination Information (MDI):
  - A PDF $f(X|\Lambda)$ defined by model with unknown parameters
  - A sample distribution p(X) derived directly from data
  - Determine Λ to minimize KL-divergence between $f(X|\Lambda)$ and p(X)

Maximum Likelihood Estimation (I)

- After data modeling, we know the model form of $p(X \mid C_i)$.
- For each class C_i, $p(X \mid C_i, \theta_i)$ with unknown parameters θ_i.
- In pattern classification problem, we usually collect a sample set for each class, we have N data sets, $D_1, D_2, \ldots, D_N$.
- The parameter estimation problem: to use the information provided by the training samples $D_1, D_2, \ldots, D_N$, to obtain good estimates for the unknown parameter vectors, $\theta_1, \theta_2, \ldots, \theta_N$.

Maximum Likelihood Estimation (II)

- The Maximum Likelihood (ML) principle: we view the parameters as fixed values but unknown. The best estimate is defined to maximize the probability of observing the samples actually observed.
  - Best interpret the data
  - Fit the data best.

- The likelihood function
  - $p(X \mid \theta)$ → data distribution PDF of different X if θ is given
  - $p(X \mid \theta)$ → likelihood function of θ if data X is given

Maximum Likelihood Estimation (III)

- Problem: use information $D_1, D_2, \ldots, D_N$ to estimate $\theta_1, \theta_2, \ldots, \theta_N$.

- Assumption I: samples in D_i give no information about θ_j if $i!=j$. Thus we estimate parameters for each class separately and estimate each θ_i solely based on D_i.
  - the joint estimation becomes: use a set D of training samples drawn independently from the probability density $p(X \mid \theta)$ to estimate the unknown parameter vector θ.

- Assumption II: all samples in each set D_i are i.i.d. (independent and identically distributed), i.e., the samples are drawn independently according to the same probability law $p(X \mid \theta_i)$.

Maximum Likelihood Estimation (IV)

- Assume D contains n samples, X_1, X_2, …, X_n, since the samples were drawn independently from p(X | θ), thus the probability of observing D is

$$p(D \mid \theta) = \prod_{k=1}^{n} p(X_k \mid \theta)$$

- If viewed as a function of θ, p(D|θ) is called the likelihood function of θ with respect to the sample set D.
- The maximum-likelihood estimate of θ is the value θ_{ML} that maximizes p(D|θ).

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \; p(D \mid \theta) = \arg\max_{\theta} \prod_{k=1}^{n} p(X_k \mid \theta)$$

- Intuitively, θ_{ML} corresponds to the value of θ which in some senses best agrees with or supports the actually observed training samples.

Maximum Likelihood Estimation (V)

- In many cases, it is more convenient to work with the logarithm of the likelihood rather than the likelihood itself.
- Denote the log-likelihood function l(θ)= ln p(D|θ), we have

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} l(\theta) = \arg\max_{\theta} \sum_{k=1}^{n} \ln p(X_k \mid \theta)$$

- How to do maximization in ML estimation:
  - For simple models: differential calculus
    - Single univariate/multivariate Gaussian model
  - Model parameters with constraints: Lagrange optimization
    - Multinomial/ Markov chain model
  - Complex models: Expectation-Maximization (EM) method
    - GMM/HMM

Maximization: Differential Calculus

- The log-likelihood function:

$$l(\theta) \equiv \ln p(D \mid \theta) = \sum_{k=1}^{n} \ln p(X_k \mid \theta)$$

- Assume θ is a p-component vector θ=(θ1, θ2,..., θp), and let ∇_θ be the gradient operator as:

$$\nabla_\theta = \begin{bmatrix} \partial / \partial \theta_1 \\ \vdots \\ \partial / \partial \theta_p \end{bmatrix}$$

- Maximization is done by equating to zero:

$$\nabla_\theta l(\theta) = \sum_{k=1}^{n} \nabla_\theta \ln p(X_k \mid \theta) \qquad \nabla_\theta l(\theta) = 0$$

Example: Univariate Gaussian (I)

- Training data D={x_1, x_2, … , x_n}
- Model the data by using a univariate Gaussian distribution:

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

- Assume we know the variance, we only need to estimate the unknown mean from the data by using ML estimation.
- The log-likelihood function:

$$l(\mu) = \ln p(D \mid \mu) = \ln \prod_{k=1}^{n} p(x_k \mid \mu)$$

$$= \sum_{k=1}^{n} \ln p(x_k \mid \mu) = \sum_{k=1}^{n} [-\frac{\ln(2\pi\sigma^2)}{2} - \frac{(x_k - \mu)^2}{2\sigma^2}]$$

Example: Univariate Gaussian (II)

- Maximization:

$$\frac{\mathrm{d}}{\mathrm{d}\mu} l(\mu) = 0$$

$$\Rightarrow \sum_{k=1}^{n} (x_k - \mu) = 0$$

$$\Rightarrow \mu_{\mathrm{ML}} = \frac{\sum_{k=1}^{n} x_k}{n}$$

- ML estimate of the unknown Gaussian mean is the sample mean.

Example: Multivariate Gaussian (I)

- Training data D={X_1, X_2, …, X_n} (a set of vectors)
- Model the data with a multivariate Gaussian distribution

$$p(X \mid \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \mid \Sigma \mid^{1/2}} \exp\left[-\frac{(X-\mu)^{\mathrm{T}} \Sigma^{-1} (X-\mu)}{2} \right]$$

- Assume both mean vector and variance matrix are unknown.
- The log-likelihood function:

$$l(\mu, \Sigma) = \ln p(D \mid \mu, \Sigma) = \sum_{k=1}^{n} \ln p(X_k \mid \mu, \Sigma)$$

$$= C - \frac{n}{2} \ln \mid \Sigma \mid - \frac{1}{2} \cdot \sum_{k=1}^{n} (X_k - \mu)^{\mathrm{T}} \Sigma^{-1} (X_k - \mu)$$

Example: Multivariate Gaussian (II)

- Maximization:

$$\frac{\partial\, l(\mu,\Sigma)}{\partial \mu}=0 \;\Rightarrow\; \sum_{k=1}^{n}\Sigma^{-1}(X_k-\mu)=0$$

$$\Rightarrow \quad \mu_{\mathrm{ML}}=\frac{1}{n}\sum_{k=1}^{n}X_k$$

$$\frac{\partial\, l(\mu,\Sigma)}{\partial \Sigma}=0$$

$$\Rightarrow -\frac{n}{2}(\Sigma^{-1})^{\mathrm{T}}+\frac{1}{2}\sum_{k=1}^{n}(\Sigma^{-1})^{\mathrm{T}}(X_k-\mu)(X_k-\mu)^{\mathrm{T}}(\Sigma^{-1})^{\mathrm{T}}=0$$

$$\Rightarrow \quad \frac{n}{2}\Sigma=\frac{1}{2}\sum_{k=1}^{n}(X_k-\mu)(X_k-\mu)^{\mathrm{T}}$$

$$\Rightarrow \quad \Sigma_{\mathrm{ML}}=\frac{1}{n}\sum_{k=1}^{n}(X_k-\mu)(X_k-\mu)^{\mathrm{T}}$$

Pattern Classification via Gaussian models

- Given N classes {C_1, C_2, …, C_N}, for each class we collect a set of training samples, D_i = {X_{i1}, X_{i2}, …, X_{iT}}, for class C_i.
- For each sample, we observe its feature vector X with its true class id
- If the feature vector is continuous and uni-modal, we may want to model each class by a multivariate Gaussian distribution, $N(\mu,\Sigma)$.
- Thus we have N different multivariate distributions, $N(\mu_i,\Sigma_i)$
- The model forms are known but parameters μ_i and Σ_i are unknown.
- Use training data to estimate the parameters based on ML criterion.
- When observing an unknown pattern Y, classify with the estimated models based on the plug-in Bayes decision rule:

$$C_Y = \arg\max_{C_i} P(C_i)\cdot p(Y \mid C_i) = \arg\max_{C_i} N(Y \mid \mu_i^{ML}, \Sigma_i^{ML})$$

Example: Multinomial Distribution (I)

- A DNA sequence consists of a sequence of 4 different types of nucleotides (G, A, T, C). For example,

  X=GAATTCTTCAAAGAGTTCCAGATATCCACAGGCAGATTCTACAAAAGAAGTGTTTCAATACTGCTCTATAAAAGATGTATTCCACTCAGTTACT
  TTCATGCACACATCTCAATGAAGTTCCTGAGAAAGCTTCTGTCTAGTTTTTATGTGAAAATATTTCCTTTTCCATCATGGGCCTCAAAGCGCTCAAA
  ATGAACCCTTGCAGATACTAGAGAAAGACTGTTTCAAAACTGCTCTATCCAAAGAACGGTTCCACTCTGTGAGGTGAATGCACACATCACAAAGC
  AGTTTCTGAGAACGCTTCTGTCTAGTTTGTAGGTGAAGATATTTCCTTTTCCTTCATAGGCCTCTAATCGCTCCAAATATCCACAAGCAGATTCTTC
  AAAATGTGTGTTTCAACACTGCTCTATCAAAAGAAAGGTTCAAGTCTGTGAGTTGAATGCACACATCACAAAGCAGTTTCTGAGAATGCCTCTGT
  CTAGTTTGTATGTGAAGATATTTCTTTTTCCGTCTTATGCCTCAAATCGCTCCAAATATCCACTTGCAGATACTTCAAAA

- If assume all nucleotides in a DNA sequence are independent, we can use multinomial distribution to model a DNA sequence
- Use p_1 to denote probability to observe G in any one location, p_2 for A, p_3 for T, p_4 for C, then $p_1+p_2+p_3+p_4=1$
- Given a DNA sequence X, the probability to observe X is

$$\Pr(X) = C \cdot \prod_{i=1}^{4} p_i^{N_i}$$

- Where N_1 is frequency of G appearing in X, N_2 frequency of A, N_3 frequency of T, N_4 frequency of C.

Example: Multinomial Distribution (II)

- Problem: estimate p_1, p_2, p_3, p_4 from a training sequence X based on the maximum likelihood criterion.
- The log-likelihood function:

$$l(p_1, p_2, p_3, p_4) = \sum_{i=1}^{4} N_i \cdot \ln p_i$$

- Maximization l(.) subject to the constraint

$$\sum_{i=1}^{4} p_i = 1$$

- Use Lagrange optimization:

$$L(p_1, p_2, p_3, p_4, \lambda) = \sum_{i=1}^{4} N_i \cdot \ln p_i - \lambda(\sum_{i=1}^{4} p_i - 1)$$

$$\frac{\partial}{\partial p_i} L(p_1, p_2, p_3, p_4, \lambda) = 0 \quad \Rightarrow \quad N_i / p_i - \lambda = 0$$

Example: Multinomial Distribution (III)

- Finally, we get the ML estimation for the multinomial distribution as:

$$p_i = \frac{N_i}{\sum_{i=1}^{4} N_i} \qquad (i = 1,2,3,4)$$

- We only need count the occurrence frequency of each nucleotides.

- Similar derivation also holds for Markov chain model.
  - An important application in language modeling (n-gram model)

Example: Markov Chain Model (I)

- Markov assumption: a discrete-time Markov chain is a random sequence x[n] whose n-th conditional probability function satisfy:

  $$p(x[n] \mid x[n-1]x[n-2]\ldots x[n-N]) = p(x[n] \mid x[n-1])$$

- In other words, probability of observing x[n] only depends on its previous one x[n-1] (for 1st order Markov chain) or the most recent history (for higher order Markov chain).

- Parameters in Markov chain model are a set of conditional probability functions.

Example: Markov Chain Model (II)

- Stationary assumption:
  $p(x[n] \mid x[n-1]) = p(x[m] \mid x[m-1])$ for all n and m.

- For stationary discrete Markov Chain model:
  - Only one set of conditional probability function

- Discrete observation: in practice, the range of values taken on by each x[n] is finite, which is called state space. Each distinct one is a Markov state.
  - An observation of a discrete Markov chain model becomes a sequence of Markov states.
  - The set of conditional probabilities → transition matrix

Example: Markov Chain Model (III)

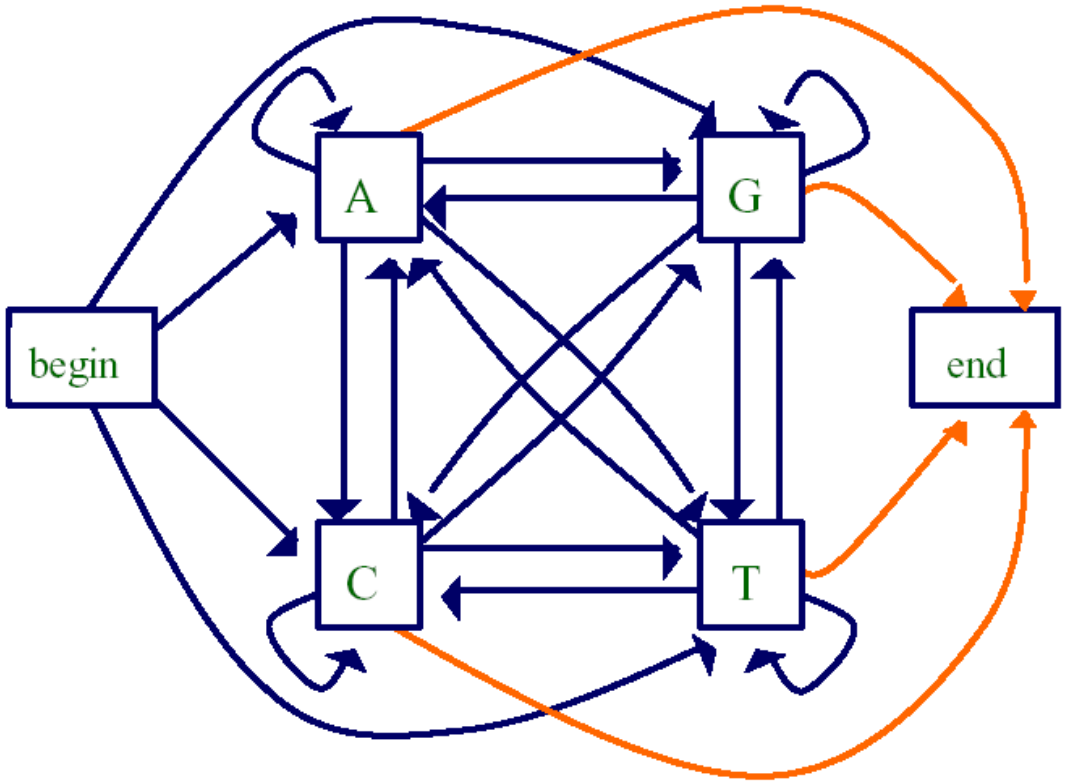
- Markov chain model (stationary & discrete):
  - A finite set of Markov states, to say M states.
  - A set of state conditional probabilities, i.e., transition matrix in 1st order Markov chain model, $a_{ij} = p(j|i)$ $(i,j=1,2,\ldots,M)$

- Markov chain model can be represented by a directed graph.
  - Node → Markov state
  - Arc → state transition (attached with a transition probability)
  - A Markov chain observation can be viewed as a path traversing

- Probability of observing a Markov chain can be calculated based on the path and the transition matrix.

Example: Markov Chain Model (IV)

- First-order Markov chain model for DNA sequence



Full Transition matrix (6 by 6)

p(A|G) = 0.16
p(C|G) = 0.34
p(G|G) = 0.38
p(T|G) = 0.12
…
…

One transition probability is attached with each arc.

Pr(GAATTC) = p(begin)p(G|begin)p(A|G)p(A|A)p(T|A)p(T|T)p(C|T)p(end|C)

Example: Markov Chain Model (V)

- Markov chain model for language modeling (n-gram)
  - Each word is a Markov state, total N words (vocabulary size)
  - A set of state (word) conditional probabilities
- Given any a sentence:

  S = I would like to fly from New York to Toronto this Friday
- 1st-order Markov chain model: N*N conditional probabilities

  Pr(S) = p(I|begin) p(would|I) p(like|would) p(to|like) p(fly|to) …
  - This is called bi-gram model
- 2nd-order Markov chain model: N*N*N

  Pr(S) = p(I|begin) p(would|I,begin) p(like|would,I) p(to|like,would) …
  - This is called tri-gram model
- Multinomial (0th-order Markov chain): N probabilities

  Pr(S) = p(I) p(would) p(like) p(to) p(fly) …
  - This is called uni-gram model

Example: Markov Chain Model (VI)

- How to estimate Markov chain model from training data
  - Similar to ML estimate of multinomial distribution
  - Maximization of log-likelihood function with constraints.
- Results:

$$p(W_i \mid W_j) = \frac{\text{Frequency of } W_j W_i \text{ in training data}}{\text{Frequency of } W_j \text{ in training data}}$$

$$p(W_i \mid W_j, W_k) = \frac{\text{Frequency of } W_k W_j W_i \text{ in training data}}{\text{Frequency of } W_k W_j \text{ in training data}}$$

- Generally n-gram model: a large number of probabilities