Pattern Classification (V)

杜俊
jundu@ustc.edu.cn

Outline

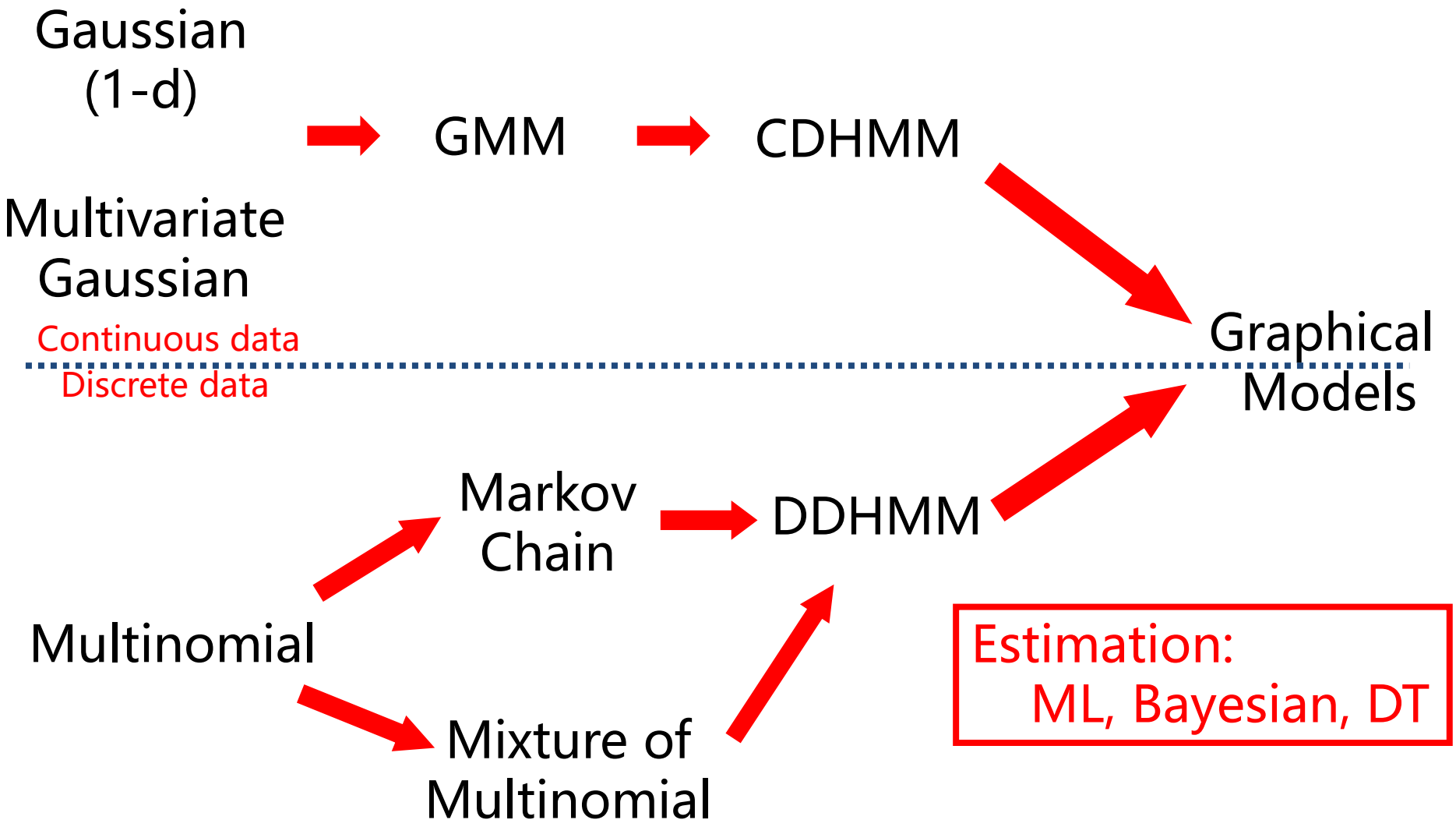
- Bayesian Decision Theory
  - How to make the optimal decision?
  - Maximum *a posterior* (MAP) decision rule
- Generative Models
  - Joint distribution of observation and label sequences
  - Model estimation: MLE, Bayesian learning, discriminative training
- Discriminative Models
  - Model the posterior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

Plug-in MAP Decision Rule

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

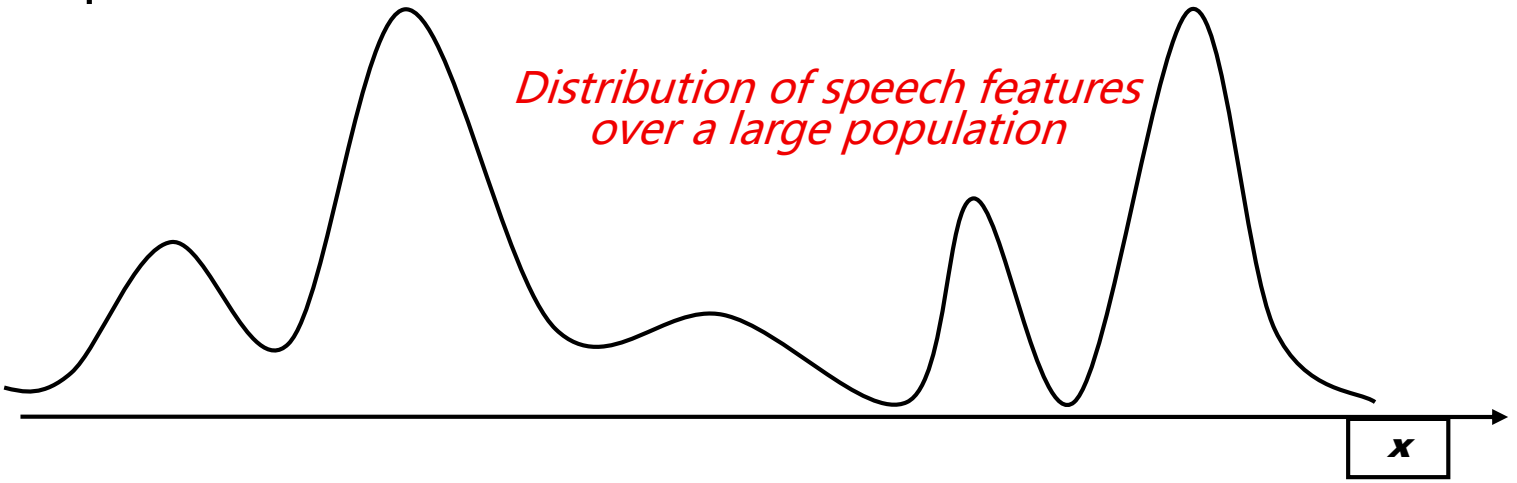
$$\approx \arg\max_{\mathrm{C}_i} \overline{P}_{\Gamma_i}(\mathrm{C}_i) \cdot \overline{p}_{\Lambda_i}(X \mid \mathrm{C}_i)$$

Statistical Models: Roadmap



Example: Gaussian Mixture Model (I)

- Gaussian distribution (univariate/multivariate) is a single mode distribution.
- In many cases, the true distribution of data is complicate and has multiple modes in nature.



- For this kind of applications, better to use a more flexible model
  - Gaussian mixture model (GMM)
  - A GMM can be tuned to approximate any arbitrary distribution

Example: Gaussian Mixture Model (II)

- Gaussian mixture model (GMM)
  - Univariate density

$$p(x) = \sum_{k=1}^{K} w_k \times N(x \mid m_k, s_k^2)$$

  - Multivariate density

$$p(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \cdot N(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$$

  - GMM is a mixture of single Gaussian distribution (each one is called mixture component) which have different means and variances.
  - ω_k is called mixture weight, prior probability of each mixture component.

$$\sum_{k=1}^{K} \omega_k = 1$$

- GMM is widely used for speaker recognition, audio classification, audio segmentation, etc.

Example: Gaussian Mixture Model (III)

- However, estimation of a GMM is not trivial.
- Consider a simple case:
  - We have a set of training data $D=\{x_1,x_2,\ldots,x_n\}$
  - Use a 2-mixture GMM to model it:

$$p(x)=\frac{0.3}{\sqrt{2ps_1^2}}e^{-\frac{(x-m_1)^2}{2s_1^2}}+\frac{0.7}{\sqrt{2ps_2^2}}e^{-\frac{(x-m_2)^2}{2s_2^2}}$$

  - We try to get ML estimate of μ_1, σ_1, μ_2, σ_2 from training data.
  - Simple maximization based on differential calculus does not work.
    - For each x_i, we do not know which mixture it comes from. The number of item in likelihood function $p(D|\ \mu_1, \sigma_1, \mu_2, \sigma_2\ )$ increases exponentially as we observe more and more data.
    - No simple solution.
- Need alternative method – Expectation Maximization (EM) algorithm

The Expectation-Maximization Algorithm (I)

- EM algorithm is an iterative method of obtaining maximum likelihood estimate of model parameters.

- EM suits best to the so-called *missing data* problem:
  - Only observe a subset of features, called *observed*, X.
  - Other features are *missing* or unobserved, denoted as Y.
  - The complete data Z={X,Y}.
  - If given the complete data Z, it is usually easy to obtain ML estimation of model parameters.
  - How to do ML estimation based on observed X only??

The Expectation-Maximization Algorithm (II)

- Initialization: find an initial values for unknown parameters

- EM algorithm consists of two steps:
  - Expectation (E-step): the expectation is calculated with respect of the missing data Y, using the current estimate of the unknown parameters and conditioned upon the observed X.

  - Maximization (M-step): provides a new estimate of unknown parameters (better than the initial ones) in terms of maximizing the above expectation → increasing likelihood function of observed.

- Iterate until convergence

EM Algorithm: E-step

- E-step: form an auxiliary function

$$Q(\theta;\theta^{(i)}) = E_Y\left[\ln p(X,Y\mid\theta) \mid X,\theta^{(i)}\right]$$

  - The expectation of log-likelihood function of complete data is calculated based on the current estimate of unknown parameter, and conditioned on the observed data.
  - $Q(q;q^{(i)})$ is a function of θ with $q^{(i)}$ assumed to be fixed.
  - If missing data Y is continuous:

$$Q(q;q^{(i)}) = \int_{L_Y} \ln p(X,Y\mid q)\times p(Y\mid X,q^{(i)})\,\mathrm{d}Y$$

  - If missing data Y is discrete:

$$Q(q;q^{(i)}) = \sum_{Y} \ln p(X,Y\mid q)\times p(Y\mid X,q^{(i)})$$

EM Algorithm: M-step

- M-step: choose a new estimate $q^{(i+1)}$ which maximizes $Q(q;q^{(i)})$

$$q^{(i+1)} = \arg\max_{q} Q(q;q^{(i)})$$

  - $q^{(i+1)}$ is a better estimate in terms of increasing likelihood value $p(X \mid \theta)$ than $q^{(i)}$

$$p(X \mid \theta^{(i+1)}) \geq p(X \mid \theta^{(i)})$$

- Replace $q^{(i+1)}$ with $q^{(i)}$ and iterate until convergence.

EM Algorithm

- EM algorithm guarantees that the log-likelihood of the observed data $p(X|\theta)$ will increase monotonically.

- EM algorithm may converge to a local maximum or global maximum. And convergence rate is reasonably good.

- Applications of the EM algorithm:
  - ML estimation of some complicated models,
    e.g., GMM, HMM, … (in general mixture models of e-family)
  - ET (emission tomography) image reconstruction
  - Active Noise Cancellation (ANC)
  - Spread-spectrum multi-user communication

An Application of EM Algorithm: ML Estimation of Multivariate GMM(I)

- Assume we observe a data set $D=\{X_1,X_2,\ldots,X_T\}$ (a set of vectors)
- We decide to model the data by using multivariate GMM:

$$p(X)=\sum_{k=1}^{K} w_k \times N(X \mid m_k, \Sigma_k) \qquad (\text{with} \sum_{k=1}^{K} w_k = 1)$$

- Problem: use data set D to estimate GMM model parameters, including ω_k, μ_k, Σ_k ($k=1,2,\ldots,K$).
- If we know the label of mixture component label l_t from which each data X_t come from, the estimation is easy.
- Since the mixture component is not available in training set, we treat it as missing data:
  - Observed data: $D=\{X_1,X_2,\ldots,X_T\}$.
  - Missing data: $L=\{l_1, l_2, \ldots, l_T\}$.
  - Complete data: $\{D,L\} = \{X_1, l_1, X_2, l_2, \ldots, X_T, l_T\}$

An Application of EM algorithm: ML estimation of Multivariate GMM(II)

- E-step:

$$Q(\{\omega_k,\mu_k,\Sigma_k\}\,|\,\{\omega_k^{(i)},\mu_k^{(i)},\Sigma_k^{(i)}\}) = \sum_L \ln p(D,L\,|\{\omega_k,\mu_k,\Sigma_k\})\cdot p(L\,|\,D,\{\omega_k^{(i)},\mu_k^{(i)},\Sigma_k^{(i)}\})$$

$$= C + \sum_L\left[\sum_{t=1}^{T}[\ln\omega_{l_t} - \frac{1}{2}\ln|\Sigma_{l_t}| - \frac{1}{2}\cdot(X_t-\mu_{l_t})^{\mathrm{T}}\Sigma_{l_t}^{-1}(X_t-\mu_{l_t})]\cdot\prod_{t=1}^{T}p(l_t\,|\,X_t,\{\omega_k^{(i)},\mu_k^{(i)},\Sigma_k^{(i)}\})\right]$$

$$= C + \sum_{k=1}^{K}\sum_{t=1}^{T}[\ln\omega_k - \frac{1}{2}\ln|\Sigma_k| - \frac{1}{2}\cdot(X_t-\mu_k)^{\mathrm{T}}\Sigma_k^{-1}(X_t-\mu_k)]\cdot p(l_t=k\,|\,X_t,\omega_k^{(i)},\mu_k^{(i)},\Sigma_k^{(i)})$$

An Application of EM Algorithm: ML Estimation of Multivariate GMM(III)

- M-step:

$$\frac{\partial Q}{\partial \mu_k}=0 \Rightarrow \mu_k^{(i+1)}=\frac{\sum_{t=1}^{T} X_t \cdot p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}{\sum_{t=1}^{T} p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}$$

$$\frac{\partial Q}{\partial \Sigma_k}=0 \Rightarrow \Sigma_k^{(i+1)}=\frac{\sum_{t=1}^{T} (X_t-\mu_k^{(i+1)})^{\mathrm{T}} \cdot (X_t-\mu_k^{(i+1)}) \cdot p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}{\sum_{t=1}^{T} p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}$$

$$\frac{\partial}{\partial \omega_k}[Q-\lambda(\sum_{k=1}^{K}\omega_k-1)]=0 \Rightarrow \omega_k=\frac{\sum_{t=1}^{T} p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}{\sum_{k=1}^{K}\sum_{t=1}^{T} p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}=\frac{\sum_{t=1}^{T} p(l_t=k \mid X_t, \omega_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)})}{T}$$

An Application of EM Algorithm: ML Estimation of Multivariate GMM(IV)

- where

$$p(l_t = k \mid X_t, \{w_k^{(i)}, m_k^{(i)}, \Sigma_k^{(i)}\}) = \frac{w_k^{(i)} \times N(X_t \mid m_k^{(i)}, \Sigma_k^{(i)})}{\sum_{k=1}^{K} w_k^{(i)} \times N(X_t \mid m_k^{(i)}, \Sigma_k^{(i)})}$$

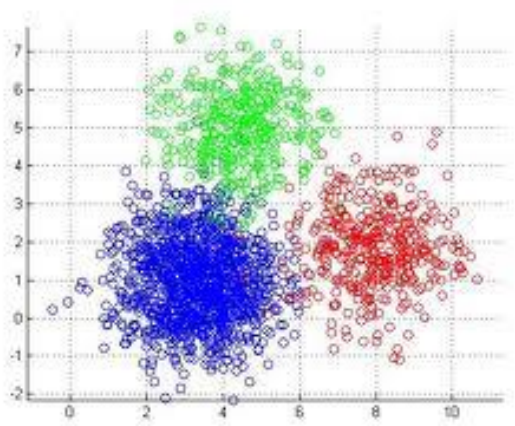
- Iterative ML estimation of GMM
  - Initiation: choose $\{w_k^{(0)}, m_k^{(0)}, \Sigma_k^{(0)}\}$. Usually use vector clustering algorithm (such as K-means) to cluster all data into K clusters. Each cluster is used to train for one Gaussian mix.
  - i=0;
  - Use EM algorithm to refine model estimation

$$\{w_k^{(i)}, m_k^{(i)}, \Sigma_k^{(i)}\} \Rightarrow \{w_k^{(i+1)}, m_k^{(i+1)}, \Sigma_k^{(i+1)}\}$$

  - i++, go back until convergence.

GMM Initialization: K-Means Clustering

- K-Means Clustering: a.k.a. unsupervised learning
- Cluster a data set into many homogeneous groups
- K-Means algorithm:
  - step 1: assign all data into one group; calculate centroid.
  - step 2: choose a group and split.
  - step 3: re-assign all data to groups.
  - step 4: calculate centroids for all groups.
  - step 5: go back to step 3 until convergence.
  - step 6: stop until K classes
- Basics for clustering:
  - distance measure
  - centroid calculation
  - choose a group and split



Applications of GMM

- GMM is widely used to model speech or audio signals. In many cases, we use diagonal covariance matrices in GMM for simplicity.
- Speaker recognition:
  - Collect some speech signals from all known speakers.
  - Train a GMM for each known speaker by using his/her voice.
  - For an unknown speaker, prompt him/her to say sth.
  - Classify it based on all trained GMM’ s and determine the speaker’ s identity or reject.
- Audio classification:
  - Classify a continuous audio/video stream (from radio or TV) into some homogeneous segments: anchor’ s speech, in-field interview, telephone interview, music, commercial ads, sports, etc.
  - For each category, train a GMM based on training data.
  - Use all trained GMMs to scan an unknown audio stream to segment it.

Project: Building a 2-Class Classifier

- Given some data from two classes
- Build a classifier with multivariate Gaussian models
  - ML estimation
  - Test with the plug-in MAP decision rule
- Improve it with GMM models
  - Initialize GMM with the K-means clustering
  - Estimate GMM with the EM algorithm
  - Investigate GMM with the mixture number = 2, 4, 8.
- Improve the Gaussian classifier with discriminative training (minimum classification error estimation)
- Preferably programming with C/C++
- Report all of your experiments and your best classifier.