Pattern Classification (VI)

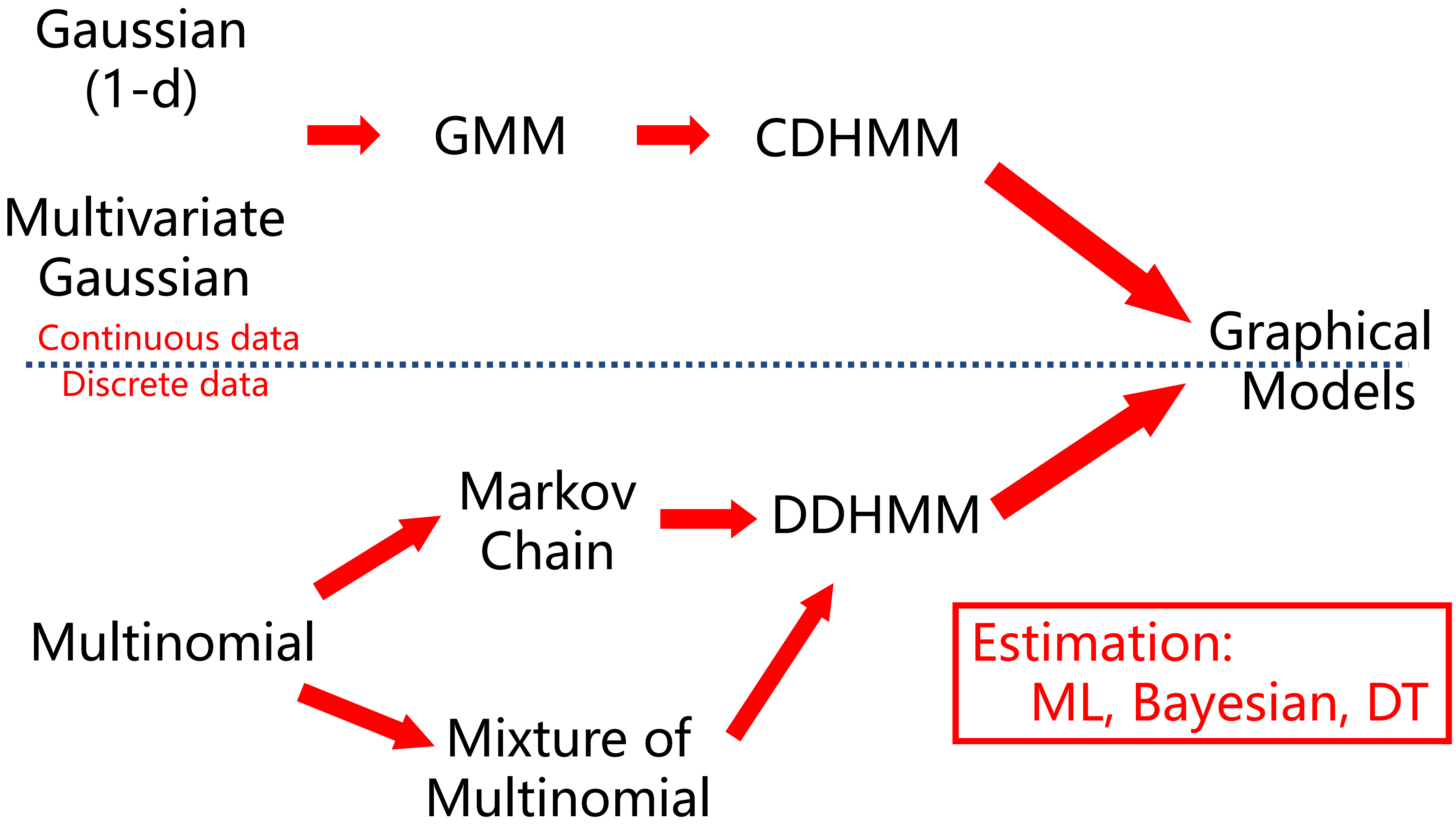
杜俊

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

Statistical Models: Roadmap



Gaussian (1-d)

GMM

CDHMM

Multivariate Gaussian

Continuous data

Discrete data

Graphical Models

Markov Chain

DDHMM

Multinomial

Mixture of Multinomial

Estimation:
ML, Bayesian, DT

Model Parameter Estimation

- Maximum Likelihood (ML) Estimation:
  - ML method: most popular model estimation
  - EM (Expected-Maximization) algorithm
  - Examples:
    - Univariate Gaussian distribution
    - Multivariate Gaussian distribution
    - Multinomial distribution
    - Gaussian Mixture model
    - Markov chain model: n-gram for language modeling
    - Hidden Markov Model (HMM)

- Discriminative Training
  - Minimum Classification Error (MCE)
  - Maximum Mutual Information (MMI)
- Bayesian Model Estimation: Bayesian theory

Minimum Classification Error Estimation (I)

- In a N-class pattern classification problem, given a set of training data, D={ (X_1, C_1), (X_2, C_2), …, (X_T, C_T)}, estimate model parameters for all class to minimize total classification errors in D.
  - MCE: minimize empirical classification errors
- Objective function ➔ total classification errors in D
  - For each training data, (X_t, C_t), define misclassification measure:

$$d(X_t, C_t) = -p(C_t)\,p(X_t \mid \lambda_{C_t}) + \max_{C \neq C_t} p(C)\,p(X_t \mid \lambda_C)$$

or

$$d(X_t, C_t) = -\ln[p(C_t)\,p(X_t \mid \lambda_{C_t})] + \max_{C \neq C_t} \ln[p(C)\,p(X_t \mid \lambda_C)]$$

If d(X_t, C_t)>0, incorrect classification for X_t ➔ 1 error

If d(X_t, C_t)<=0, correct classification for X_t ➔ 0 error

Minimum Classification Error Estimation (II)

- Soft-max: approximate d(X$_t$, C$_t$) by a differentiable function:

$$d(X_t, C_t) \approx -p(C_t)p(X_t \mid \lambda_{C_t}) + \ln\left[\frac{1}{N-1}\sum_{C \neq C_t} \exp[\eta \cdot p(C)p(X_t \mid \lambda_C)]\right]^{1/\eta}$$

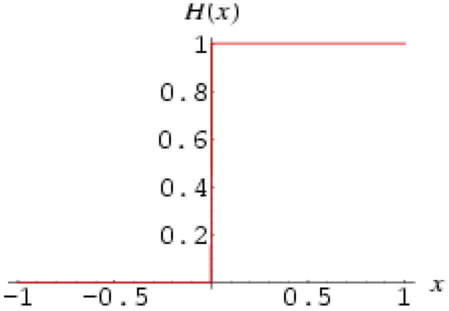
or

$$d(X_t, C_t) \approx -\ln[p(C_t)p(X_t \mid \lambda_{C_t})] + \ln\left[\frac{1}{N-1}\sum_{C \neq C_t} \exp[\eta \cdot \ln(p(C)p(X_t \mid \lambda_C))]\right]^{1/\eta}$$

where η>1.

Minimum Classification Error Estimation (III)

- Error count for one data (X_t, C_t), is a step function H(d(X_t, C_t))
- Total errors in training set:



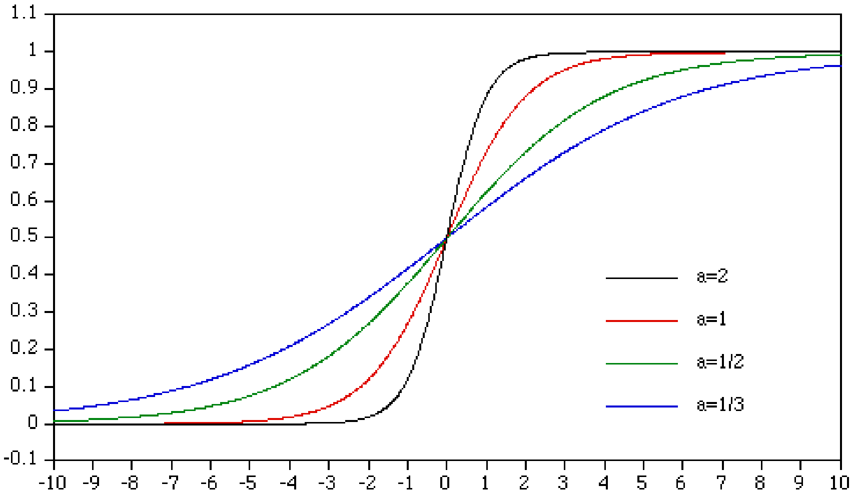
$$Q(\Lambda)=\sum_{t=1}^{T}H(d(X_t,C_t))$$

- Step function is not differentiable, approximated by a sigmoid function ➔ smoothed total errors in training set.

$$Q(\Lambda)\approx Q'(\Lambda)=\sum_{t=1}^{T}l(d(X_t,C_t))$$

where $$l(d)=\frac{1}{1+e^{-a\times d}}$$

a>0 is a parameter to control its shape.



Minimum Classification Error Estimation (IV)

- MCE estimation of model parameters for all classes:

$$\{\lambda_1 \cdots \lambda_N\}_{\mathrm{MCE}} = \underset{\lambda_1 \cdots \lambda_N}{\arg\min}\; Q'(\lambda_1 \cdots \lambda_N)$$

- Optimization: no simple solution is available
  - Iterative gradient descent method.
    - Stochastic GD, batch mode, mini-batch mode

$$l_i^{(n+1)} = l_i^{(n)} - e \times \frac{¶}{¶ l_i} Q'(l_1 \cdots l_N)|_{l_i = l_i^{(n)}}$$

Minimum Classification Error Estimation (V)

- Find initial model parameters, e.g., ML estimates
- Calculate gradient of the objective function
- Calculate the value of the gradient based on the current parameters
- Update model parameters

$$/_i^{(n+1)} = /_i^{(n)} - e \times \frac{\P}{\P /_i} Q'(/_1 \cdots /_N)\big|_{/_i = /_i^{(n)}}$$

- Iterate until convergence

How to Calculate Gradient?

$$\frac{\partial}{\partial \lambda_i} Q'(\lambda_1 \cdots \lambda_N) = \sum_{t=1}^{T} \frac{\partial}{\partial \lambda_i} l\left[d(X_t, C_t)\right]$$

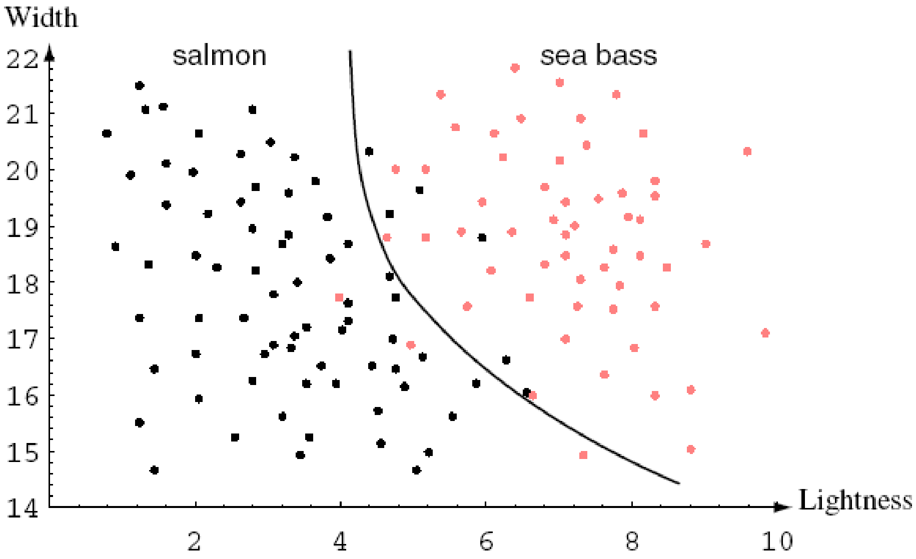
$$= \sum_{t=1}^{T} \frac{\partial l(d)}{\partial d} \cdot \frac{\partial d(X_t, C_t)}{\partial \lambda_i}$$

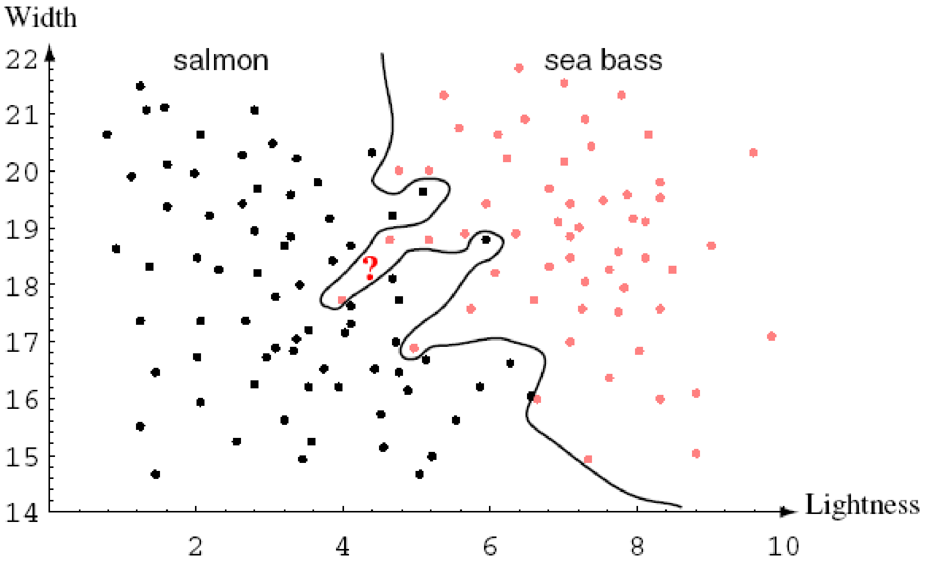
$$= \sum_{t=1}^{T} a \cdot l(d) \cdot [1 - l(d)] \cdot \frac{\partial d(X_t, C_t)}{\partial \lambda_i}$$

- The key issue in MCE training is to set a proper step size experimentally.

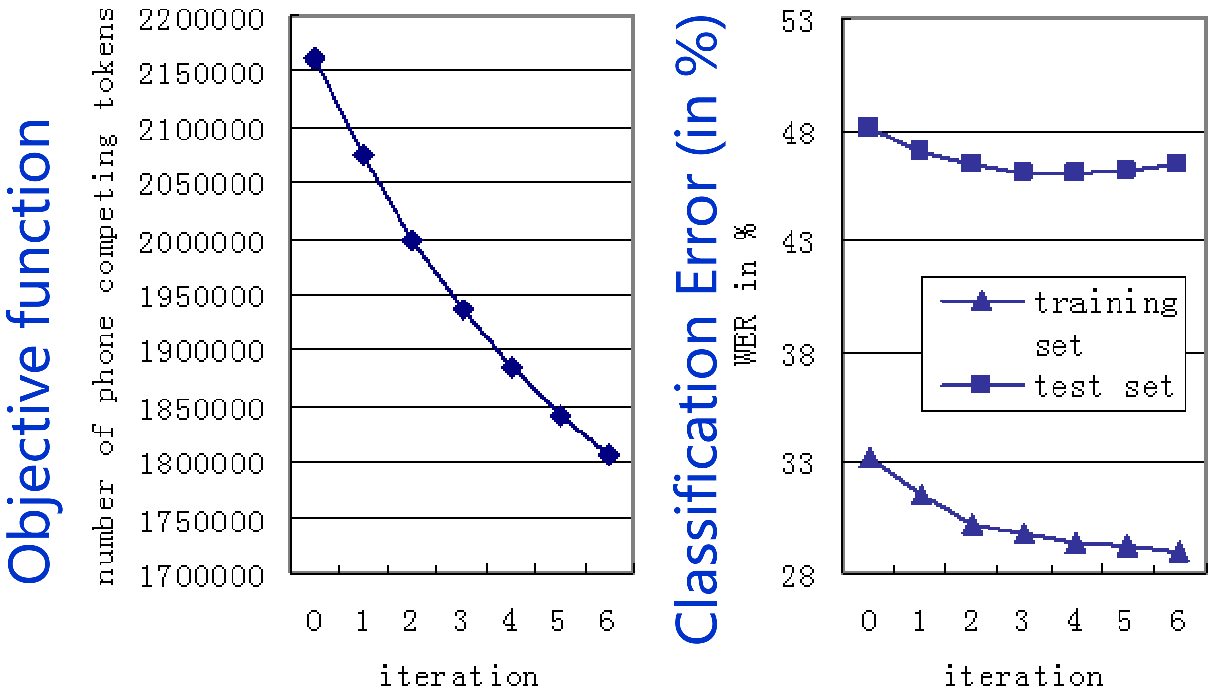
Overtraining (Overfitting)

- Low classification error rate in training set does not always lead to a low error rate in a new test set due to overtraining.





Measuring Performance of MCE



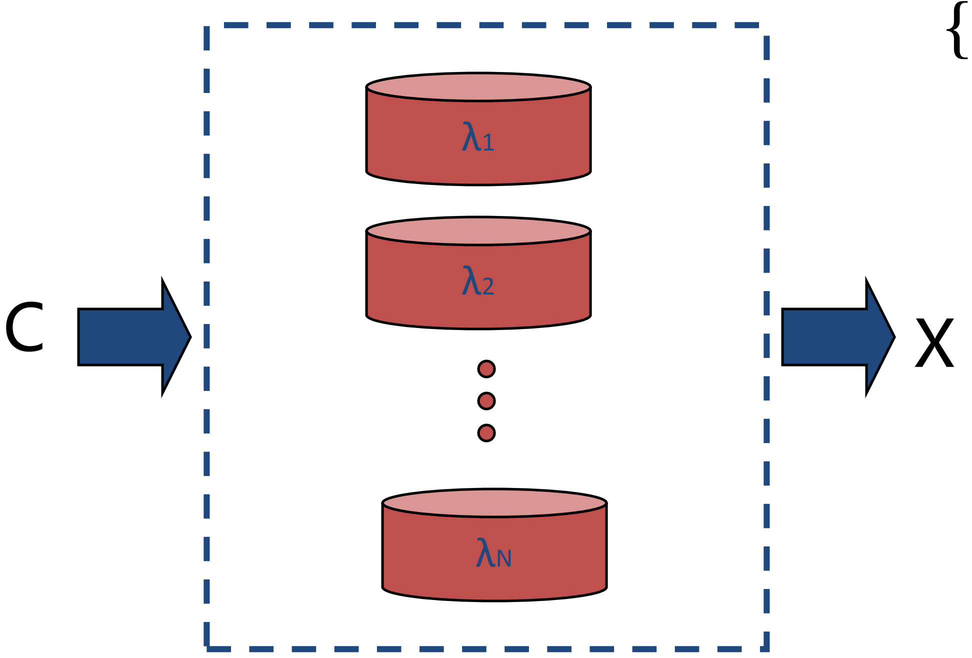
- When to converge: monitor three quantities in the MCE
  - The objective function
  - Error rate in training set
  - Error rate in test set

Maximum Mutual Information Estimation (I)

- The model is viewed as a noisy data generation channel

  Class id C ➔ observation feature X

- Maximize mutual information between C and X



noisy data generation channel

$$\{\lambda_1\cdots\lambda_N\}_{\mathrm{MMI}} = \underset{\lambda_1\cdots\lambda_N}{\arg\max}\ I(C,X)$$

$$I(C,X) = \sum_C\sum_X p(C,X)\log_2\frac{p(C,X)}{p(C)p(X)}$$

$$= \sum_C\sum_X p(C,X)\log_2\frac{p(X\,|\,C)}{p(X)}$$

$$= \sum_C\sum_X p(C,X)\log_2\frac{p(X\,|\,C)}{\sum_C p(X\,|\,C)}$$

$$= \sum_C\sum_X p(C,X)\log_2\frac{p(X\,|\,\lambda_C)}{\sum_C p(X\,|\,\lambda_C)}$$

Maximum Mutual Information Estimation (II)

- Difficulty: joint distribution p(C,X) is unknown.
- Solution: collect a representative training set $(X_1, C_1), (X_2, C_2), \ldots, (X_T, C_T)$ to approximate the joint distribution.

$$\{\lambda_1 \cdots \lambda_N\}_{\mathrm{MMI}} = \arg\max_{\lambda_1 \cdots \lambda_N} \; I(C, X)$$

$$= \arg\max_{\lambda_1 \cdots \lambda_N} \sum_{C} \sum_{X} p(C, X) \log_2 \frac{p(X \mid \lambda_C)}{\sum_{C} p(X \mid \lambda_C)}$$

$$\approx \arg\max_{\lambda_1 \cdots \lambda_N} \sum_{t=1}^{T} \log_2 \frac{p(X_t \mid \lambda_{C_t})}{\sum_{C} p(X_t \mid \lambda_C)}$$

- Optimization:
  - Iterative gradient-ascent method
  - Growth-transformation method

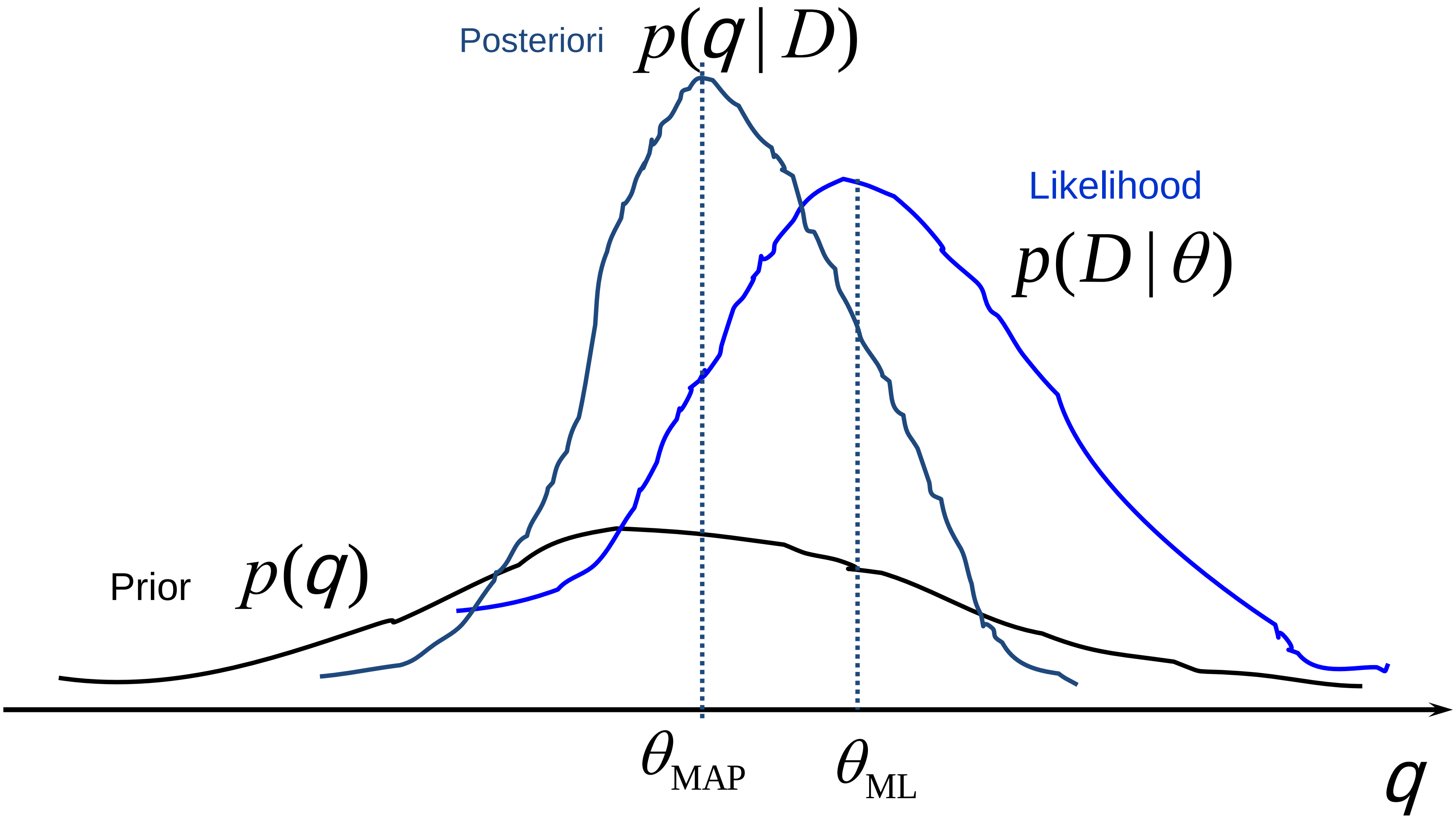
Bayesian Model Estimation

- Bayesian methods view model parameters as random variables having some known prior distribution. (Prior specification)
  - Specify prior distribution of model parameters θ as p(θ).

- Training data D allow us to convert the prior distribution into a posteriori distribution. (Bayesian learning)

$$p(q\,|\,D) = \frac{p(q)\times p(D\,|\,q)}{p(D)}\ \mu\ \ p(q)\times p(D\,|\,q)$$

Bayesian Learning



MAP Estimation

- Do a point estimate about θ based on the posteriori distribution

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta}\ p(\theta \mid D) = \arg\max_{\theta}\ p(\theta)\cdot p(D \mid \theta)$$

- Then θ_{MAP} is treated as estimate of model parameters (just like ML estimate). Sometimes need the EM algorithm to derive it.

- MAP estimation optimally combine prior knowledge with new information provided by data.

- MAP estimation is used in speech recognition to adapt speech models to a particular speaker to cope with various accents
  - From a generic speaker-independent speech model ➔ prior
  - Collect a small set of data from a particular speaker
  - The MAP estimate give a speaker-adaptive model which suits better to this particular speaker.

How to Specify Priors

- Noninformative priors
  - Without enough prior knowledge, just use a flat prior

- Conjugate priors: for computation convenience
  - After Bayesian leaning the posterior will have the exact same function form as the prior except the all parameters are updated.
  - Not every model has conjugate prior.

Conjugate Prior

- For a univariate Gaussian model with only unknown mean:

$$p(x) = N(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp[-\frac{(x-\mu)^2}{2\sigma^2}]$$

- The conjugate prior of Gaussian is Gaussian

$$p(m) = N(m| m_0, s_0^2) = \frac{1}{\sqrt{2ps_0^2}}\exp[- \frac{(m- m_0)^2}{2s_0^2}]$$

- After observing a new data x1, the posterior will still be Gaussian:

$$p(\mu|x_1) = N(\mu|\mu_1,\sigma_1^2) = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp[-\frac{(\mu-\mu_1)^2}{2\sigma_1^2}]$$

where $$\mu_1 = \frac{\sigma_0^2}{\sigma_0^2+\sigma^2}x_1 + \frac{\sigma^2}{\sigma_0^2+\sigma^2}\mu_0$$

$$\sigma_1^2 = \frac{\sigma_0^2\sigma^2}{\sigma_0^2+\sigma^2}$$

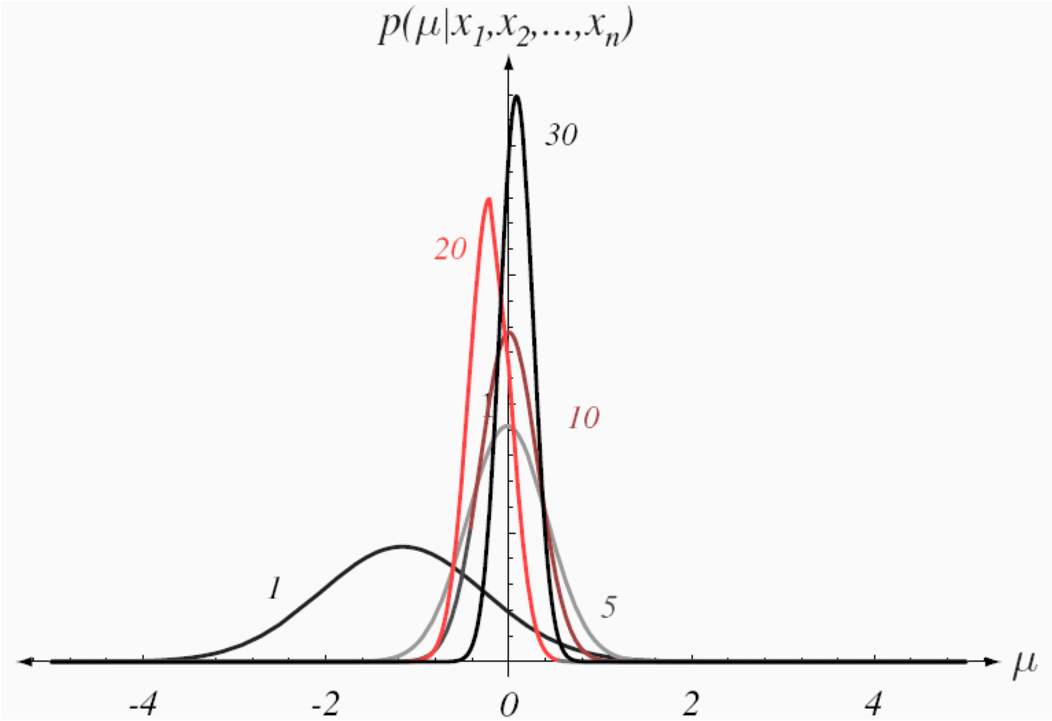
The Sequential MAP Estimate of Gaussian

- For univariate Gaussian with unknown mean, the MAP estimate of its mean after observing x1:

$$m_1 = \frac{s_0^2}{s_0^2 + s^2} x_1 + \frac{s^2}{s_0^2 + s^2} m_0$$

- After observing next data x2:

$$m_2 = \frac{s_1^2}{s_1^2 + s^2} x_2 + \frac{s^2}{s_1^2 + s^2} m_1$$


$p(\mu|x_1,x_2,\ldots,x_n)$
30
20
10
1
5
1
μ
-4
-2
0
2
4