9.0 Some Fundamental Problem-solving Approaches

References: 1. 4.3.1, 4.3.2, 4.4.2 of Huang, or 9.1-9.3 of Jelinek

2. 6.4.3 of Rabiner and Juang

3. " The Expectation-Maximization Algorithm", IEEE Signal Processing Magazine, Nov 1996

4. " Minimum Classification Error Rate Methods for Speech Recognition", IEEE Trans. Speech and Audio Processing, May 1997

EM (Expectation and Maximization) Algorithm

- **Goal**

  estimating the parameters for some probabilistic models based on some criteria

- **Parameter Estimation Principles given some observations $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\ldots, \boldsymbol{x}_N]$:**
  - Maximum Likelihood (ML) Principle
    find the model parameter set θ such that the likelihood function is maximized, $P(\boldsymbol{X}|\theta)=\max$.
    - For example, if $\theta=\{\mu,\Sigma\}$ is the parameters of a normal distribution, and **X** is i.i.d, then the ML estimate of $\theta=\{\mu,\Sigma\}$ is

$$\mu_{ML}=\frac{1}{N}\sum_{i=1}^{N}x_i \ , \ \Sigma_{ML}=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\mu_{ML}\right)\left(x_i-\mu_{ML}\right)^t$$

  - the Maximum A Posteriori (MAP) Principle
    - Find the model parameter θ so that the A Posterior probability is maximized

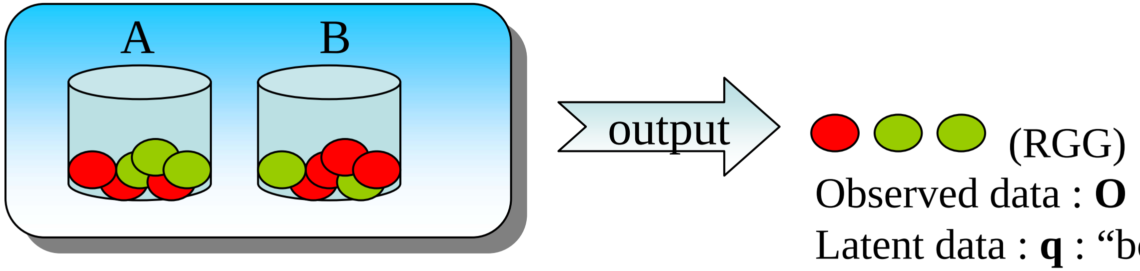
      i.e. $P(\theta|\boldsymbol{X})=P(\boldsymbol{X}|\theta)\,P(\theta)/P(\boldsymbol{X})=\max$

      $\Rightarrow P(\boldsymbol{X}|\theta)\,P(\theta)=\max$

EM (Expectation and Maximization) Algorithm

- **Why EM?**
  - In some cases the evaluation of the objective function (e.g. likelihood function) depends on some intermediate variables (latent data) which are not observable (e.g. the state sequence for HMM parameter training)
  - direct estimation of the desired parameters without such latent data is impossible or difficult
    e.g. almost impossible to estimate $\{A, B, \pi\}$ for HMM without considerations on the state sequence
- **Iteractive Procedure with Two Steps in Each Iteration:**
  - ***E*** (Expectation): expectation with respect to the possible distribution (values and probabilities) of the latent data based on the current estimates of the desired parameters conditioned on the given observations
  - ***M*** (Maximization): generating a new set of estimates of the desired parameters by maximizing the objective function (e.g. according to ML or MAP)
  - the objective function increased after each iteration, eventually converged

EM Algorithm: An example



Observed data : **O** : “ball sequence”: RGG
Latent data : **q** : “bottle sequence”: AAB

Parameter to be estimated : λ ={P(A),P(B),P(R|A),P(G|A), P(R|**B**), P(G|B)}

- **First, randomly assigned** $\lambda^{(0)}$={$P^{(0)}$(A),$P^{(0)}$(B),$P^{(0)}$(R|A),$P^{(0)}$(G|A), $P^{(0)}$(R|B), $P^{(0)}$(G|B)}
  for example :
  {$P^{(0)}$(A)=0.4,$P^{(0)}$(B)=0.6,$P^{(0)}$(R|A)=0.5,$P^{(0)}$(G|A) =0.5, $P^{(0)}$(R|B) =0.5, $P^{(0)}$(G|B) =0.5}
- **Expectation Step : find the *expectation* of logP(O| λ)**
  8 possible state sequences q_i :{AAA},{BBB},{AAB},{BBA},{ABA},{BAB},{ABB},{BAA}

$$E_{\mathbf{q}}\left(\log P(O|\lambda)\right)=\sum_{i=1}^{8}\log P(\mathbf{O},\mathbf{q}_i|\lambda)P\left(\mathbf{q}_i|\mathbf{O},\lambda^{(0)}\right)=\sum_{i=1}^{8}\log P(\mathbf{O},\mathbf{q}_i|\lambda)\frac{P\left(\mathbf{O},\mathbf{q}_i|\lambda^{(0)}\right)}{P\left(\mathbf{O}|\lambda^{(0)}\right)}=\frac{1}{P\left(\mathbf{O}|\lambda^{(0)}\right)}\sum_{i=1}^{8}\log P(\mathbf{O},\mathbf{q}_i|\lambda)P\left(\mathbf{O},\mathbf{q}_i|\lambda^{(0)}\right)$$

For example, when q_i = {AAB}

$$P\left(\mathbf{O}=RGG,\mathbf{q}_i=AAB|\lambda^{(0)}\right)=P\left(\mathbf{O}=RGG|\mathbf{q}_i=AAB,\lambda^{(0)}\right)P\left(\mathbf{q}_i=AAB|\lambda^{(0)}\right)$$
$$=\left[P^{(0)}(R|A)P^{(0)}(G|A)P^{(0)}(G|B)\right]\left[P^{(0)}(A)P^{(0)}(A)P^{(0)}(B)\right]=0.5*0.5*0.5*0.4*0.4*0.6\quad(\text{known values})$$
$$\log P\left(\mathbf{O}=RGG,\mathbf{q}_i=AAB|\lambda\right)=\log\left[P(R|A)P(G|A)P(G|B)\right]\left[P(A)P(A)P(B)\right](\text{with unknown parameters})$$

- **Maximization Step : find $\lambda^{(1)}$ to maximize the expectation function** E_q(logP(O| λ))
- **Iterations :** $\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \lambda^{(2)} \rightarrow$....

EM Algorithm

- **In Each Iteration (assuming logP(x |θ) is the objective function)**
  - E step: expressing the log-likelihood logP($\boldsymbol{x}$|θ) in terms of *the distribution of the latent data conditioned on* [x, $\theta^{(k)}$]
  - M step: find a way to maximized the above function, such that the above function increases monotonically, i.e., $\log P(\boldsymbol{x}|\theta^{(k+1)}) \geq \log P(\boldsymbol{x}|\theta^{(k)})$
- **The Conditions for the Iterations to Converge**
  - $\boldsymbol{x}$: observed (incomplete) data, $\boldsymbol{z}$: latent data, {$\boldsymbol{x}$, $\boldsymbol{z}$} : complete data

$$p(x, z|\theta) = p(z|x, \theta)p(x|\theta)$$

$$\Rightarrow \log p(x|\theta) = \log p(x, z|\theta) - \log p(z|x, \theta)$$

$$\text{assuming } z \text{ is generated based on } p\left(z|x, \theta^{[k]}\right),$$

$$E_z[\log p(x|\theta)] = E_z[\log p(x, z|\theta)] - E_z[\log p(z|x, \theta)]$$

$$= \int \log p(x, z|\theta)\, p\left(z|x, \theta^{[k]}\right)dz - \int \log p(z|x, \theta)\, p\left(z|x, \theta^{[k]}\right)dz$$

$$= Q\left(\theta, \theta^{[k]}\right) - H\left(\theta, \theta^{[k]}\right)$$

EM Algorithm

- **For the EM Iterations to Converge:**

$$E_z[\log p(x|\theta)] = E_z[\log p(x, z|\theta)] - E_z[\log p(z|x,\theta)]$$

$$= \int \log p(x, z|\theta)\, p\left(z|x, \theta^{[k]}\right) dz - \int \log p\,(z|x,\theta)\, p\,\left(z|x,\theta^{[k]}\right) dz$$

$$= Q\left(\theta, \theta^{[k]}\right) - H\left(\theta, \theta^{[k]}\right)$$

  – to make sure $\log P(\boldsymbol{x}|\theta^{[k+1]}) \geq \log P(\boldsymbol{x}|\theta^{[k]})$

$$\Rightarrow Q\left(\theta^{[k+1]}, \theta^{[k]}\right) - Q\left(\theta^{[k]}, \theta^{[k]}\right) - H\left(\theta^{[k+1]}, \theta^{[k]}\right) + H\left(\theta^{[k]}, \theta^{[k]}\right) \geq 0$$

  – $H(\theta^{[k+1]},\theta^{[k]}) \leq H(\theta^{[k]},\theta^{[k]})$ due to Jenson's Inequality

$$\sum_i p_i \log p_i \geq \sum_i p_i \log q_i \text{ , or } \sum_i p_i \log p_i - \sum_i p_i \log q_i \geq 0$$

$$= \text{when } p_i = q_i$$

  – the only requirement for convergence is to have $\theta^{[k+1]}$ such that

$$Q(\theta^{[k+1]},\theta^{[k]}) - Q(\theta^{[k]},\theta^{[k]}) \geq 0$$

  – $Q(\theta,\theta^{[k]})$: auxiliary function, or Q-function, the expectation of the objective function in terms of the distribution of the latent data conditioned on $(x,\theta^{[k]})$

Example: Use of EM Algorithm in Solving Problem 3 of HMM

- **Observed data :** ***observations* O, latent data :** ***state sequence* q**
- **The probability of the complete data is**

$$P(\mathbf{O},\mathbf{q}|\lambda) = P(\mathbf{O}|\mathbf{q},\lambda)P(\mathbf{q}|\lambda)$$

- **E-Step :**

  $Q(\lambda, \lambda^{[k]}) = E[\log P(\mathrm{O},\mathrm{q}|\lambda)|\mathrm{O}, \lambda^{[k]}] = \sum_{\mathrm{q}} P(\mathrm{q}|\mathrm{O}, \lambda^{[k]})\log[P(\mathrm{O},\mathrm{q}|\lambda)]$
  - $\lambda^{[k]}$: k-th estimate of λ (known), λ: unknown parameter to be estimated
- **M-Step :**
  - Find $\lambda^{[k+1]}$ such that $\lambda^{[k+1]} = \arg\max_{\lambda} Q(\lambda, \lambda^{[k]})$
- **Given the Various Constraints** (e.g. $\sum_i \pi_i = 1, \sum_j a_{ij} = 1,\ etc.$), **It can be shown**
  - the above maximization leads to the formulas obtained previously
  - $P(O|\lambda^{[k+1]}) \geq P(O|\lambda^{[k]})$

Minimum-Classification-Error (MCE) Training

- **General Objective : find an optimal set of parameters (e.g. for recognition models) to *minimize the expected error of classification***
  – the statistics of test data may be quite different from that of the training data
  – training data is never enough
- **Assume the recognizer is operated with the following classification principles :**

  $\{C_i, i=1,2,...M\}$, M classes
  $\lambda^{(i)}$: statistical model for C_i
  $\Lambda=\{\lambda^{(i)}\}_{i=1......M}$, the set of all models for all classes
  X : observations
  $g_i(X,\Lambda)$: class conditioned likelihood function, for example,
  $g_i(X,\Lambda) = P\ (X|\lambda^{(i)})$
  – $\mathbf{C}(X) = \boldsymbol{C_i}$ if $g_i(X,\Lambda) = \max_j g_j(X,\Lambda)$: classification principles
  an error happens when $P(X|\lambda^{(i)}) = \max$ but $X \notin \boldsymbol{C_i}$
- **Conventional Training Criterion :**

  find $\lambda^{(i)}$ such that $P(X|\lambda^{(i)})$ is maximum (Maximum Likelihood) if $X \in \boldsymbol{C_i}$
  – This does not always lead to minimum classification error, since it doesn't consider the mutual relationship among competing classes
  – The competing classes may give higher likelihood function for the test data

Minimum-Classification-Error (MCE) Training

- **One form of the misclassification measure**

$$d_i(X,\Lambda) = -g_i(X,\Lambda) + \left[\frac{1}{M-1}\sum_{j\neq i} g_j(X,\Lambda)^{\alpha}\right]^{\frac{1}{\alpha}} \qquad X \in C_i$$

  – Comparison between the likelihood functions for the correct class and the competing classes

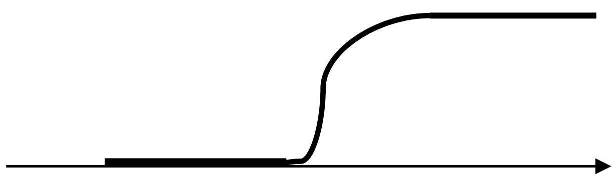
  $\alpha = 1$ all other classes included and averaged with equal weights
  $\alpha \to \infty$ only the most competing one considered
  $d_i(X) \geq 0$ implies a classification error
  $d_i(X) < 0$ implies a correct classification

- **A continuous loss function is defined**

$$l_i(X,\Lambda) = l(d_i(X,\Lambda)),\ \mathrm{X} \in \mathrm{C_i}$$

$$l(d) = \frac{1}{1+\exp[-\gamma(d-\theta)]},\ \textit{sigmoid function}$$

  – l(d) $\to$ 0 when d $\to -\infty$
  l(d) $\to$ 1 when d $\to \infty$
  $\theta = 0$ switching from 0 to 1 near θ
  γ : determining the slope at switching point



- **Overall Classification Performance Measure :**

$$L(\Lambda) = E_X[L(X,\Lambda)] = \sum_X [L(X,\Lambda)] = \sum_X \sum_{i=1}^{M} l_i(X,\Lambda)\,\delta(X \in C_i)$$

$$\delta(X \in C_i) = \begin{cases} 1 & \text{if } X \in C_i \\ 0 & \text{otherwise} \end{cases}$$

Minimum-Classification-Error (MCE) Training

- **Find $\hat{\Lambda}$ such that**

$$\hat{\Lambda} = \arg\min_{\Lambda} L(\Lambda) = \arg\min_{\Lambda} E_X\left[L(X, \Lambda)\right]$$

  - the above objective function in general is difficult to minimize directly
  - local minimum can be obtained iteratively using gradient (steepest) descent algorithm

$$\Lambda_{t+1} = \Lambda_t - \varepsilon_t \nabla L(\Lambda_t)$$

∇ : partial differentiation with respect to all different parameters individually

t : the t-th iteration

ε : adjustment step size, should be carefully chosen

$$a_{t+1} = a_t - \varepsilon_t \frac{\partial L(\Lambda)}{\partial a}, \; a : \text{an arbitrary parameter of } \Lambda$$

  - every training observation may change the parameters of ALL models, not the model for its class only

- **Using MCE in feature optimization**

$$\hat{f} = \arg\min_{f} E_X\left[L\left(f(X), \Lambda^{(f)}\right)\right]$$

f : a transformation function (with a set of parameters) to obtain better features from the original feature X

  - when the features changed, the models also changed accordingly

Maximum Mutual Information Estimation

- **Mutual Information**

m_k: k-th symbol sent at transmitter

$\hat{m}_k$: decided k-th symbol at receiver

$m_k, \hat{m}_k \in \{x_1, x_2, x_3, \ldots x_M\}$, M possibilities

  – knowledge at D about the event $m_k = x_i$

    before reception/decision : $p(x_i)$

    after reception/decision : $p(x_i|x_j)$, when $\hat{m}_k = x_j$

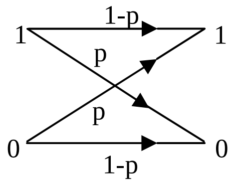
  – Quantity of Information Changed by the reception/decision of $\hat{m}_k = x_j$ about the event $m_k = x_i$

$$I(x_i; x_j) = \log\left[\frac{p(x_i|x_j)}{p(x_i)}\right] = \log\left[\frac{p(x_i, x_j)}{p(x_i)p(x_j)}\right] = I(x_j; x_i) = \text{mutual information}$$

- **Example :Binary Symmetric Channel**

$p(1)= \frac{1}{2}$,

$p(0)= \frac{1}{2}$,



$p(1|1) = p(0|0) = 1 - p$, $\quad \mathrm{I}(1;1) = \mathrm{I}(0;0) = \log_2[2(1-p)]$

$p(1|0) = p(0|1) = p$, $\quad \mathrm{I}(1;0) = \mathrm{I}(0;1) = \log_2[2p]$

$0 < p < \frac{1}{2}$

  – $\mathrm{I}(1;1)=\log_2[2(1\text{-}p)]$,

$p = 0$	$\rightarrow$ exact transmission,	I (1; 1) = 1 bit	(of information)
$0 < p < \frac{1}{2}$	$\rightarrow$ noisy transmission,	I (1; 1) < 1 bit	(of information)
$p = \frac{1}{2}$	$\rightarrow$ completely confusing channel,	I (1; 1) = 0 bit	(of information)

$\mathrm{I}(1;0)=\log_2[2p]$,

$p = \frac{1}{2}$	$\rightarrow$ completely confusing channel,	I (1; 0) = 0 bit	(of information)
$0 < p < \frac{1}{2}$	$\rightarrow$ noisy transmission,	I (1; 0) < 0 bit	(of information)
$p = 0$	$\rightarrow$ exact transmission,	I (1; 0) = -∞	(impossible)

Maximum Mutual Information Estimation

- **Classification Problem**

  $\{C_i, i=1,2,...M\}$, M classes ; X: some observation

  $$\boxed{S} \xrightarrow[X]{} \boxed{\text{Classifier}} \xrightarrow[C_j \text{ decided}]{} \boxed{D}$$

  $$I(X;C_j) = \log\left[\frac{p(X,C_j)}{p(X)p(C_j)}\right] = \max$$

- **MAP Principle**

  $$p(C_j|X) = \frac{p(X|C_j)p(C_j)}{p(X)} \Rightarrow p(X|C_j)p(C_j) = \max$$

  – if $p(C_j) = \frac{1}{M}, \ all \ j$

  $$p(C_j|X) = \frac{p(X|C_j)}{p(X)} = \frac{p(X,C_j)}{p(X)p(C_j)} = e^{I(X;C_j)} = \max, \ \text{max mutual information}$$

- **When All Classes are Equally Probable, MAP Principle Gives Maximum Mutual Information**
  – maximum mutual information considers differently from MAP if $p(C_j)$ are not equal

Maximum Mutual Information Estimation

- **A Different View (of MAP)**

$$p(C_j|X) = \frac{p(X|C_j)\,p(C_j)}{p(X)} = \frac{p(X|C_j)\,p(C_j)}{\sum_k p(X|C_k)\,p(C_k)} \qquad , X \in C_j$$

$$= \frac{p(X|C_j)\,p(C_j)}{\underset{\substack{\uparrow\\ \text{correct model}}}{p(X|C_j)\,p(C_j)} + \sum_{k\neq j} \underset{\substack{\uparrow\\ \text{competing model}}}{p(X|C_k)\,p(C_k)}} = \frac{1}{1 + \frac{\sum_{k\neq j} p(X|C_k)\,p(C_k)}{p(X|C_j)\,p(C_j)}} = \max$$

- **Discriminative Training**

$$F(\Lambda) = \frac{p(X|\lambda^{(j)})\,p(C_j)}{\sum_{k\neq j} p(X|\lambda^{(k)})\,p(C_k)} = \max, \qquad \Lambda = \left\{\lambda^{(1)}, \lambda^{(2)}, \ldots \lambda^{(M)}\right\}$$

set of models for all classes

$$\Lambda_{t+1} = \Lambda_t - \varepsilon_t\, \nabla F(\Lambda_t) : \text{gradient descent algorithm}$$

- discriminating capabilities among competing models considered
- optimization with respect to the probability scores instead of error rates
- $P(C_k)$ included in optimization
- number of completing models considered may be empirically chosen practically