Return to Main

Introduction:

Mutual Information:

Minimum Error Classification:

On-Line Resources:

Ganapath: Discriminative Techniques
Woodland: MLLR
Juang: MCE

LECTURE 41: DISCRIMINATIVE TRAINING

- Objectives:
  - Mutual Information
  - Maximum Mutual Information Estimation
  - Minimum Error Rate Estimation

This lecture follows the course textbook:

> X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development*, Prentice Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-022616-5, 2001.

Another good source is:

> A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Mississippi State University, January 2002.

LECTURE 41: DISCRIMINATIVE TRAINING

- Objectives:
  - Mutual Information
  - Maximum Mutual Information Estimation
  - Minimum Error Rate Estimation

This lecture follows the course textbook:

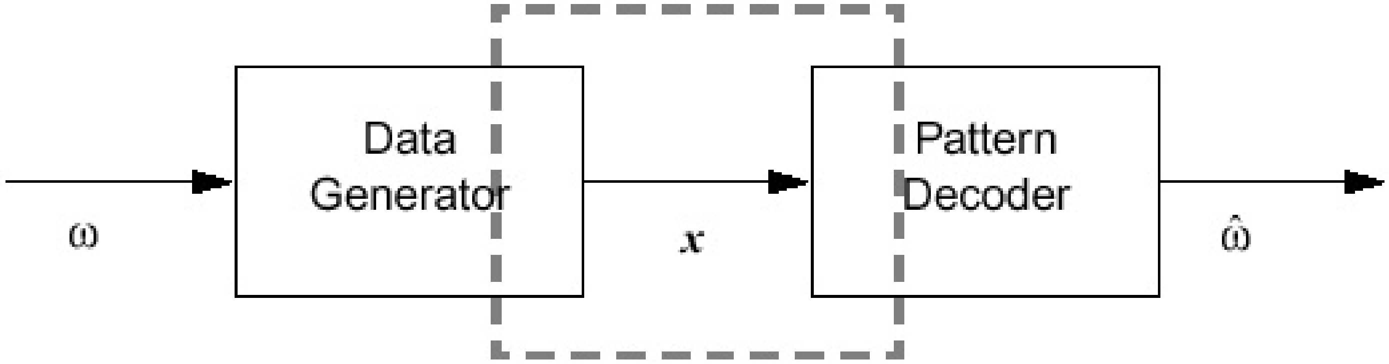
> X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development*, Prentice Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-022616-5, 2001.

Another good source is:

A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Mississippi State University, January 2002.

THE PATTERN RECOGNITION PROBLEM

Recall our communication theory model for speech recognition (simplified):



The rule for minimum error rate classification is to select the class ω_i with the maximum posterior probability, $P(\omega_i|x)$. Recalling Bayes' rule:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$$

The probability of the observation, $p(x)$, can be expressed as:

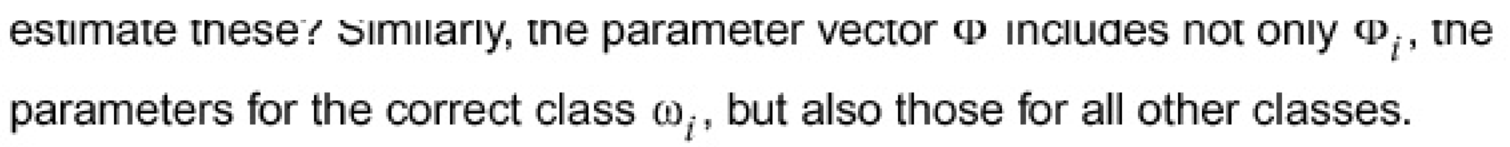
$$p(x) = \sum_k p(x|\omega_k)p(\omega_k)$$

Recall that in the classification stage, $p(x)$ can be considered a constant.

A conditional maximum likelihood estimator (CMLE), denoted , is defined as follows:

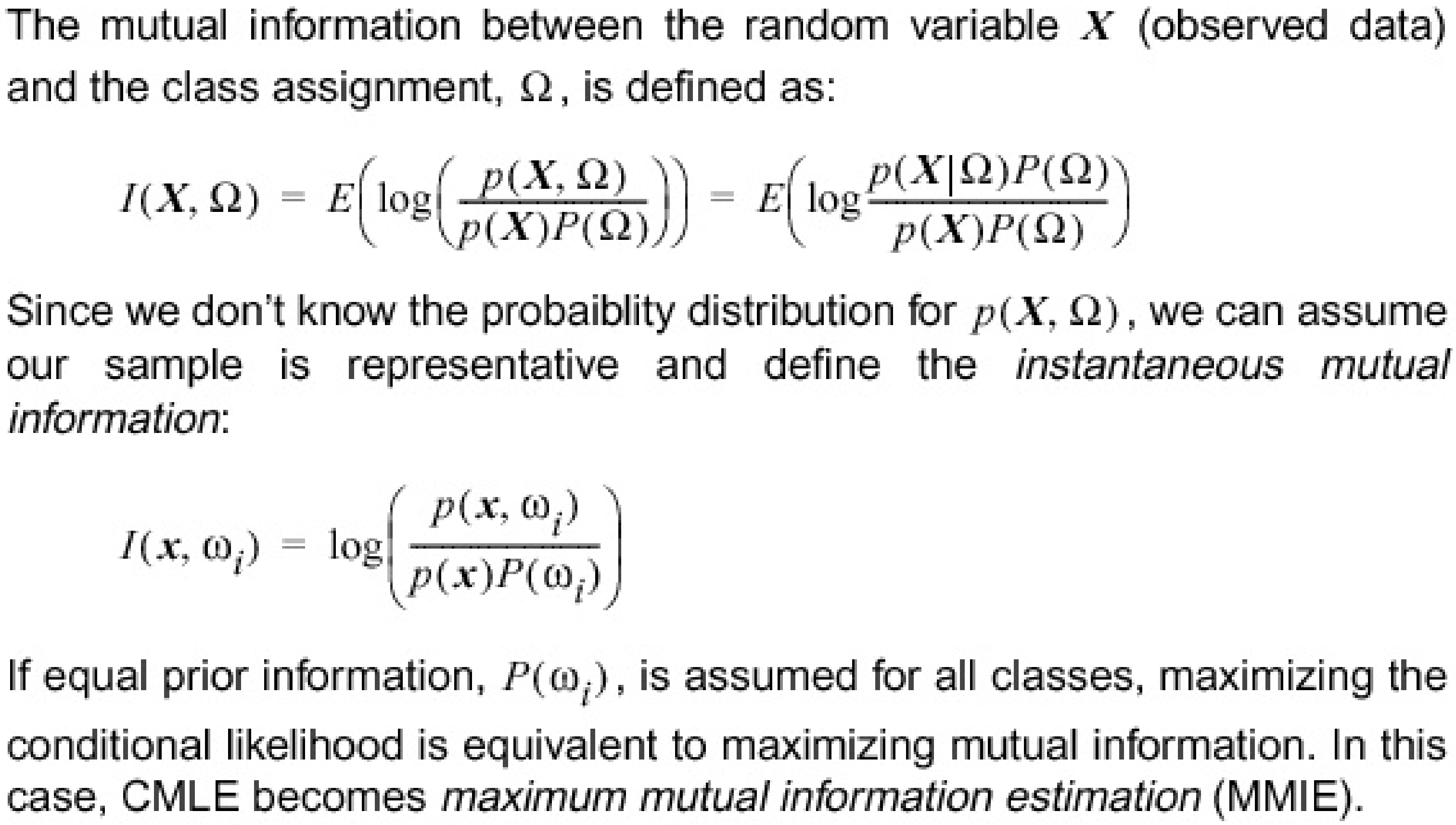
$$\theta_{CMLE}(x) = \Phi_{MAP} = \overset{\text{argmax}}{\Phi} p_{\Phi}(\omega_i|x)$$

Note that the summation in our equation for $p(x)$ extends over all possible classes (correct and incorrect!) and sums partial probabilities. How will we



estimate these? Similarly, the parameter vector Φ includes not only Φ_i, the parameters for the correct class ω_i, but also those for all other classes.

CONDITIONAL LIKELIHOOD AND MUTUAL INFORMATION



The mutual information between the random variable X (observed data) and the class assignment, Ω, is defined as:

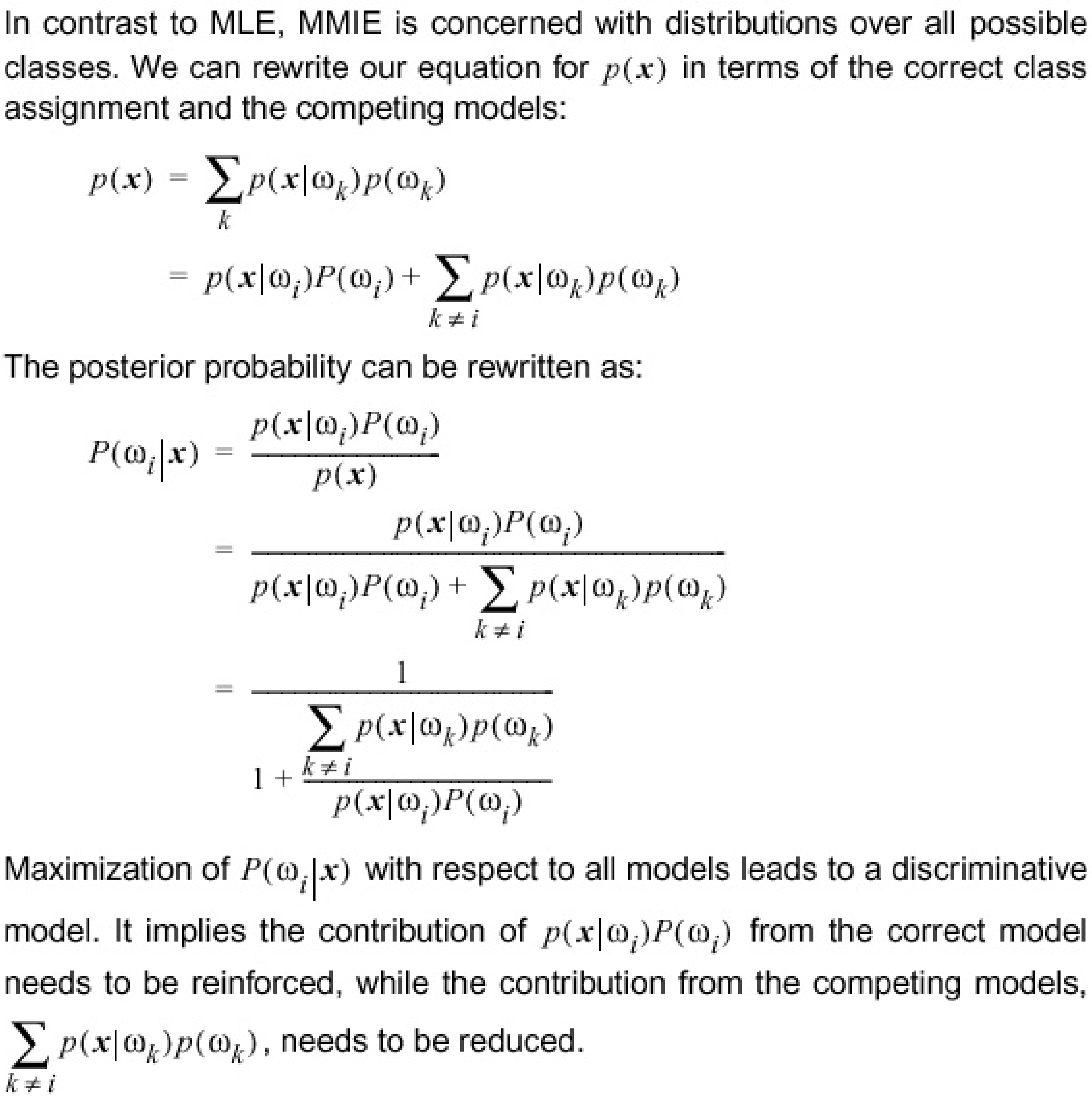
$$I(X, \Omega) = E\left(\log\left(\frac{p(X, \Omega)}{p(X)P(\Omega)}\right)\right) = E\left(\log\frac{p(X|\Omega)P(\Omega)}{p(X)P(\Omega)}\right)$$

Since we don't know the probaiblity distribution for $p(X, \Omega)$, we can assume our sample is representative and define the *instantaneous mutual information*:

$$I(x, \omega_i) = \log\left(\frac{p(x, \omega_i)}{p(x)P(\omega_i)}\right)$$

If equal prior information, $P(\omega_i)$, is assumed for all classes, maximizing the conditional likelihood is equivalent to maximizing mutual information. In this case, CMLE becomes *maximum mutual information estimation* (MMIE).

A DISCRIMINANT MODEL



In contrast to MLE, MMIE is concerned with distributions over all possible classes. We can rewrite our equation for $p(\boldsymbol{x})$ in terms of the correct class assignment and the competing models:

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{k} p(\boldsymbol{x}|\omega_k)p(\omega_k) \\ &= p(\boldsymbol{x}|\omega_i)P(\omega_i) + \sum_{k \neq i} p(\boldsymbol{x}|\omega_k)p(\omega_k) \end{aligned}$$

The posterior probability can be rewritten as:

$$\begin{aligned} P(\omega_i|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x}|\omega_i)P(\omega_i) + \sum_{k \neq i} p(\boldsymbol{x}|\omega_k)p(\omega_k)} \\ &= \frac{1}{1 + \frac{\sum_{k \neq i} p(\boldsymbol{x}|\omega_k)p(\omega_k)}{p(\boldsymbol{x}|\omega_i)P(\omega_i)}} \end{aligned}$$

Maximization of $P(\omega_i|\boldsymbol{x})$ with respect to all models leads to a discriminative model. It implies the contribution of $p(\boldsymbol{x}|\omega_i)P(\omega_i)$ from the correct model needs to be reinforced, while the contribution from the competing models, $\sum_{k \neq i} p(\boldsymbol{x}|\omega_k)p(\omega_k)$, needs to be reduced.

MMIE AND MLE ARE SIMILAR AND YET DIFFERENT

- In MLE, only the correct model is updated during training. In MMIE, all models are updated during training, even with one training sample.

- The greater the prior information on the class assignment, the more effect it has on the MMIE estimator.

- If the assumption of the underlying distribution is correct, MMIE and MLE should converge to the same result. However, in practice, MMIE must produce a lower likelihood for the true class assignment (underlying distribution).

- MMIE and MLE are consistent estimators, but

MMIE has greater variance. MMIE tries not only to increase the likelihood of the correct class, but decrease the likelihood of the incorrect class.

- MMIE is computationally expensive. Why?

- How do we estimate the probability of the class assignment for the incorrect classes?

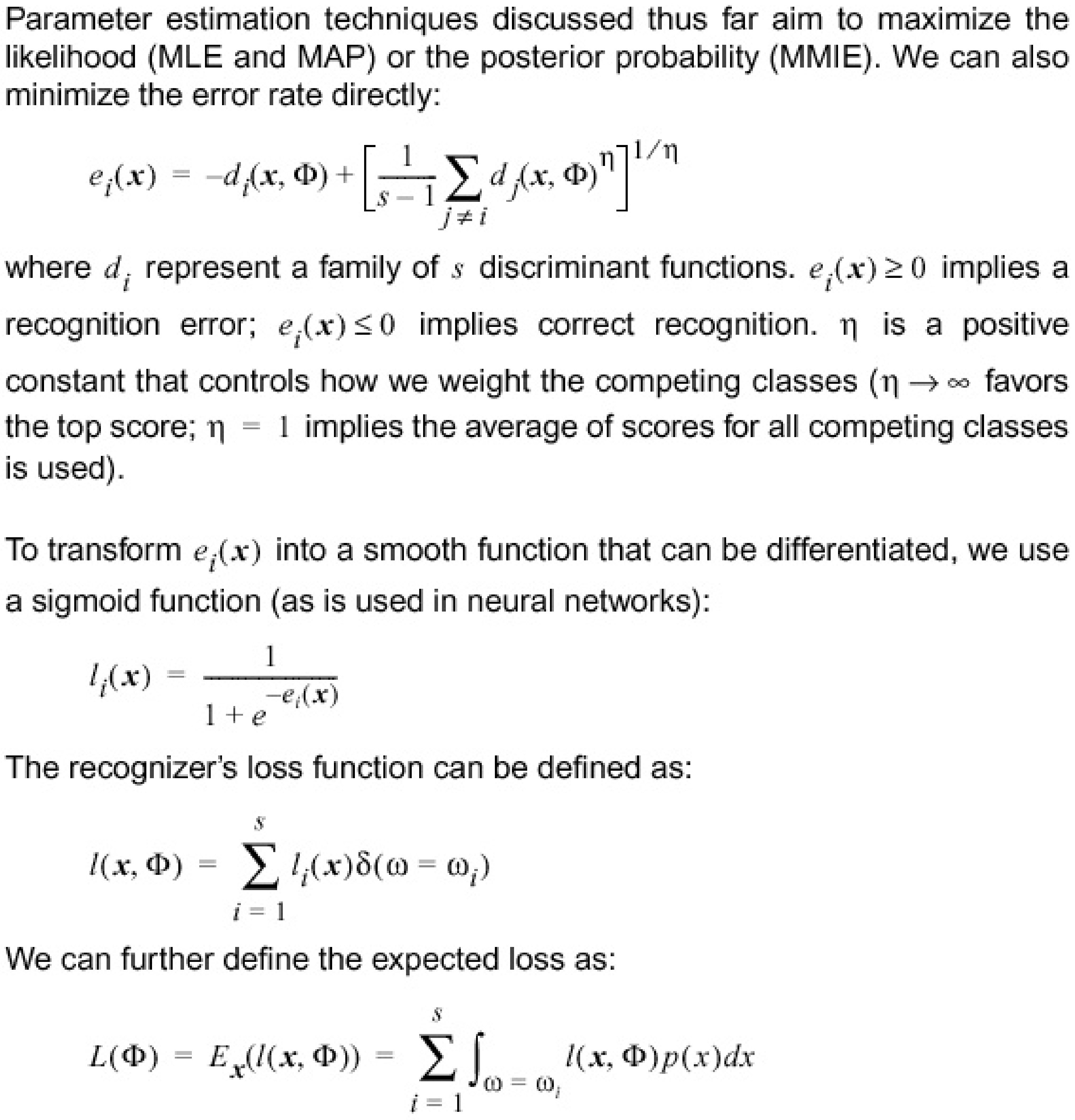
- Experimental results: CU/HTK word error rates on eval97sub and eval98 using h5train00sub training:

MMIE	%WER	
Iteration	eval97sub	eval98
0 (MLE)	46.0	46.5

1	43.8	45.0
2	43.7	44.6
3	44.1	44.7

The results in Table 3 show that again the peak improvement comes after two iterations, but there is an even larger reduction in WER: 2.3% absolute on eval97sub and 1.9% absolute on eval98. The word error rate for the 1-best hypothesis from the original bigram word lattices measured on 10% of the training data was 27.4%. The MMIE models obtained after two iterations on the same portion of training data gave an error rate of 21.2%, so again MMIE provided a very sizeable reduction in training set error.

MISCLASSIFICATION ERROR RATE AND LOSS FUNCTIONS



Parameter estimation techniques discussed thus far aim to maximize the likelihood (MLE and MAP) or the posterior probability (MMIE). We can also minimize the error rate directly:

$$e_i(\boldsymbol{x}) = -d_i(\boldsymbol{x}, \Phi) + \left[\frac{1}{s-1}\sum_{j \neq i} d_j(\boldsymbol{x}, \Phi)^{\eta}\right]^{1/\eta}$$

where d_i represent a family of s discriminant functions. $e_i(\boldsymbol{x}) \geq 0$ implies a recognition error; $e_i(\boldsymbol{x}) \leq 0$ implies correct recognition. η is a positive constant that controls how we weight the competing classes ($\eta \to \infty$ favors the top score; $\eta = 1$ implies the average of scores for all competing classes is used).

To transform $e_i(x)$ into a smooth function that can be differentiated, we use a sigmoid function (as is used in neural networks):

$$l_i(\boldsymbol{x}) = \frac{1}{1 + e^{-e_i(\boldsymbol{x})}}$$

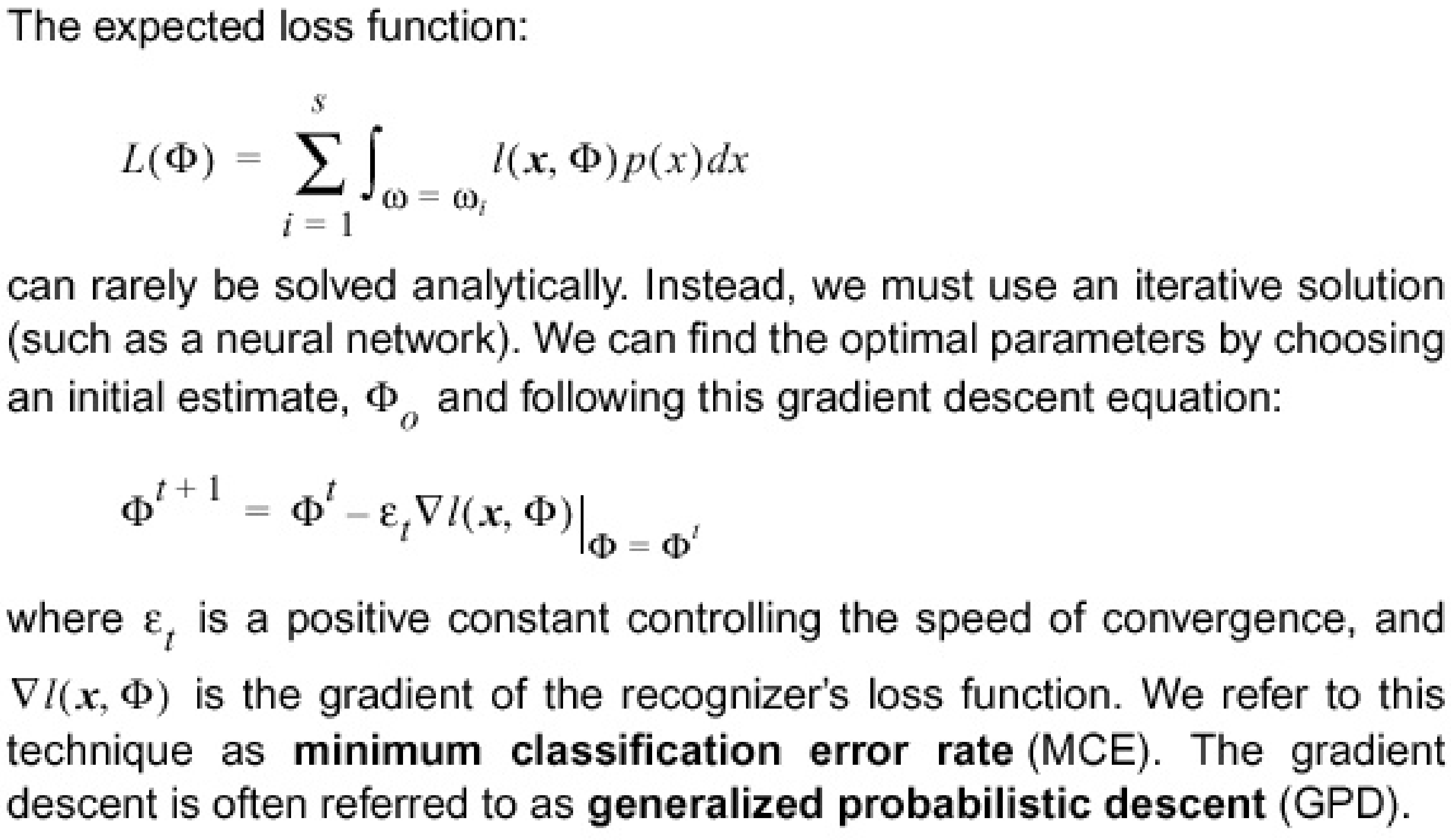
The recognizer's loss function can be defined as:

$$l(\boldsymbol{x}, \Phi) = \sum_{i=1}^{s} l_i(\boldsymbol{x})\delta(\omega = \omega_i)$$

We can further define the expected loss as:

$$L(\Phi) = E_{\boldsymbol{x}}(l(\boldsymbol{x}, \Phi)) = \sum_{i=1}^{s} \int_{\omega = \omega_i} l(\boldsymbol{x}, \Phi)p(x)dx$$

GRADIENT DESCENT SOLUTIONS



The expected loss function:

$$L(\Phi) = \sum_{i=1}^{S} \int_{\omega = \omega_i} l(\boldsymbol{x}, \Phi) p(x) dx$$

can rarely be solved analytically. Instead, we must use an iterative solution (such as a neural network). We can find the optimal parameters by choosing an initial estimate, Φ_o and following this gradient descent equation:

$$\Phi^{t+1} = \Phi^t - \varepsilon_t \nabla l(\boldsymbol{x}, \Phi)\Big|_{\Phi = \Phi^t}$$

where ε_t is a positive constant controlling the speed of convergence, and $\nabla l(\boldsymbol{x}, \Phi)$ is the gradient of the recognizer's loss function. We refer to this technique as **minimum classification error rate** (MCE). The gradient descent is often referred to as **generalized probabilistic descent** (GPD).

COMPARISON OF PERFORMANCE

- MMIE and MCE are very expensive and often application specific. A similar, more pragmatic approach, is **corrective training**.
- In corrective training, we keep a "near-miss" list and reinforce correct choices, and penalize near misses. This is an ad-hoc procedure that works well in practice.
- MCE and MMIE produce very similar results.