CS 2750 Machine Learning
Lecture 11

Multi-way classification

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Multi-way classification

- **Binary classification** $Y = \{0,1\}$
- **Multi-way classification**
  - **K classes** $Y = \{0,1,\ldots,K-1\}$
  - **Goal:** learn to classify correctly K classes
  - Or **learn** $f : X \to \{0,1,\ldots,K-1\}$
- **Errors:**
  - **Zero-one (misclassification) error for an example:**

$$Error_1(\mathbf{x}_i, y_i) = \begin{cases} 1 & f(\mathbf{x}_i, \mathbf{w}) \neq y_i \\ 0 & f(\mathbf{x}_i, \mathbf{w}) = y_i \end{cases}$$

  - **Mean misclassification error (for a dataset):**

$$\frac{1}{n}\sum_{i=1}^{n} Error_1(\mathbf{x}_i, y_i)$$

Multi-way classification

Approaches:

- **Generative model approach**
  - Generative model of the distribution p(**x**,y)
  - Learns the parameters of the model through density estimation techniques
  - Discriminant functions are based on the model
    - "Indirect" learning of a classifier
- **Discriminative approach**
  - Parametric discriminant functions
  - Learns discriminant functions **directly**
    - A logistic regression model.

Generative model approach

Indirect:

1. **Represent and learn the distribution** $p(\mathbf{x}, y)$
2. **Define and use probabilistic discriminant functions**

$$g_i(\mathbf{x}) = \log\ p(y = i \mid \mathbf{x})$$

Model $p(\mathbf{x}, y) = p(\mathbf{x} \mid y)p(y)$

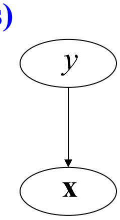
- $p(\mathbf{x} \mid y)$ = **Class-conditional distributions (densities)**

  k class conditional distributions

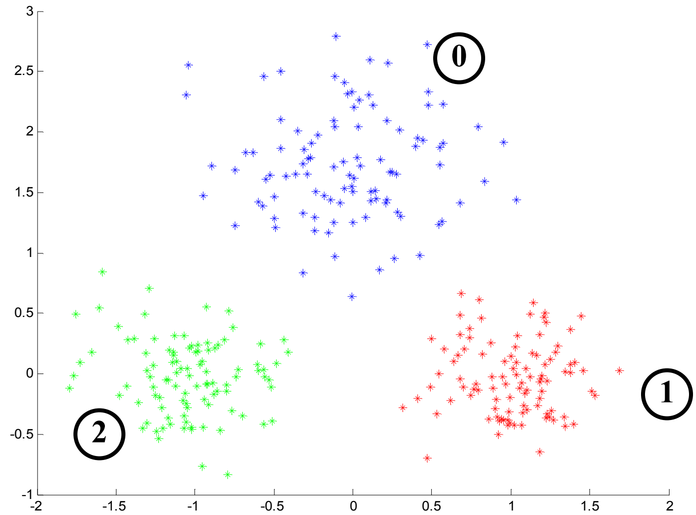
$$p(\mathbf{x} \mid y = i) \qquad \forall i \quad 0 \le i \le K-1$$

- $p(y)$ = **Priors on classes**
- **-** probability of class *y*

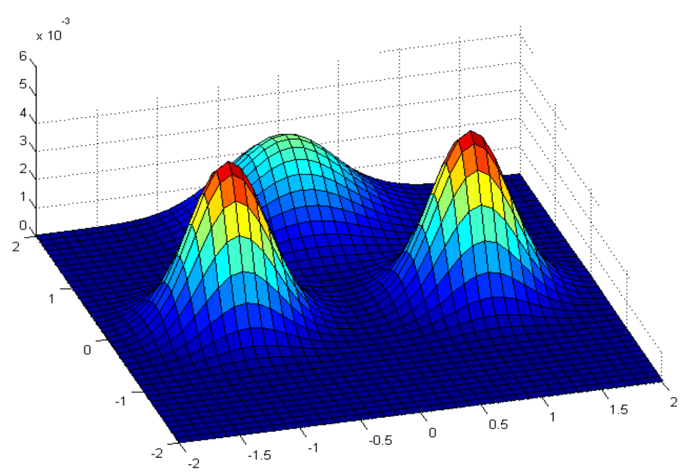
$$\sum_{i=1}^{K-1} p(y = i) = 1$$



Multi-way classification. Example



Multi-way classification



Making class decision

Discriminant functions can be based on:

- **Likelihood of data** – choose the class (Gaussian) that explains the input data ($\mathbf{x}$) better (likelihood of the data)

Choice: $$i = \arg\max_{i=0,\ldots k-1} \; p(\mathbf{x} \mid \boldsymbol{\theta}_i)$$

$$p(\mathbf{x} \mid \boldsymbol{\theta}_i) \approx p(\mathbf{x} \mid \mu_i, \boldsymbol{\Sigma}_i) \qquad \text{For Gaussians}$$

- **Posterior of a class** – choose the class with higher posterior probability

Choice: $$i = \arg\max_{i=0,\ldots k-1} \; p(y = i \mid \mathbf{x}, \boldsymbol{\theta}_i)$$

$$p(y = i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \Theta_i)\, p(y = i)}{\sum_{j=0}^{k-1} p(\mathbf{x} \mid \Theta_j)\, p(y = j)}$$

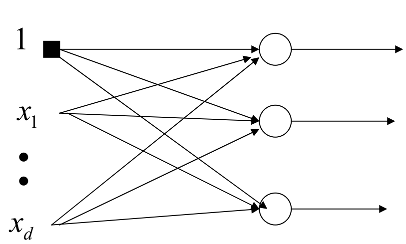
Discriminative approach

- **Parameteric model** of discriminant functions
- Learns the discriminant functions directly

How to learn to classify multiple classes, say 0,1,2?

Approach 1:

– A binary logistic regression on every class versus the rest

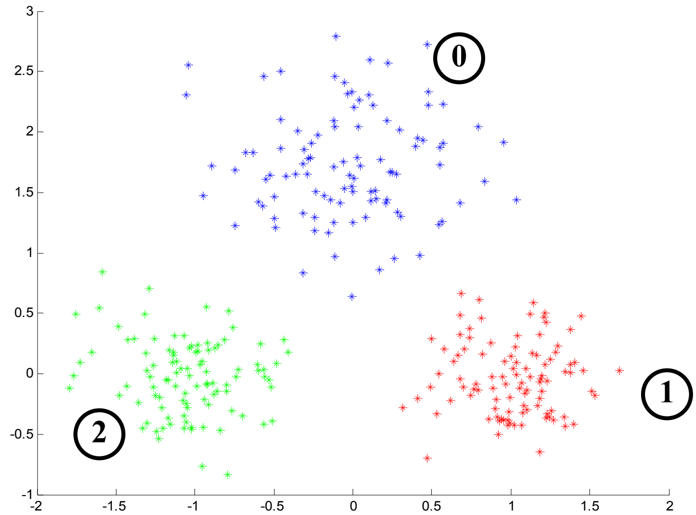


0 vs. (1 or 2)

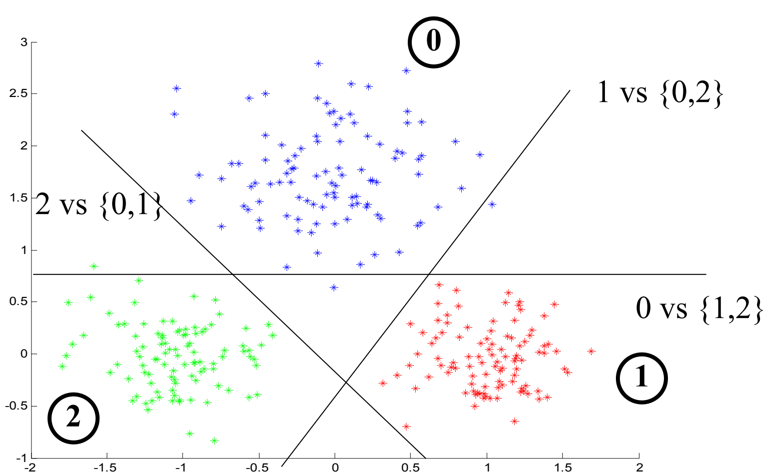
1 vs. (0 or 2)

2 vs. (0 or 1)

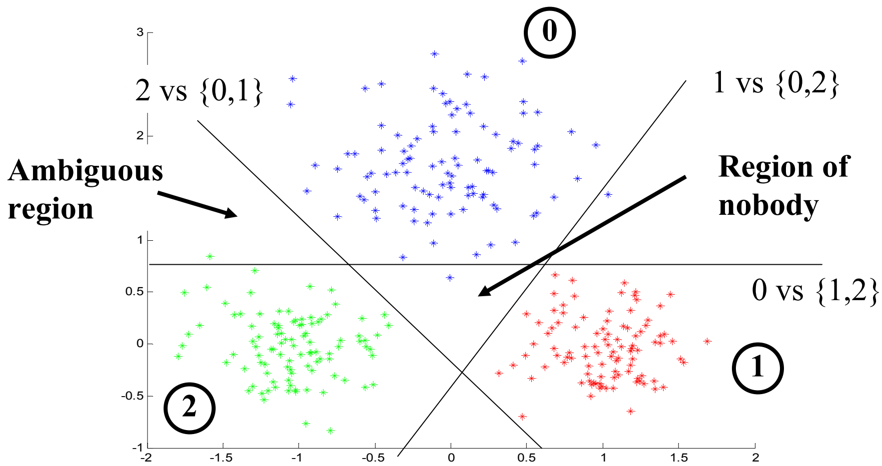
Multi-way classification. Example



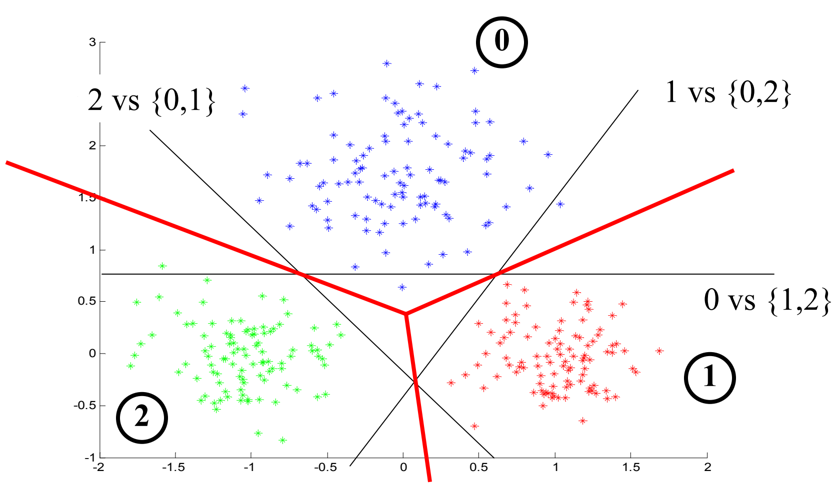
Multi-way classification. Approach 1.



Multi-way classification. Approach 1.



Multi-way classification. Approach 1.

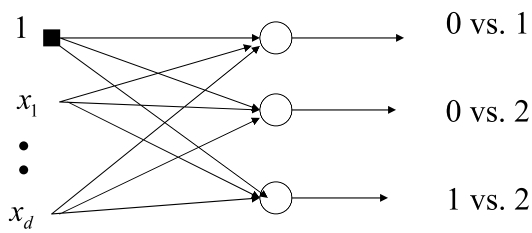


Discriminative approach.

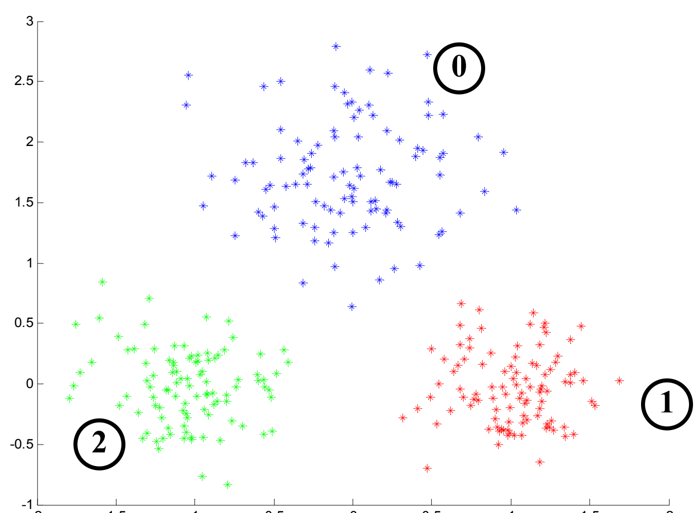
How to learn to classify multiple classes, say 0,1,2 ?

Approach 2:

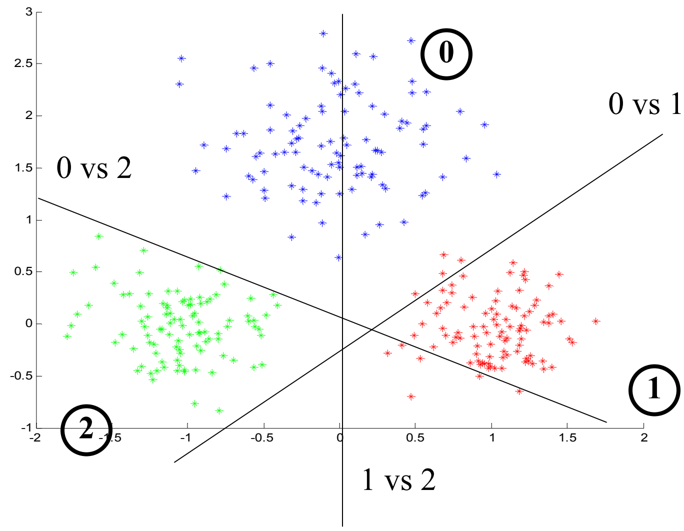
– A binary logistic regression on all pairs



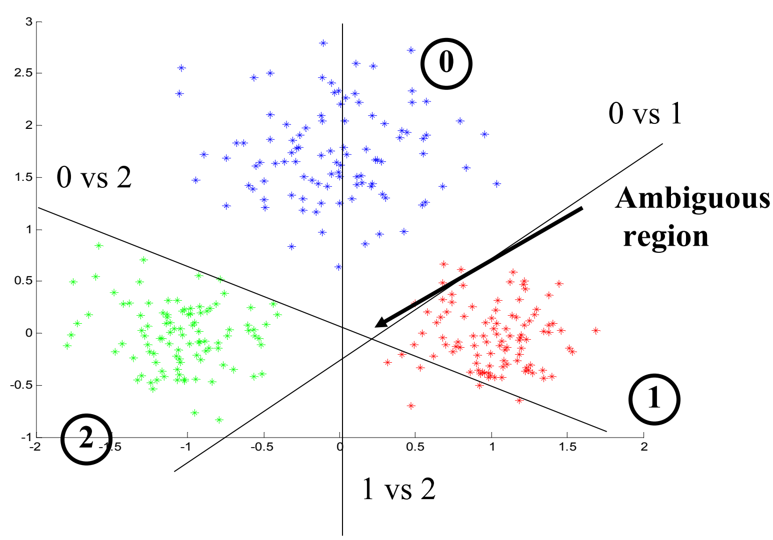
Multi-way classification. Example



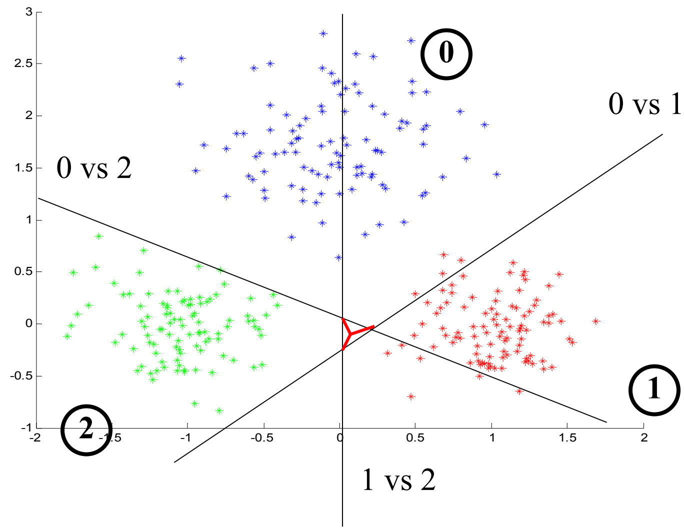
Multi-way classification. Approach 2



Multi-way classification. Approach 2

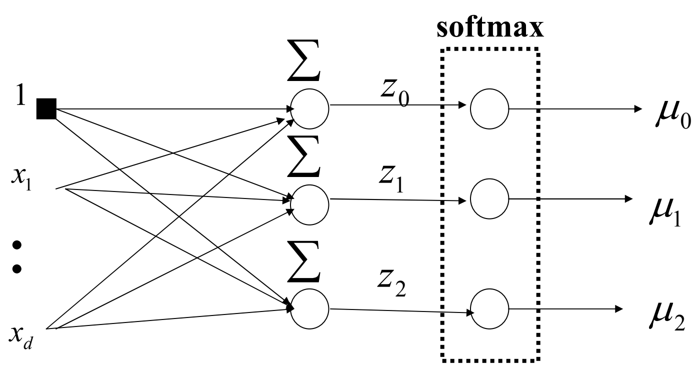


Multi-way classification. Approach 2



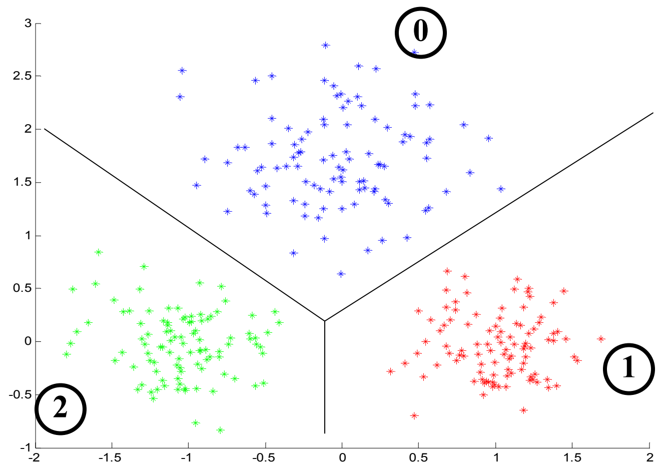
Multi-way classification with softmax

- A solution to the problem of having an ambiguous region



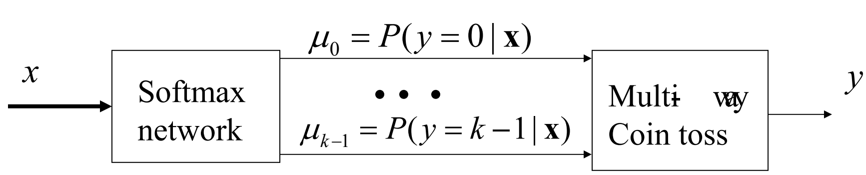
$$p(y=i \mid \mathbf{x}) = \mu_i = \frac{\exp(\mathbf{w}_i^T \mathbf{x})}{\sum_j \exp(\mathbf{w}_j^T \mathbf{x})} \qquad \sum_i \mu_i = 1$$

Multi-way classification with softmax



Learning of the softmax model

- Learning of parameters **w**: statistical view



Assume outputs y are transformed as follows

$$y \in \{0 \quad 1 \quad .. \quad k-1\} \quad \Rightarrow \quad y \in \left\{ \begin{pmatrix} 1 \\ 0 \\ .. \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ .. \\ 0 \end{pmatrix} \quad .. \quad \begin{pmatrix} 0 \\ 0 \\ .. \\ 1 \end{pmatrix} \right\}$$

Learning of the softmax model

- Learning of the parameters **w**: statistical view
- **Likelihood of outputs**

$$L(D,\mathbf{w}) = p(\mathbf{Y}\,|\,\mathbf{X},w) = \prod_{i=1,..n} p(y_i\,|\,\mathbf{x}_i,\mathbf{w})$$

- We want parameters **w** that maximize the likelihood
- **Log-likelihood trick**
  - Optimize log likelihood of outputs instead:

$$l(D,\mathbf{w}) = \log \prod_{i=1,..n} p(y_i\,|\,\mathbf{x},\mathbf{w}) = \sum_{i=1,..n} \log p(y_i\,|\,\mathbf{x},\mathbf{w})$$

$$= \sum_{i=1,..n} \sum_{q=0}^{k-1} \log \mu_i^{y_{i,q}} = \sum_{i=1,..n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$$

- **Objective to optimize** $J(D_i,\mathbf{w}) = -\sum_{i=1}^{n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$

Learning of the softmax model

- **Error to optimize:**

$$J(D_i,\mathbf{w}) = -\sum_{i=1}^{n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$$

- **Gradient**

$$\frac{\partial}{\partial w_{jk}} J(D_i,\mathbf{w}) = \sum_{i=1}^{n} -x_{i,j}(y_{i,j} - \mu_{i,j})$$

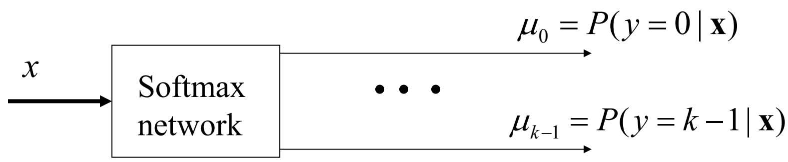
- The same very easy **gradient update** as used for the binary logistic regression

$$\mathbf{w}_j \leftarrow \mathbf{w}_j + \alpha \sum_{i=1}^{n} (y_{i,j} - \mu_{i,j})\mathbf{x}_i$$

- But now we have to update the weights of k networks

Multi-way classification

- When is the **softmax** the right model ?



- Assume:

$$p(\mathbf{x} \mid y=i) = h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - A(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} \right\}$$

$\boldsymbol{\theta}_i$ - location parameter for class conditional i

$\boldsymbol{\varphi}$ - scaling parameter (the same for all classes)

Multi-way classification

- **Class conditional:**

$$p(\mathbf{x} \mid y=i) = h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - A(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} \right\}$$

- **Class posterior:**

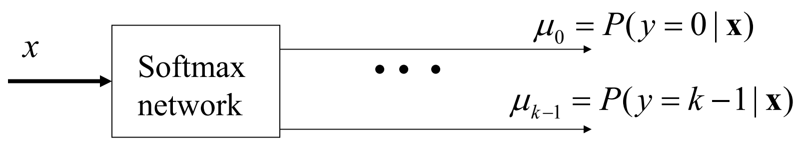
$$p(y=i \mid x) = \frac{p(\mathbf{x} \mid y=i)\, p(y=i)}{\sum_j p(\mathbf{x} \mid y=j)\, p(y=j)}$$

$$= \frac{h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - A(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} \right\} p(y=i)}{\sum_j h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_j^T \mathbf{x} - A(\boldsymbol{\theta}_j))}{a(\boldsymbol{\varphi})} \right\} p(y=j)} = \frac{\exp(\mathbf{w}_i^T \mathbf{x} + b_i)}{\sum_j \exp(\mathbf{w}_j^T \mathbf{x} + b_j)}$$

$$\mathbf{w}_i = \frac{\boldsymbol{\theta}_i}{a(\boldsymbol{\varphi})} \qquad b_i = \frac{A(\boldsymbol{\theta}_i)}{a(\boldsymbol{\varphi})} + \ln p(y=i)$$

Multi-way classification

- **Softmax model is an accurate model** when class conditional densities are represented with densities from the exponential family with the same scaling parameter
- For **two classes** it reduces to the **logistic regression model**



$$p(\mathbf{x} \mid y = i) = \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - b(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} + c(\mathbf{x}, \boldsymbol{\varphi}) \right\}$$

$\boldsymbol{\theta}_i$ - location parameter for class conditional i

$\boldsymbol{\varphi}$ - scaling parameter (the same for all classes)

Bayesian decision theory

Confusion matrix

Results of classification are recorded in:

- **Confusion matrix:**
  - Counts of examples with:
  - class label ω_j that are classified with a label α_i

	$\omega = 0$	$\omega = 1$	$\omega = 2$
$\alpha = 0$	140	20	22
$\alpha = 1$	17	54	8
$\alpha = 2$	12	4	76

agreement

Zero-one loss function

- **Misclassification error**
  - Based on the zero one loss function
    - Any misclassified example counts as 1
    - Correctly classified example counts as 0

	$\omega = 0$	$\omega = 1$	$\omega = 2$
$\alpha = 0$	140	20	22
$\alpha = 1$	17	54	8
$\alpha = 2$	12	4	76

agreement

- What is the zero one loss for the confusion matrix?

General loss function

- **Error function based on a more general loss function**
  - Different misclassifications have different weight (loss)
  - α_i our choice
  - ω_j true label
  - $\lambda(\alpha_i \mid \omega_j)$ loss for classification

Example:

$\lambda(\alpha_i \mid \omega_j)$

	$\omega = 0$	$\omega = 1$	$\omega = 2$
$\alpha = 0$	0	1	5
$\alpha = 1$	3	0	2
$\alpha = 2$	3	1	0

Bayesian decision theory

- **More general loss function**
  - Different misclassifications have different weight (loss)

  $\lambda(\alpha_i \mid \omega_j)$
- **Expected loss for the classification choice** α_i

$$R(\alpha_i \mid \mathbf{x}) = \sum_j \lambda(\alpha_i \mid \omega_j) \underbrace{P(y = \omega_j \mid \mathbf{x})}_{\text{Posterior of the class}}$$

  - Also called conditional risk
- **Decision rule:** $\alpha(\mathbf{x})$
  - Chooses label (action) according to the input
- **The optimal decision rule**

$$\alpha^*(\mathbf{x}) = \arg\min_{\alpha_i} \sum_j \lambda(\alpha_i \mid \omega_j) P(y = \omega_j \mid \mathbf{x})$$

Bayesian decision theory

- **The optimal decision rule**

$$\alpha^*(\mathbf{x}) = \arg\min_{\alpha_i} \sum_j \lambda(\alpha_i \mid \omega_j) P(y = \omega_j \mid \mathbf{x})$$

How to modify classifiers to handle different loss?

- **Discriminative models:**
  - Directly optimize the parameters according to the new loss function
- **Generative models:**
  - Learn probabilities as before
  - Decisions about classes are biased to minimize the empirical loss (as seen above)

Calculating the loss for data

- **Confusion matrix:**
  - Counts of examples with:
  - class label ω_j that are classified with a label α_i

	$\omega = 0$	$\omega = 1$	$\omega = 2$
$\alpha = 0$	140	20	22
$\alpha = 1$	17	54	8
$\alpha = 2$	12	4	76

agreement

- **Loss** $\frac{1}{N} \sum_i \sum_j \lambda(\alpha_i \mid \omega_j) N(\alpha_j \mid \omega_j)$