CS 2750 Machine Learning
Lecture 13

Multi-way classification

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

Administrative announcements

Homework 6 due on Wednesday

Plan for the upcoming month:

- **Homework 7 due on Wednesday, 03/03/04**
- **Midterm: 03/17/04**
- **Project proposals: 03/25/04**

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** $f : X \rightarrow \{0,1,\ldots,K-1\}$
- **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:

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

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

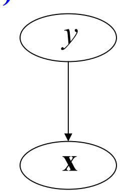
- $p(\mathbf{x} \mid y)$ = **Class-conditional distributions (densities)**

  k class-conditional distributions

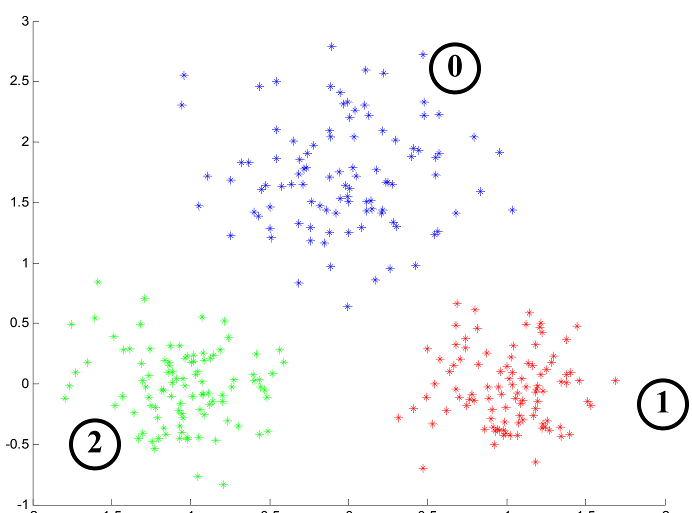
  $$p(\mathbf{x} \mid y = i) \qquad \forall i \quad 0 \le i \le K-1$$

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

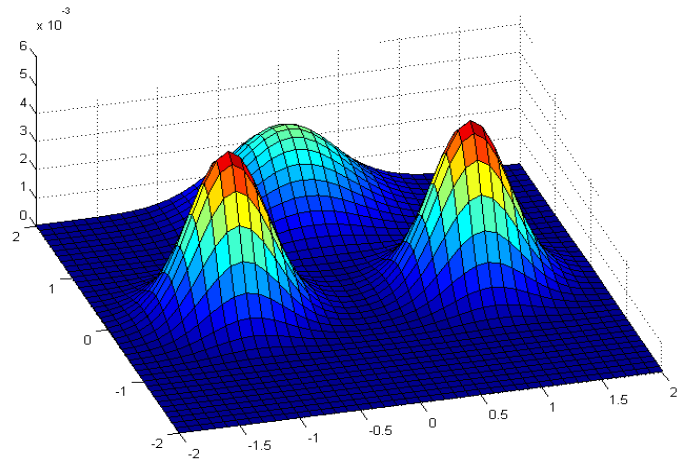
$$\sum_{i=1}^{K-1} p(y = i) = 1$$



Multi-way classification. Example



Multi-way classification



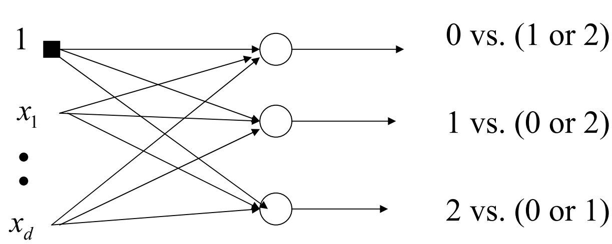
Discriminative approach.

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

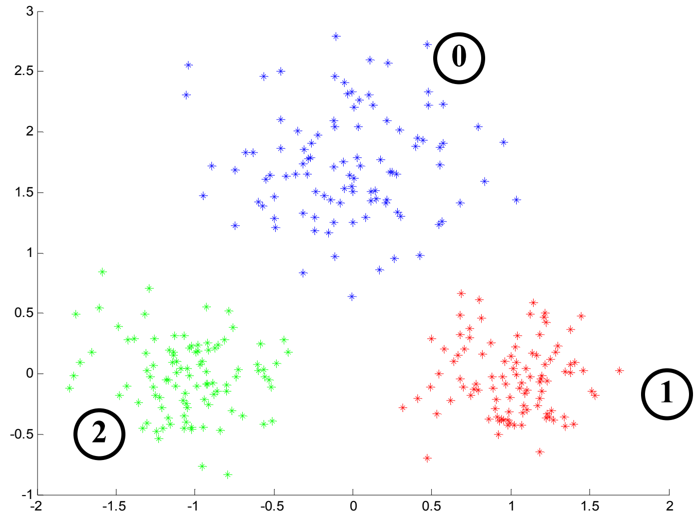
How to learn to classify multiple classes, say 0,1,2?

Approach 1:

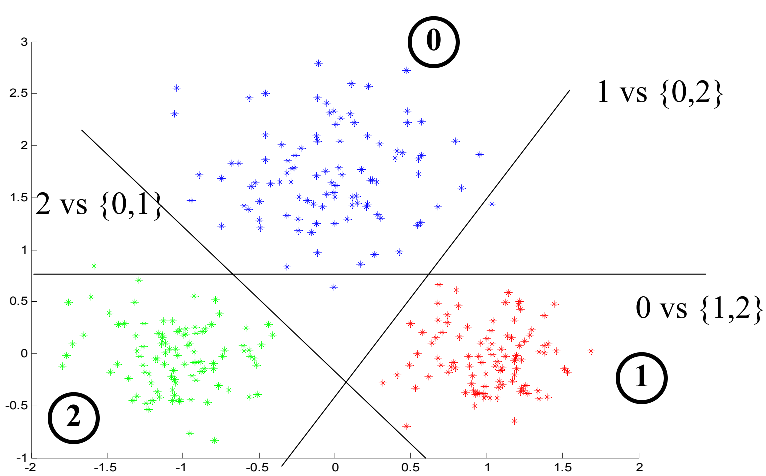
– A binary logistic regression on every class versus the rest



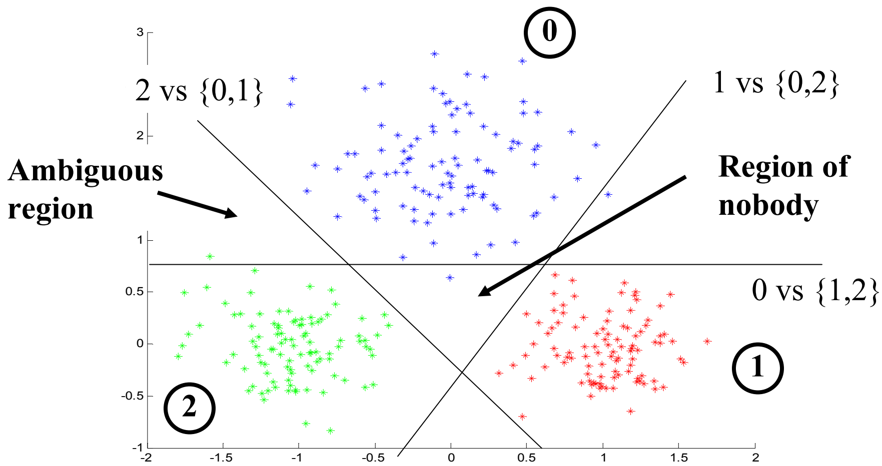
Multi-way classification. Example



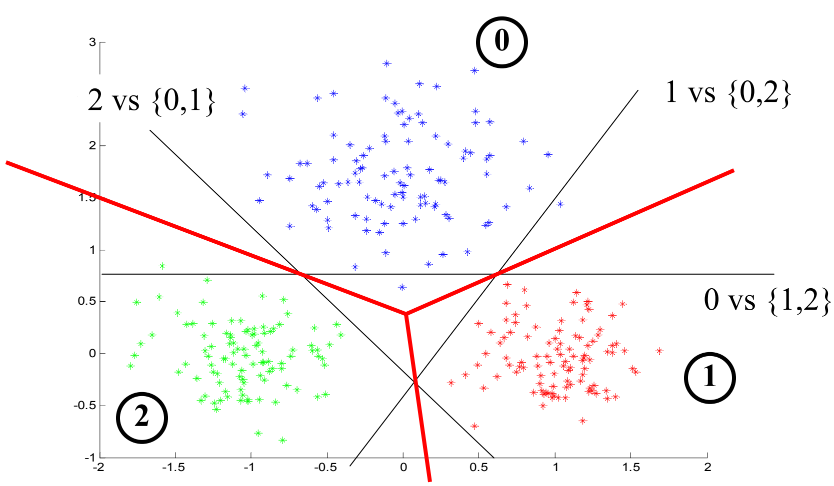
Multi-way classification. Approach 1.



Multi-way classification. Approach 1.



Multi-way classification. Approach 1.

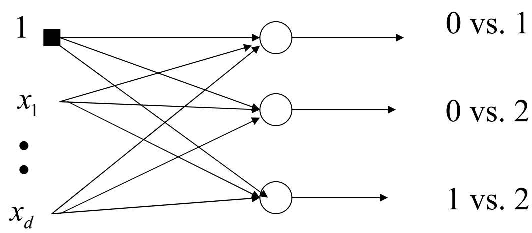


Discriminative approach.

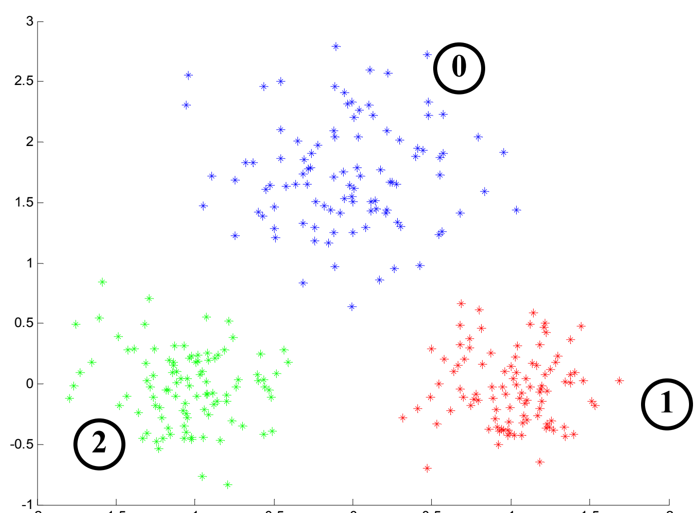
How to learn to classify multiple classes, say 0,1,2 ?

Approach 2:

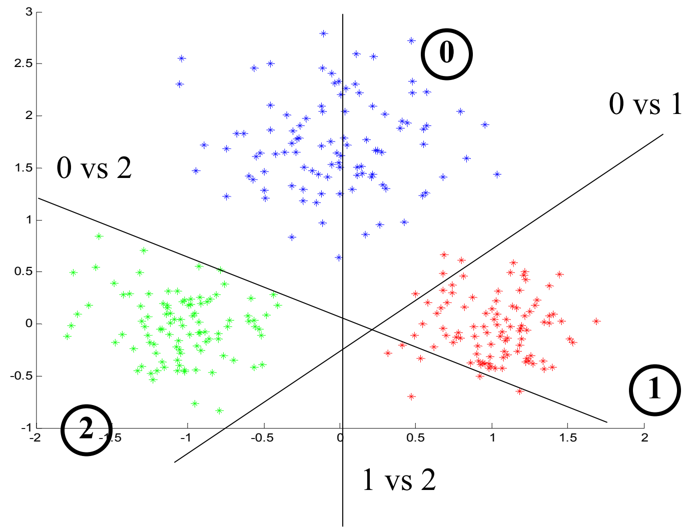
– A binary logistic regression on all pairs



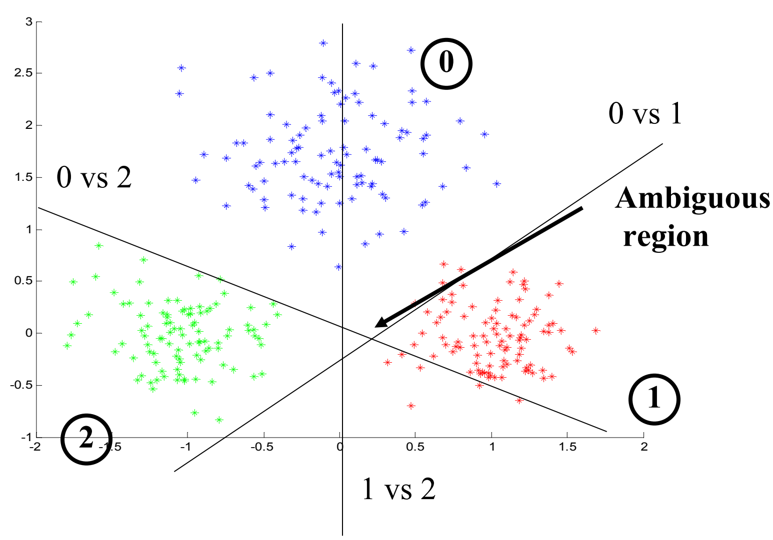
Multi-way classification. Example



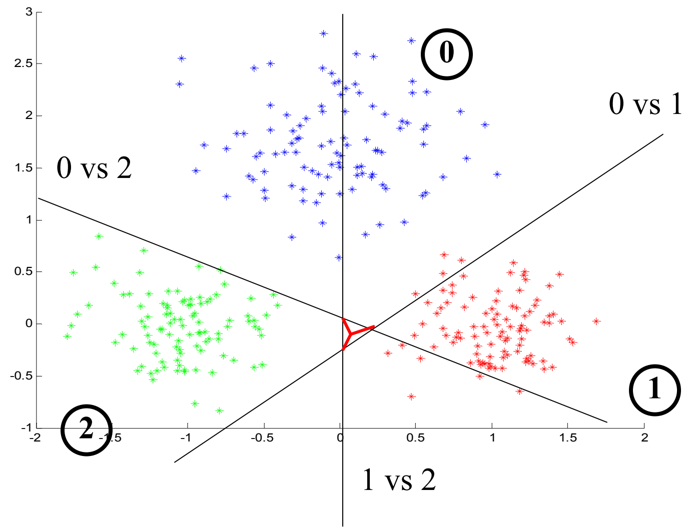
Multi-way classification. Approach 2



Multi-way classification. Approach 2

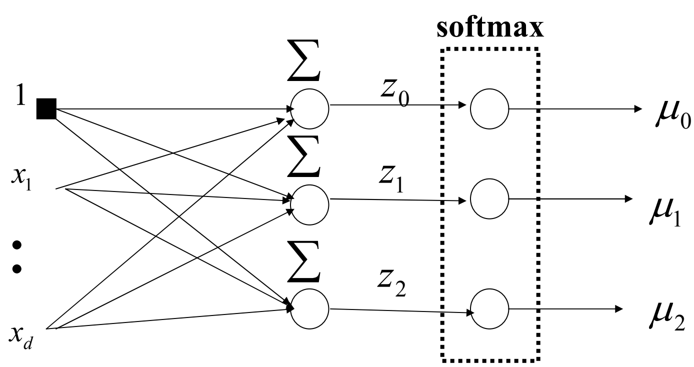


Multi-way classification. Approach 2



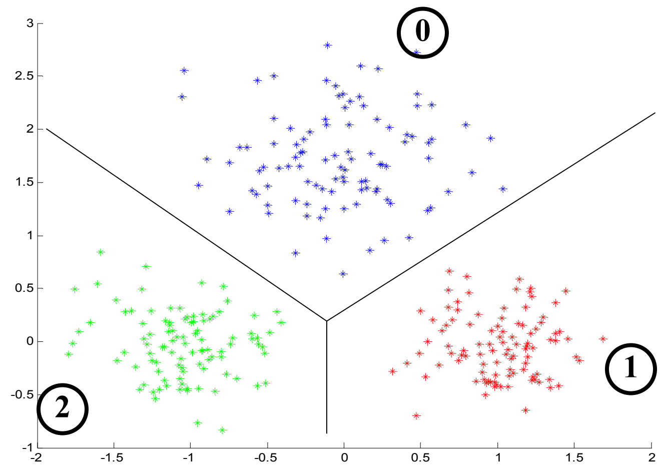
Multi-way classification with softmax

- 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

x → Softmax network → $\mu_0 = P(y=0 \mid \mathbf{x})$ • • • $\mu_{k-1} = P(y=k-1 \mid \mathbf{x})$ → Multi-way Coin toss → y

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**

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

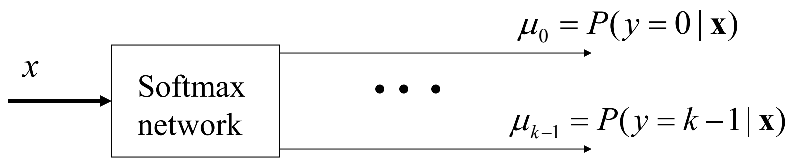
- The same very easy **gradient update** as used for the binary logistic regression

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

- But now we have to update 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:**

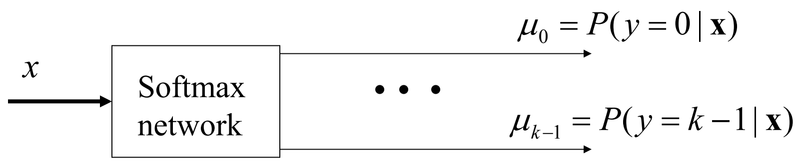
$$p(y = i \mid x) = \frac{p(\mathbf{x} \mid y = i)\, p(y = i)}{\sum_j p(\mathbf{x} \mid y = j)\, p(y = j)}$$

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

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

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



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

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

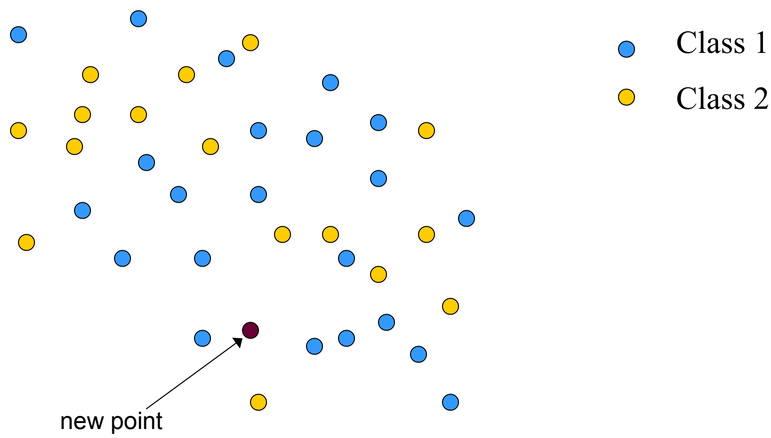
$\boldsymbol{\varphi}$ - scaling parameter (the same for all classes)

CS 2750 Machine Learning
Lecture 13b

Nearest-neighbor classifiers

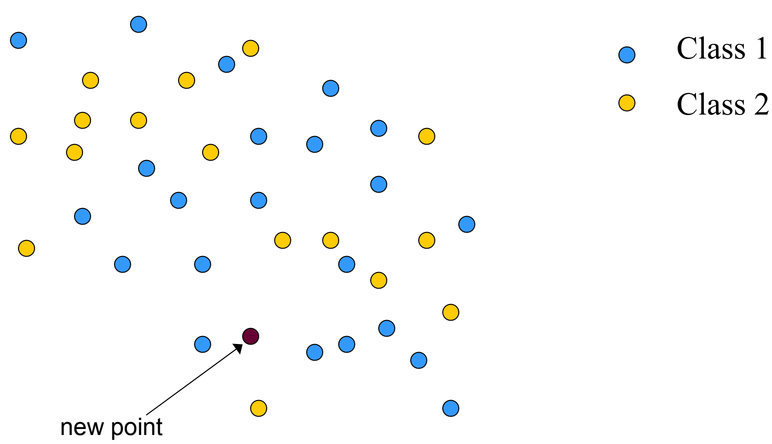
Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Classification problem



What class label would you assign to the new data point ?

Nearest neighbor classification



Class 1 since there are more Class 1 points in its neighborhood

Nearest neighbor classification

Classification:

- **Memory based** – use all examples in the data directly
  - As opposed to a parametric models in which the effect of data is captured by parameters and their values
- **Ramifications:**
  - **No learning (optimization of parameters) is necessary**
  - **All work is done at the time of prediction**

- **Problems:**
  - **Who are the neighbors?**
  - We need a metric to define the neighborhood.

Nearest neighbor classification

Example of a simple metric:

- **Euclidean**

$$D(x, x') = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}$$

- **Nearest neighbor classification:**
  - **K-nearest neighbors: use k examples closest to x**
  - **Nearest neighbor: use a single example closest to x**
- **Decision:**
  - **A simple majority vote on k examples closest to x**
  - **A weighted majority vote on k examples**
    - **A weight defines an importance of a point**
    - **Importance in terms of a distance**

Nearest neighbor classification

- **A weighted majority vote on k examples**
  - **A weight defines an importance of a point**
  - **Importance in terms of a distance**
- An example:
  - **A set of k examples closest to the target x**
    $(x^{(j)}, y^{(j)})$ **with the distance** $D(x, x^{(j)})$

A datapoint $(x^{(j)}, y^{(j)})$ comes with the weight $\dfrac{D(x, x^{(j)})}{\sum_{u=1}^{k} D(x, x^{(u)})}$

Decision: add weights for the same target label
choose the winner