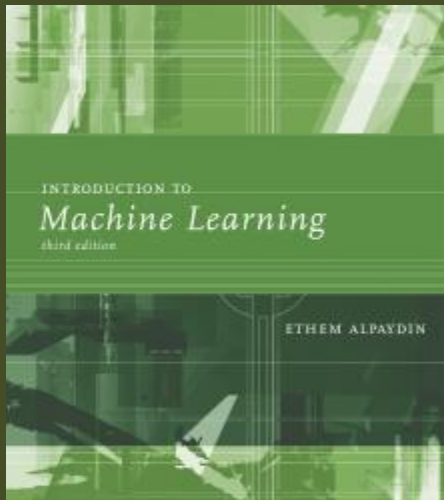


Lecture Slides for

INTRODUCTION TO MACHINE LEARNING

3RD EDITION

ETHEM ALPAYDIN

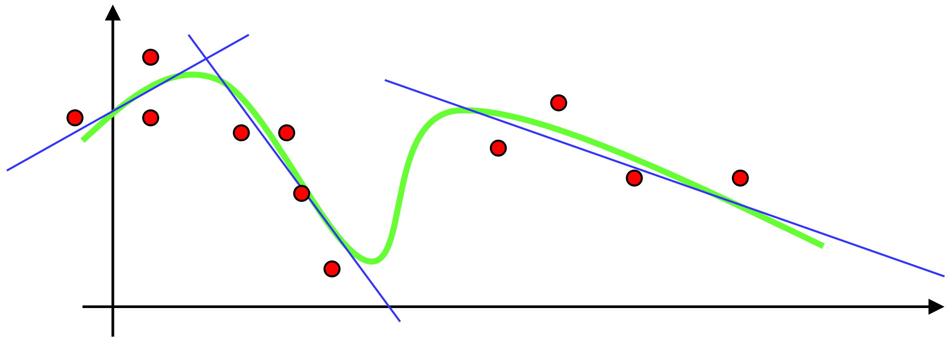
alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

CHAPTER 12:

LOCAL MODELS

Introduction

- Divide the input space into local regions and learn simple (constant/linear) models in each patch



- Unsupervised: Competitive, online clustering
- Supervised: Radial-basis functions, mixture of experts

Competitive Learning

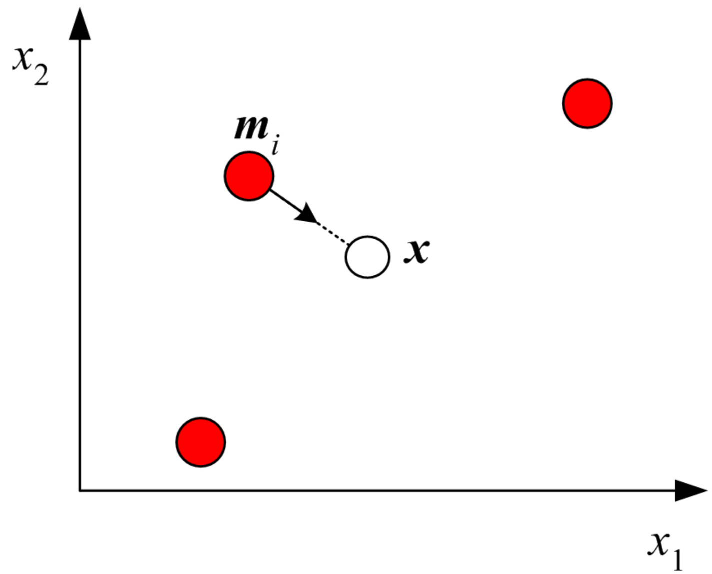
$$E\left(\{\mathbf{m}_i\}_{i=1}^{k} \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|$$

$$b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_l \left\|\mathbf{x}^t - \mathbf{m}_l\right\| \\ 0 & \text{otherwise} \end{cases}$$

Batch k-means: $\mathbf{m}_i = \dfrac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t}$

Online k-means:

$$\Delta m_{ij} = -\eta \frac{\partial E^t}{\partial m_{ij}} = \eta b_i^t \left(x_j^t - m_{ij}\right)$$



Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

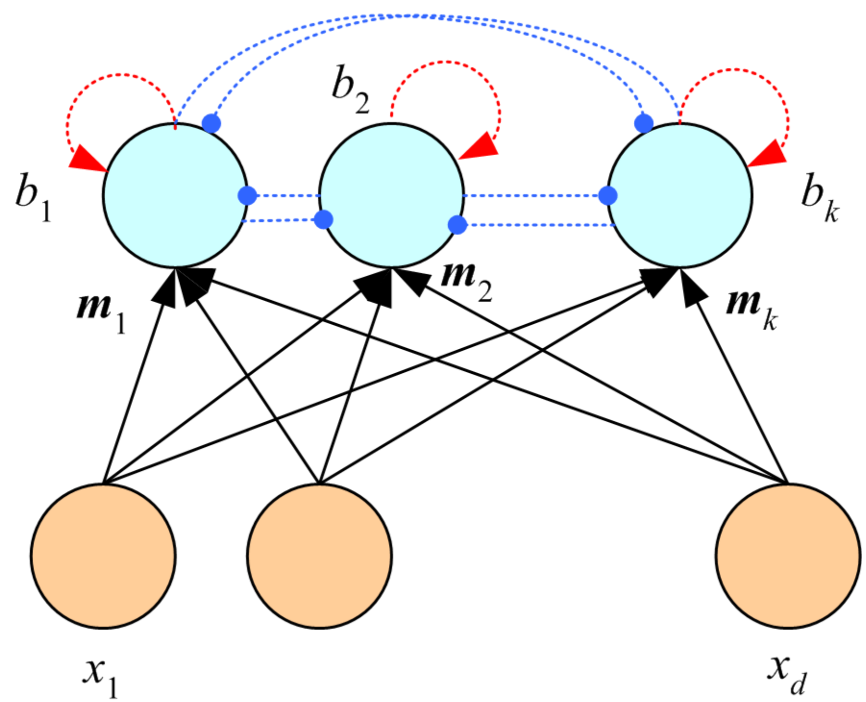
For all $\boldsymbol{x}^t \in \mathcal{X}$ in random order

$i \leftarrow \arg\min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\|$

$\boldsymbol{m}_i \leftarrow \boldsymbol{m}_i + \eta(\boldsymbol{x}^t - \boldsymbol{m}_j)$

Until $\boldsymbol{m}_i$ converge

Winner-take-all network

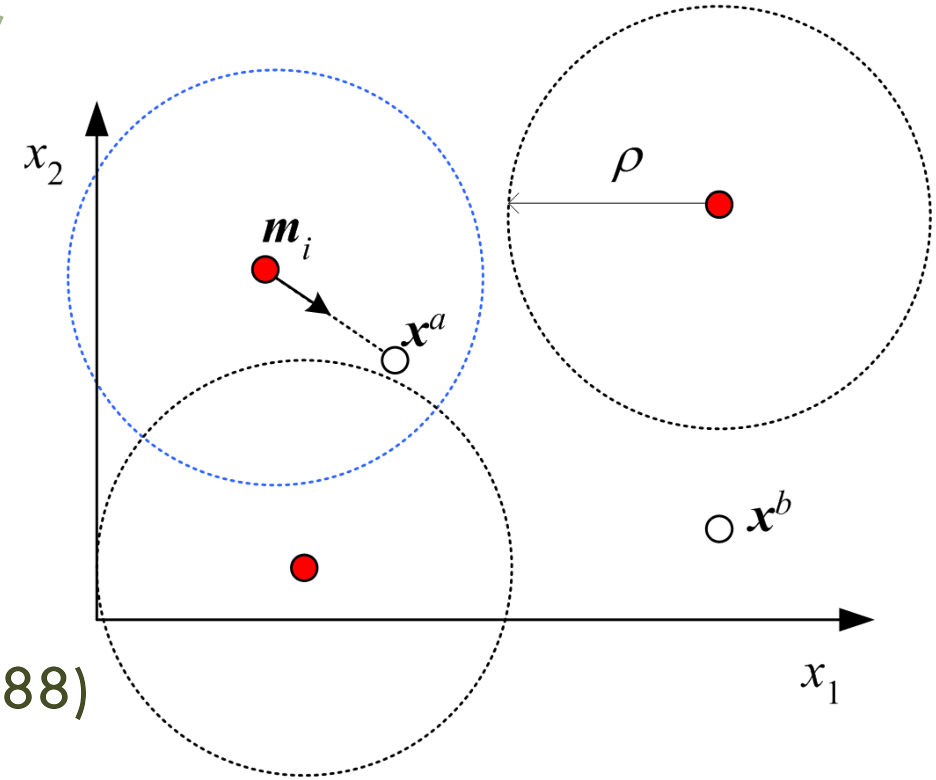


Adaptive Resonance Theory

- Incremental; add a new cluster if not covered; defined by vigilance, ρ

$$b_i^t = \left\| \mathbf{x}^t - \mathbf{m}_i \right\| = -\min_{l=1}^{k} \left\| \mathbf{x}^t - \mathbf{m}_l \right\|$$

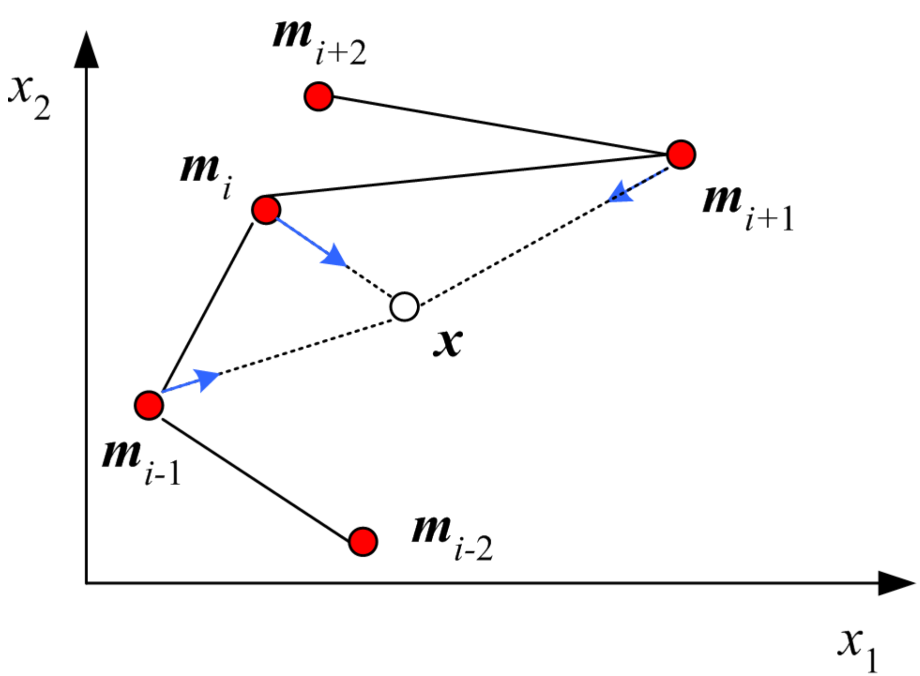
$$\begin{cases} \mathbf{m}_{k+1} \leftarrow \mathbf{x}^t & \text{if } b_i > \rho \\ \Delta \mathbf{m}_i = \eta\left(\mathbf{x}^t - \mathbf{m}_i\right) & \text{otherwise} \end{cases}$$



(Carpenter and Grossberg, 1988)

Self-Organizing Maps

- Units have a neighborhood defined; $\boldsymbol{m}_i$ is "between" $\boldsymbol{m}_{i-1}$ and $\boldsymbol{m}_{i+1}$, and are all updated together
- One-dim map:



(Kohonen, 1990)

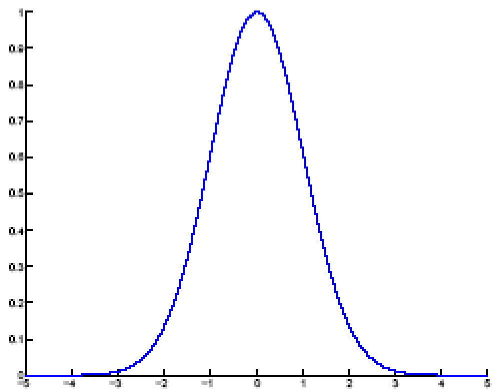
$$\Delta \mathbf{m}_l = \eta e(l,i)\left(\mathbf{x}^t - \mathbf{m}_l\right)$$

$$e(l,i) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(l-i)^2}{2\sigma^2}\right]$$

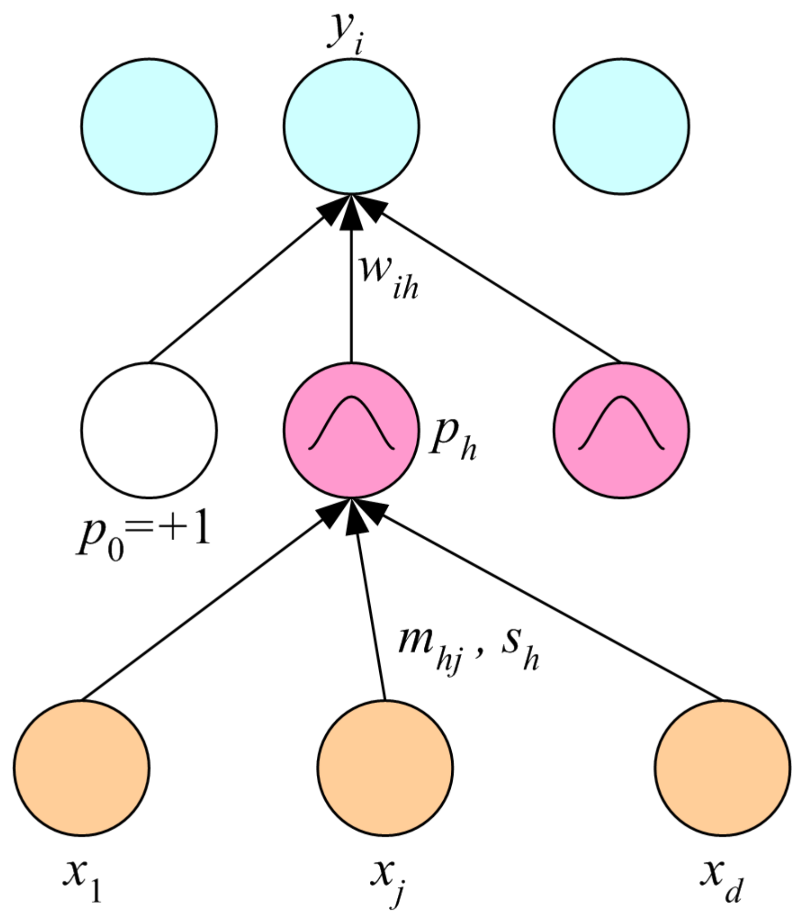
Radial-Basis Functions

- Locally-tuned units:

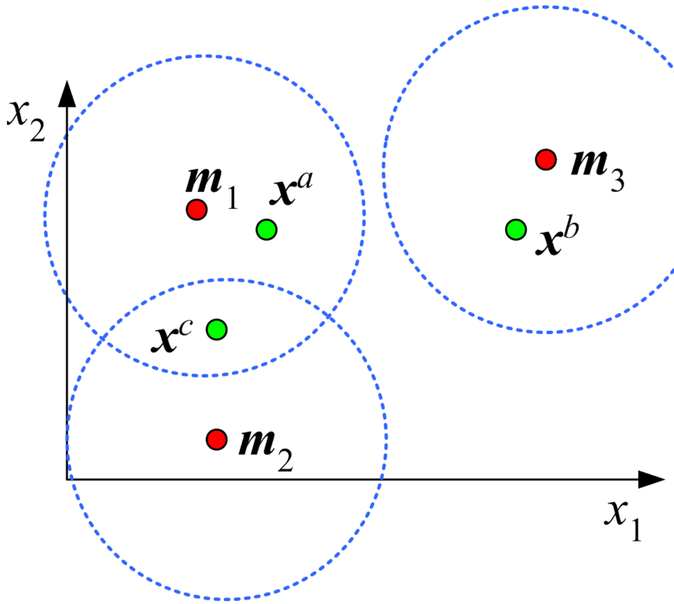
$$p_h^t = \exp\left[-\frac{\left\|\mathbf{x}^t - \mathbf{m}_h\right\|^2}{2s_h^2}\right]$$

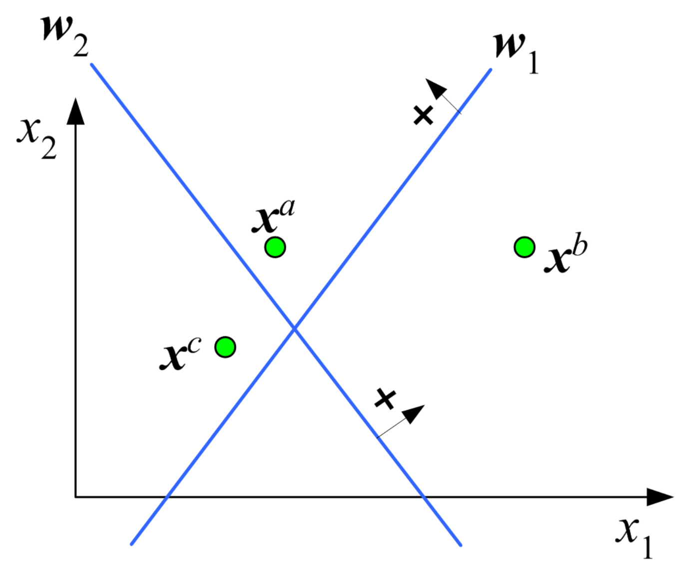


$$y^t = \sum_{h=1}^{H} w_h p_h^t + w_0$$



Local vs Distributed Representation





Local representation in the space of (p_1, p_2, p_3)

$\boldsymbol{x}^a$: $(1.0, 0.0, 0.0)$
$\boldsymbol{x}^b$: $(0.0, 0.0, 1.0)$
$\boldsymbol{x}^c$: $(1.0, 1.0, 0.0)$

Distributed representation in the space of (h_1, h_2)

$\boldsymbol{x}^a$: $(1.0, 1.0)$
$\boldsymbol{x}^b$: $(0.0, 1.0)$
$\boldsymbol{x}^c$: $(1.0, 0.0)$

Training RBF

- Hybrid learning:
  - First layer centers and spreads:

    Unsupervised *k*-means
  - Second layer weights:

    Supervised gradient-descent
- Fully supervised

  (Broomhead and Lowe, 1988; Moody and Darken, 1989)

Regression

$$E\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = \frac{1}{2}\sum_t \sum_i \left(r_i^t - y_i^t\right)^2$$

$$y_i^t = \sum_{h=1}^{H} w_{ih} p_h^t + w_{i0}$$

$$\Delta w_{ih} = \eta \sum_t \left(r_i^t - y_i^t\right) p_h^t$$

$$\Delta m_{hj} = \eta \sum_t \left[\sum_i \left(r_i^t - y_i^t\right) w_{ih}\right] p_h^t \frac{\left(x_j^t - m_{hj}\right)}{s_h^2}$$

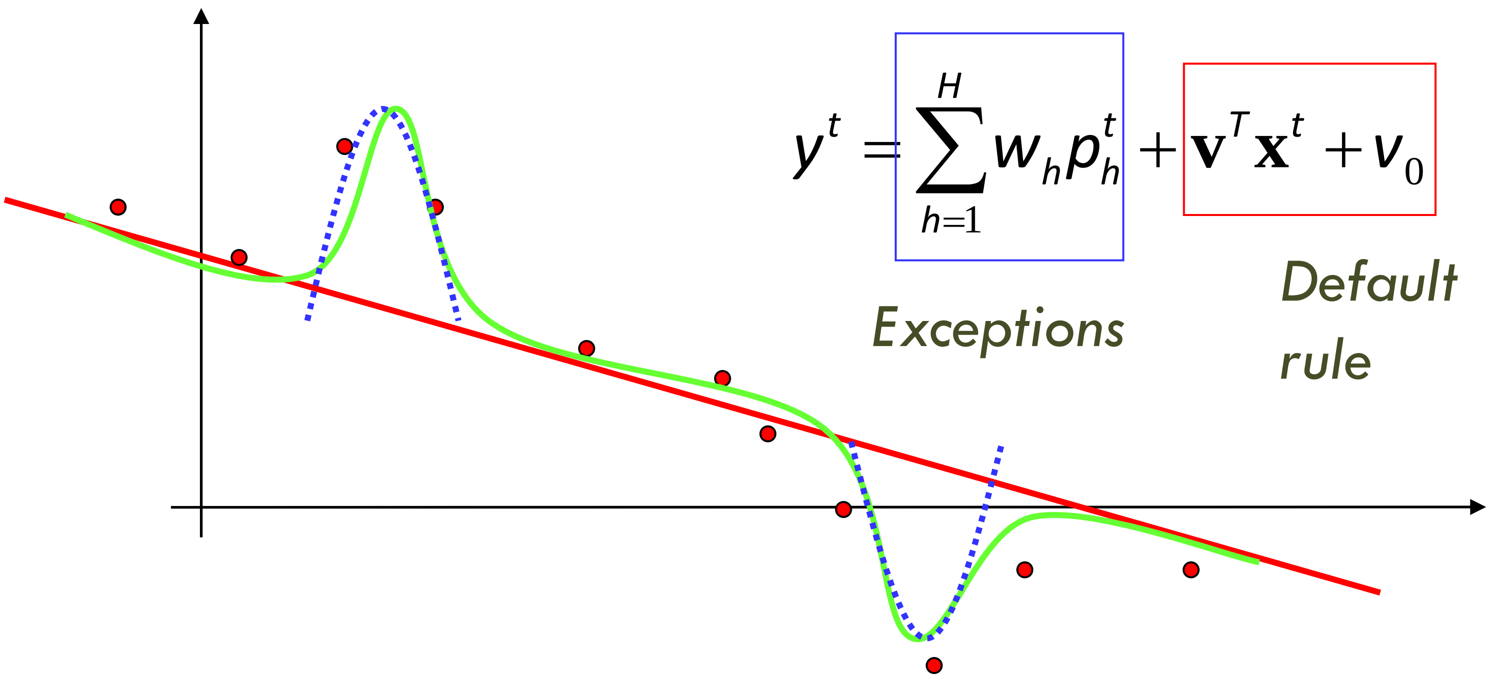
$$\Delta s_h = \eta \sum_t \left[\sum_i \left(r_i^t - y_i^t\right) w_{ih}\right] p_h^t \frac{\left\|\mathbf{x}^t - \mathbf{m}_h\right\|^2}{s_h^3}$$

Classification

$$E\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = -\sum_t \sum_i r_i^t \log y_i^t$$

$$y_i^t = \frac{\exp\left[\sum_h w_{ih} p_h^t + w_{i0}\right]}{\sum_k \exp\left[\sum_h w_{kh} p_h^t + w_{k0}\right]}$$

Rules and Exceptions



$$y^t = \boxed{\sum_{h=1}^{H} w_h p_h^t} + \boxed{\mathbf{v}^T \mathbf{x}^t + v_0}$$

Exceptions

Default rule

Rule-Based Knowledge

$$\text{IF } ((x_1 \approx a) \text{ AND } (x_2 \approx b)) \text{ OR } (x_3 \approx c) \text{ THEN } y = 0.1$$

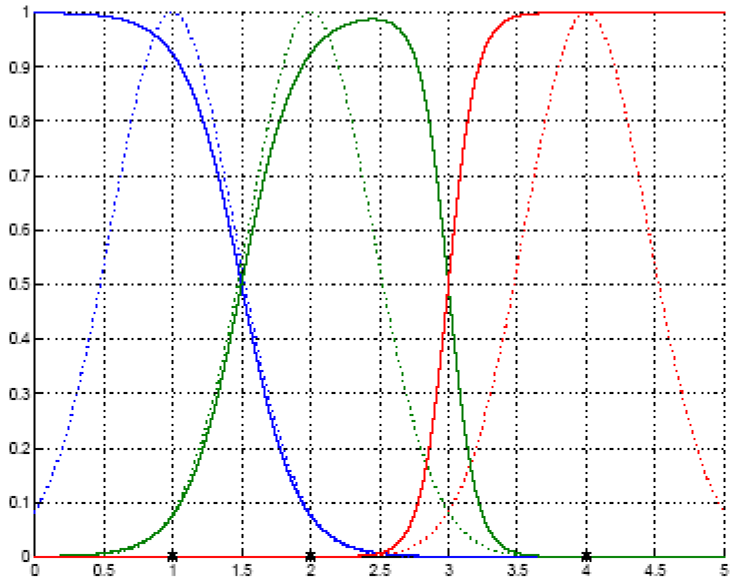
$$p_1 = \exp\left[-\frac{(x_1 - a)^2}{2s_1^2}\right] \cdot \exp\left[-\frac{(x_2 - b)^2}{2s_2^2}\right] \text{ with } w_1 = 0.1$$

$$p_2 = \exp\left[-\frac{(x_3 - c)^2}{2s_3^2}\right] \text{ with } w_2 = 0.1$$

- Incorporation of prior knowledge (before training)
- Rule extraction (after training) (Tresp et al., 1997)
- Fuzzy membership functions and fuzzy rules

Normalized Basis Functions

$$g_h^t = \frac{p_h^t}{\sum_{l=1}^{H} p_l^t} = \frac{\exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_h\right\|^2 / 2s_h^2\right]}{\sum_l \exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_l\right\|^2 / 2s_l^2\right]}$$



$$y_i^t = \sum_{h=1}^{H} w_{ih} g_h^t$$

$$\Delta w_{ih} = \eta \sum_t \left(r_i^t - y_i^t\right) g_h^t$$

$$\Delta m_{hj} = \eta \sum_t \sum_i \left(r_i^t - y_i^t\right)\left(w_{ih} - y_i^t\right) g_h^t \frac{\left(x_j^t - m_{hj}\right)}{s_h^2}$$

Competitive Basis Functions

- Mixture model: $p(\mathbf{r}^t \mid \mathbf{x}^t) = \sum_{h=1}^{H} p(h \mid \mathbf{x}^t) p(\mathbf{r}^t \mid h, \mathbf{x}^t)$

$$p(h \mid \mathbf{x}^t) = \frac{p(\mathbf{x}^t \mid h) p(h)}{\sum_l p(\mathbf{x}^t \mid l) p(l)}$$

$$g_h^t = \frac{a_h \exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_h\right\|^2 / 2s_h^2\right]}{\sum_l a_l \exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_l\right\|^2 / 2s_l^2\right]}$$

Regression

$$p\left(\mathbf{r}^t \mid \mathbf{x}^t\right)=\prod_i \frac{1}{\sqrt{2\pi\sigma}}\exp\left[-\frac{\left(r_i^t-y_i^t\right)}{2\sigma^2}\right]$$

$$\mathcal{L}\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right)=\sum_t \log \sum_h g_h^t \exp\left[-\frac{1}{2}\sum_i \left(r_i^t-y_{ih}^t\right)^2\right]$$

$y_{ih}^t = w_{ih}$ is the constant fit

$$\Delta w_{ih}=\eta\sum_t \left(r_i^t-y_{ih}^t\right)f_h^t \qquad \Delta m_{hj}=\eta\sum_t \left(f_h^t-g_h^t\right)\frac{\left(x_j^t-m_{hj}\right)}{s_h^2}$$

$$f_h^t=\frac{g_h^t \exp\left[-(1/2)\sum_i \left(r_i^t-y_{ih}^t\right)^2\right]}{\sum_l g_l^t \exp\left[-(1/2)\sum_i \left(r_i^t-y_{il}^t\right)^2\right]}$$

$$p(h \mid \mathbf{r},\mathbf{x})=\frac{p(h \mid \mathbf{x})\,p(\mathbf{r} \mid h,\mathbf{x})}{\sum_l p(l \mid \mathbf{x})\,p(\mathbf{r} \mid l,\mathbf{x})}$$

Classification

$$\mathcal{L}\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = \sum_t \log \sum_h g_h^t \prod_i \left(y_{ih}^t\right)^{r_i^t}$$

$$= \sum_t \log \sum_h g_h^t \exp\left[\sum_i r_i^t \log y_{ih}^t\right]$$

$$y_{ih}^t = \frac{\exp w_{ih}}{\sum_k \exp w_{kh}}$$

$$f_h^t = \frac{g_h^t \exp\left[\sum_i r_i^t \log y_{ih}^t\right]}{\sum_l g_l^t \exp\left[\sum_i r_i^t \log y_{il}^t\right]}$$

EM for RBF (Supervised EM)

- E-step:

$$f_h^t \equiv p\left(\mathbf{r} \mid h, \mathbf{x}^t\right)$$

- M-step:

$$\mathbf{m}_h = \frac{\sum_t f_h^t \mathbf{x}^t}{\sum_t f_h^t}$$

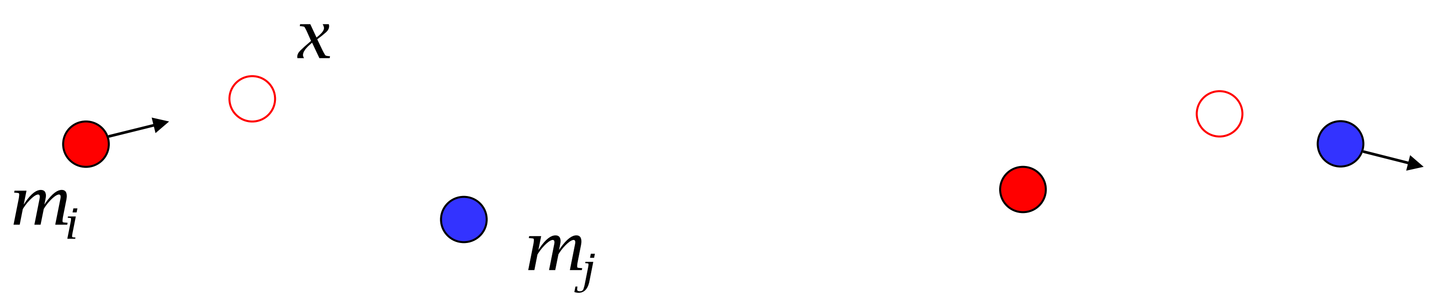
$$s_h = \frac{\sum_t f_h^t \left(\mathbf{x}^t - \mathbf{m}_h\right)\left(\mathbf{x}^t - \mathbf{m}_h\right)^T}{\sum_t f_h^t}$$

$$w_{ih} = \frac{\sum_t f_h^t r_i^t}{\sum_t f_h^t}$$

Learning Vector Quantization

- *H* units per class prelabeled (Kohonen, 1990)
- Given **x**, $\boldsymbol{m}_i$ is the closest:

$$\begin{cases} \Delta \mathbf{m}_i = \eta\left(\mathbf{x}^t - \mathbf{m}_i\right) & \text{if label}(\mathbf{x}^t) = \text{label}(\mathbf{m}_i) \\ \Delta \mathbf{m}_i = -\eta\left(\mathbf{x}^t - \mathbf{m}_i\right) & \text{otherwise} \end{cases}$$

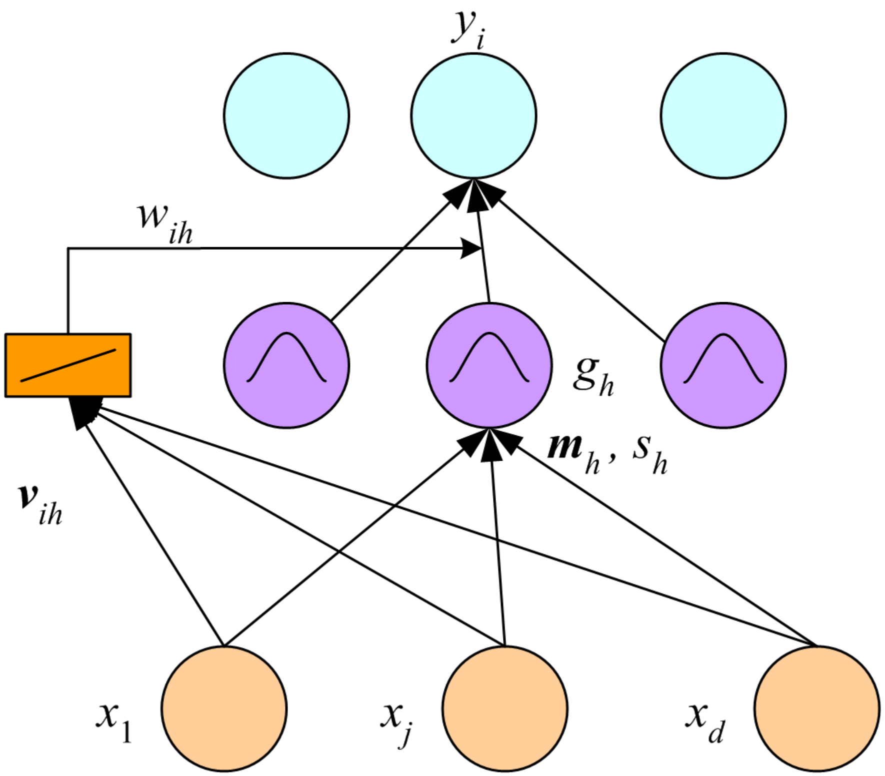


Mixture of Experts

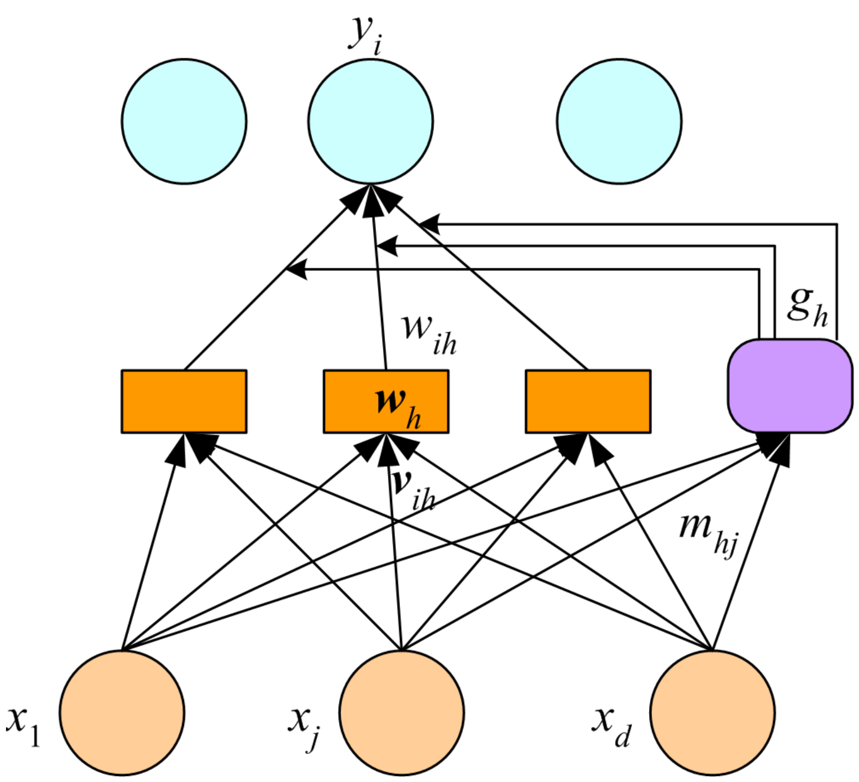
- In RBF, each local fit is a constant, w_{ih}, second layer weight
- In MoE, each local fit is a linear function of x, a local expert:

$$w_{ih}^t = \mathbf{v}_{ih}^T \mathbf{x}^t$$

(Jacobs et al., 1991)



MoE as Models Combined



- Radial gating:

$$g_h^t = \frac{\exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_h\right\|^2 / 2s_h^2\right]}{\sum_l \exp\left[-\left\|\mathbf{x}^t - \mathbf{m}_l\right\|^2 / 2s_l^2\right]}$$

- Softmax gating:

$$g_h^t = \frac{\exp\left[\mathbf{m}_h^T \mathbf{x}^t\right]}{\sum_l \exp\left[\mathbf{m}_l^T \mathbf{x}^t\right]}$$

Cooperative MoE

- Regression

$$E\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = \frac{1}{2}\sum_t \sum_i \left(r_i^t - y_i^t\right)^2$$

$$\Delta \mathbf{v}_{ih} = \eta \sum_t \left(r_i^t - y_{ih}^t\right) g_h^t \mathbf{x}^t$$

$$\Delta m_{hj} = \eta \sum_t \left(r_i^t - y_{ih}^t\right)\left(w_{ih}^t - y_i^t\right) g_h^t x_j^t$$

Competitive MoE: Regression

$$\mathcal{L}\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = \sum_t \log \sum_h g_h^t \exp\left[-\frac{1}{2}\sum_i \left(r_i^t - y_{ih}^t\right)^2\right]$$

$$y_{ih}^t = w_{ih} = \mathbf{v}_{ih}\mathbf{x}^t$$

$$\Delta \mathbf{v}_{ih} = \eta \sum_t \left(r_i^t - y_{ih}^t\right) f_h^t \mathbf{x}^t$$

$$\Delta \mathbf{m}_h = \eta \sum_t \left(f_h^t - g_h^t\right) \mathbf{x}^t$$

Competitive MoE: Classification

$$\mathcal{L}\left(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} \mid \mathcal{X}\right) = \sum_t \log \sum_h g_h^t \prod_i \left(y_{ih}^t\right)^{r_i^t}$$

$$= \sum_t \log \sum_h g_h^t \exp\left[\sum_i r_i^t \log y_{ih}^t\right]$$

$$y_{ih}^t = \frac{\exp w_{ih}}{\sum_k \exp w_{kh}} = \frac{\exp \mathbf{v}_{ih}\mathbf{x}^t}{\sum_k \exp \mathbf{v}_{kh}\mathbf{x}^t}$$

Hierarchical Mixture of Experts

- Tree of MoE where each MoE is an expert in a higher-level MoE
- Soft decision tree: Takes a weighted (gating) average of all leaves (experts), as opposed to using a single path and a single leaf
- Can be trained using EM (Jordan and Jacobs, 1994)