KHU-BME
Pattern Classification
Lecture 06

# Multi-category Generalization: Multi-class Perceptron (DHS 5.12)

Decision Rule:

$g_i(\underline{x})=\underline{w}^{(i)T}\underline{x}$

if $g_i(\underline{x}) > g_j(\underline{x})$ for all $j \neq i$, assign $\underline{x}$ to $S_i$

Algorithm (given in class):

For each prototype $\underline{y}^{(i)}$

If $\underline{y}^{(i)} \in S_i$ but machine assigns it to $S_j$

Update for $S_i$, $w^{(i)}(k+1)=\underline{w}^{(i)}(k)+\alpha y^{(i)}$

Update for $S_j$, $w^{(j)}(k+1)=\underline{w}^{(j)}(k)-\alpha y^{(i)}$

$w^{(l)}(k+1)=\underline{w}^{(l)}(k)$ all $l \neq i,j$ (DHS p. 267, Eq. (115))

If machine classifies $\underline{y}^{(i)}$ correctly, do not update the prototype.

The algorithm is guaranteed to converge (for a fixed increment).

Convergence of algorithm is proved in DHS Sec. 5.12.2 (not covered in class)

**Absolute Correction (one example):**

If machine puts $\underline{y}_m^{(k)}$ into $S_j$

Want to:

$$w_k^T(i+1)\underline{y}_m^{(k)} > w_j^T(i+1)\underline{y}_m^{(k)}$$

$$[w_k(i)+\alpha\underline{y}_m^{(k)}]^T\underline{y}_m^{(k)} > [w_j(i)-\alpha\underline{y}_m^{(k)}]^T\underline{y}_m^{(k)}$$

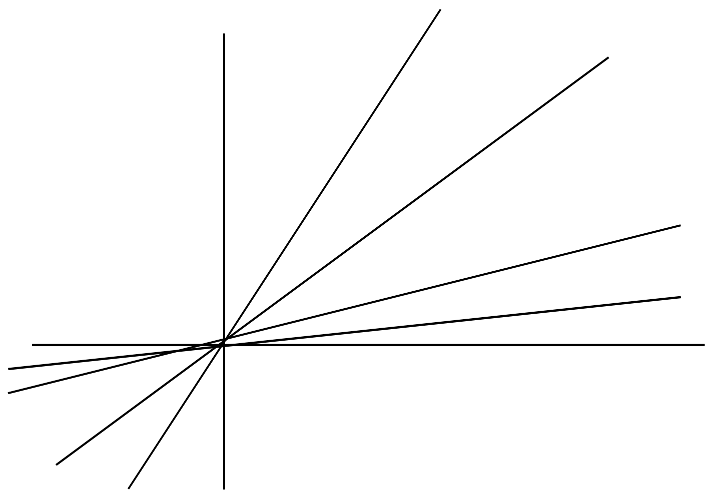
$$w_k^T(i)\underline{y}_m^{(k)} + \alpha\underline{y}_m^{(k)^T}\underline{y}_m^{(k)} > w_j^T(i)\underline{y}_m^{(k)} - \alpha\underline{y}_m^{(k)^T}\underline{y}_m^{(k)}$$

Guarantees re-classification of $\underline{y}_m^{(k)}$ will not result in the same error

$$\alpha = \frac{\lfloor \underline{w}_j^T(i) - \underline{w}_k^T(i) \rfloor \underline{y}_m^{(k)}}{2\underline{y}_m^{(k)T}\underline{y}_m^{(k)}}$$

Doesn't guarantee correct classification of $\underline{y}_m^{(k)}$ after 1 update.

## <u>Relaxation Procedures</u> (DHS 5.6, p. 235)



Augmented space incorporates a safety margin.
Gives a smoother surface than Perceptron

Perceptron: $J(\underline{w})=\sum_{\underline{y}\in\underline{Y}}(-\underline{w}^T\underline{y})$ (DHS p. 227, Eq. (16))

New cost: $J(\underline{w})=\sum_{\underline{y}\in\underline{Y}}(\underline{w}^T\underline{y})^2$ (DHS p. 235, Eq. (32))

See Fig. 5.11 for the gradients of various $J(\underline{w})$'s

**<u>Relaxation</u>**

Cost function with margin:

$$J(\underline{w})=\frac{1}{2}\sum_{\underline{y}\in\underline{Y}}\frac{[\underline{w}^T\underline{y}-b]^2}{\|\underline{y}\|^2}$$ (DHS p. 235, Eq. (33))

b is a margin factor

$\underline{Y}$ represents the set of prototypes for which $\underline{w}^T\underline{y}\le b$

if $\underline{Y}=\varnothing$ then $J(\underline{w})=0$

$J(\underline{w})\ge 0$

$J(\underline{w})=0$ only of $\underline{w}^T\underline{y}\ge b$, $\forall\underline{y}$

The gradient of the cost function with margin:

$$\nabla_{\underline{w}} J(\underline{w}) = \sum \frac{(\underline{w}^T \underline{y} - b)}{\|\underline{y}\|^2} \underline{y}$$

(Appendix A2.4 for vector matrix calculus)

Gradient Descent:

$\underline{w}(k+1)=\underline{w}(k)-\alpha(k)\nabla J(\underline{w})$

Relaxation Procedure

**i) One at a time = Single-sample Relaxation with Margin (DHS, p. 236 Algorithm 9)**

$\underline{w}(0)$=arbitrary

$$\underline{w}(k+1)=\underline{w}(k)- \alpha(k) \frac{\underline{w}^T \underline{y}_k - b}{\|\underline{y}_k\|^2} \underline{y}_k \qquad \text{if } \underline{w}^T(k)\underline{y}_k \le b$$

$\underline{w}(k+1)=\underline{w}(k)$ if $\underline{w}^T(k)\underline{y}_k > b$

**ii) Many at a time = Batch Relaxation with Margin (DHS, p. 236 Algorithm 8)**

If $\underline{w}(0)$=arbitrary

$$\underline{w}(k+1)=\underline{w}(k)-\alpha(k) \sum \frac{\underline{w}^T \underline{y} - b}{\|\underline{y}\|^2} \underline{y}$$

**So far,**

**Previous Methods - Error Correcting**

- **Update only for misclassified samples**
- **Search for error-free solutions**