Brief summary on Lecture 1

- Bayesian decision rule:

$$\delta(x): \begin{array}{l} P(H_1|x) \geq P(H_0|x) \Rightarrow \Gamma_1 \Rightarrow H_1 \\ P(H_1|x) < P(H_0|x) \Rightarrow \Gamma_0 \Rightarrow H_0 \end{array}$$

- MAP decision rule:

$$H_j = \underset{j=0,1,\ldots,M\text{-}1}{\operatorname{argmax}} P(H_j|x)$$

$$\text{-}\, x \Rightarrow \boxed{f(x,\theta)} \Rightarrow y = f(x,\theta)$$

$$\theta = \underset{\theta}{\operatorname{argmin}} \int \left[f(x,\theta) - d(x) \right]^2 p(x)\, dx$$

$$d(x) = P(C_1|x)$$

- Remark: $f(x,\theta) \Rightarrow d(x) = P(C_1|x)$ is what the prediction model is learning to approximate in a MSE sense, where the weight is $p(x)$.

Statistical learning from "data"

- For binary classification, we have

$$P\left(C_1 \middle| x\right) \geq P\left(C_2 \middle| x\right) \Rightarrow \text{since } P\left(C_2 \middle| x\right) = 1 - P\left(C_1 \middle| x\right)$$

$$\Rightarrow P\left(C_1 \middle| x\right) \geq 0.5 \Rightarrow H_1 \text{ or } C_1.$$

- For multiclass classification, we have

$$x \Rightarrow \boxed{f\left(x,\theta\right)} \Rightarrow y = \begin{cases} f_1\left(x,\theta\right) \rightarrow C_1 \\ f_2\left(x,\theta\right) \rightarrow C_2 \\ f_3\left(x,\theta\right) \rightarrow C_3 \end{cases}$$

- Class coding schemes:

  - one-versus-rest (OVR)

  - Winner-Takes-All (WTA)

$$\varepsilon = \sum_{i=1}^{3} \int \left[f_i\left(x,\theta\right) - P\left(C_i \middle| x\right) \right]^2 p\left(x\right) dx$$

Statistical learning from "data"

- Output code assignment:

	$f_1(x,\theta)$	$f_2(x,\theta)$	$f_3(x,\theta)$
$x \in C_1$	1	0	0
$x \in C_2$	0	1	0
$x \in C_3$	0	0	1

- Remark: The training of multiclass predictor shows that the parameter θ of the prediction model are being "adjusted/learned" to simultaneously "approximate" multiple yet different output fnuctions, i.e., the Bayes posterior class probabilities.

- Remark: since $\sum_{i=1}^{3} f_i(x,\theta) \neq 1$, "normalization" over $f_i(x,\theta)$ is required, see Gish's paper.

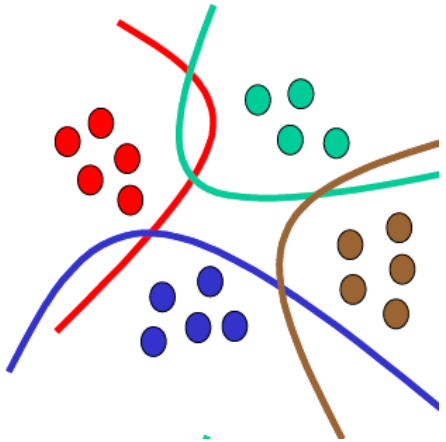
* HW-2: (1) seklf-reading Gish's paper and Chp 3; (2) derivation of cost/objective/criterion function for multiclass predictor learning.

Linear Discriminant Analysis (LDA)

- We are talking about Chapter 5 now.

- Suppose that each class conditional probability density function is a d-dimensional multivariate Gaussian:

$$p_k(x) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$



- Linear discriminant analysis (LDA) arises in the special case when:

$$\mathbf{\Sigma}_k = \mathbf{\Sigma}, \ \forall k$$

i.e., all classes share the common covariance matrix.

Linear Discriminant Analysis (LDA)

- Consider the log-ratio between any two classes, we have:

$$\log\frac{P\left(C_k\middle|\mathbf{x}\right)}{P\left(C_l\middle|\mathbf{x}\right)}=\log\frac{\pi_k}{\pi_l}+\log\frac{p_k\left(\mathbf{x}\right)}{p_l\left(\mathbf{x}\right)}$$

$$=\log\frac{\pi_k}{\pi_l}-\frac{1}{2}\left(\boldsymbol{\mu}_k+\boldsymbol{\mu}_l\right)^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}_k-\boldsymbol{\mu}_l\right)+\boxed{\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}_k-\boldsymbol{\mu}_l\right)}$$

- Remark: $\mathbf{s}^T\mathbf{s},\mathbf{s}^T\mathbf{x},\mathbf{s}^T\mathbf{A}\mathbf{s},\mathbf{s}^T\mathbf{A}\mathbf{x},$ are all called in "quadratic form" and are scalar!

- Note that the log-ratio between any two classes is an expression "linear in $\mathbf{x}$", i.e., the decision boundary between class k and class l is linear in $\mathbf{x}$ in a d-dimensional hyperplane.

Linear Discriminant Analysis (LDA)

- We can then define the "linear discriminant functions" log[p(x|k)]

$$\delta_j(\mathbf{x}) = \boxed{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j} + \log \pi_j - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j$$

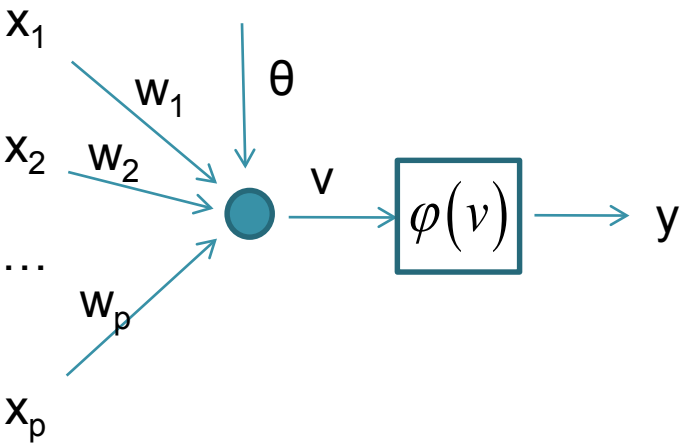
and accordingly, based on the MAP decision rule, to define the equivalent decision rule as:

$$k = \arg\max_j \delta_j(\mathbf{x}).$$

- Remark: LDA requires Gaussian assumption and common covariance matrix! If the data distribuition is not Gaussian with common covariance matrix, the LDA will not be accurate. Thus, it makes sense to find directly the decision boundary that empirically minimizes the training error.

Single-Layer Perceptron (SLP)

- Consider a linearly separable binary classification problem. We first define the "single neuron model":

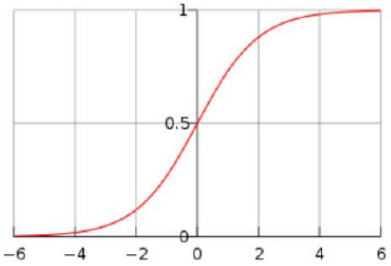


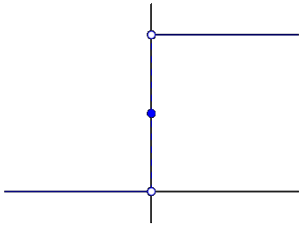
where w_i is the "synaptic weight", θ is the bias/threshold.

- Note that the output v is a linear weighted combination of the inputs

$$v = \sum_{i=1}^{p} w_i x_i + \theta, \quad y = \varphi(v),$$

where $\varphi(.)$ is the activation function (e.g., sigmiod, hard-limit).





What purpose? recall the interpretation of output

Single-Layer Perceptron (SLP)

- Logic/natural decision boundary:

$$\sum_{i=1}^{p} w_i x_i + \theta = 0$$

i.e., a linear equation of x_i determined by all the coefficient w_i.

- SLP learning algorithm:

let $\mathbf{x} = \left[+1, x_1(n), x_2(n), ..., x_p(n)\right]^T$,

$\mathbf{w} = \left[\theta, w_1(n), w_2(n), ..., w_p(n)\right]$,

where n denotes both the index of input data point vector and the index of iteration in the sequential learning process.

- We have $v(n) = \mathbf{w}^T(n)\mathbf{x}(n)$, and the decision boundary can be re-expressed as $\mathbf{w}^T(n)\mathbf{x}(n) = 0$.

How to interpret weight vector w?
recall the 'norm' of a hyperplane

Single-Layer Perceptron (SLP)

- Let the training sample be: $\mathbf{X}_1 \in C_1, \mathbf{X}_2 \in C_2$.

Our objective is to find $\mathbf{w}$ such that

$$\begin{cases} \mathbf{w}^T\mathbf{x} \geq 0 & \mathbf{x} \in C_1 \\ \mathbf{w}^T\mathbf{x} < 0 & \mathbf{x} \in C_2 \end{cases}$$

- Let $\begin{cases} e(n) = d(n) - y(n), \\ \Delta\mathbf{w}(n) = \eta e(n)\mathbf{x}(n), \\ \mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n). \end{cases}$

- How do we come here? Gradient descent/asceent method to find (next slide)

$$\text{MMSE: } \frac{1}{2}\frac{\partial\left[e^2(n)\right]}{\partial\mathbf{w}(n)} = \frac{1}{2}\cdot 2e(n)\frac{\partial\left[d(n)-y(n)\right]}{\partial\mathbf{w}(n)} = -e(n)\frac{\partial\left[y(n)\right]}{\partial v(n)}\frac{\partial\left[v(n)\right]}{\partial\mathbf{w}(n)}$$

$$= -e(n)\varphi'\mathbf{x}(n) \Rightarrow \Delta\mathbf{w}(n) = -\eta\left(\frac{1}{2}\frac{\partial\left[e^2(n)\right]}{\partial\mathbf{w}(n)}\right) = \eta e(n)\mathbf{x}(n).$$

Single-Layer Perceptron (SLP)

Gradient Descent Rule:

$$\underline{\mathbf{w}}_{new} = \underline{\mathbf{w}}_{old} - \eta \, \Delta\varepsilon(\underline{\mathbf{w}})$$

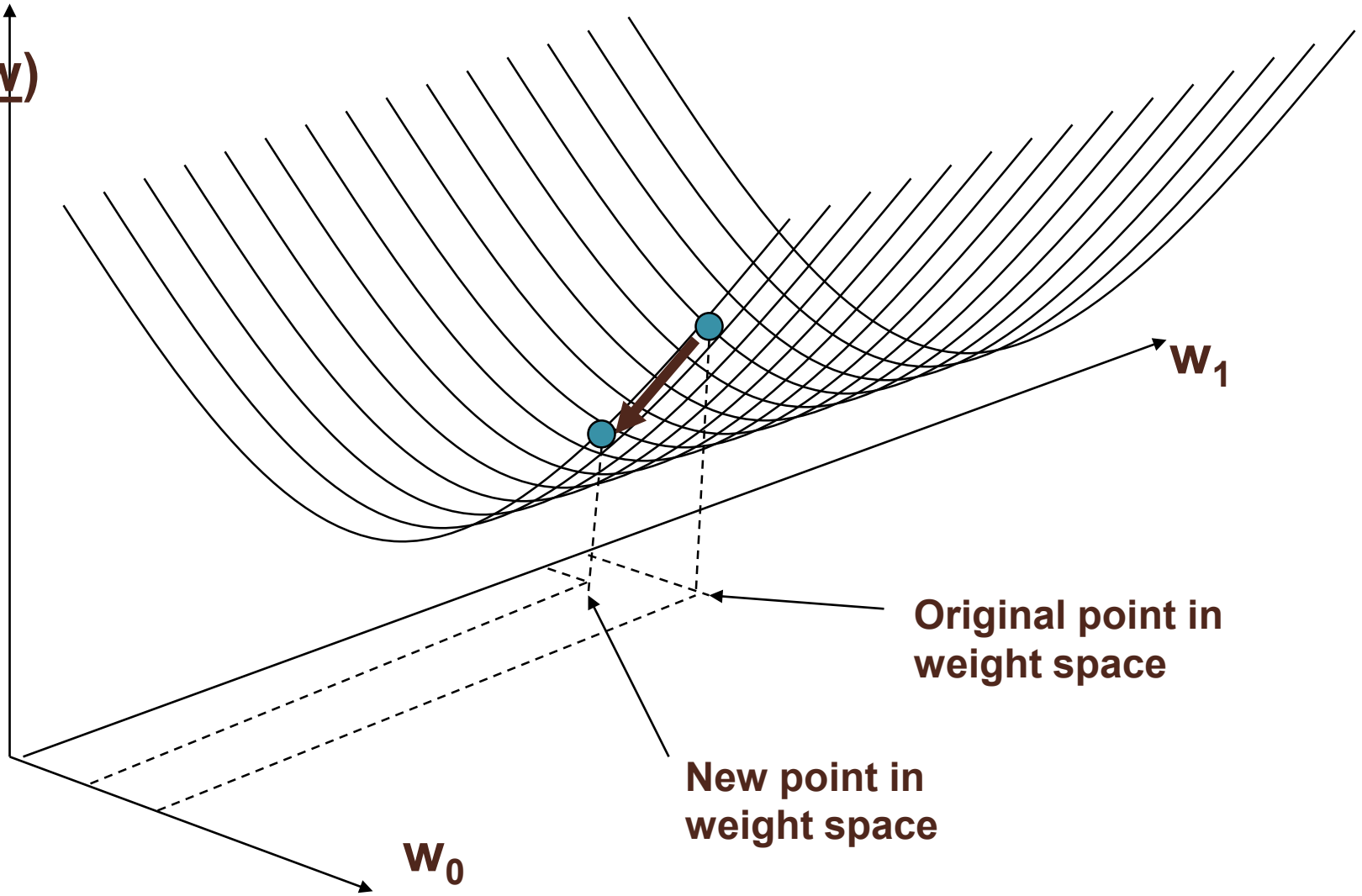
where

$\Delta(\underline{w})$ is the gradient and
η is the learning rate (small, positive)

Notes:

1. This moves us downhill in direction $\Delta\varepsilon(\underline{\mathbf{w}})$ (steepest downhill direction)
2. How far we go is determined by the value of η (learning rate)
3. The perceptron learning rule is a special case of this general method

Single-Layer Perceptron (SLP)



Single-Layer Perceptron (SLP)

- Consistent with "error-correction learning" geometric interpretation

Case 1: For correct classifications, no update is needed, i.e.,

$$\text{If } \begin{cases} \mathbf{w}^T\mathbf{x} \geq 0 & \mathbf{x} \in C_1 \\ \mathbf{w}^T\mathbf{x} < 0 & \mathbf{x} \in C_2 \end{cases}, \text{ then } \mathbf{w}(n+1) = \mathbf{w}(n).$$

Case 2: For incorrect classifications, update is needed, i.e.,

Case 2-1: If $\mathbf{x} \in C_2$ while $\mathbf{w}^T\mathbf{x} \geq 0$, then $e(n) = -1$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta\mathbf{x}(n) \quad \text{or}$$

$$\mathbf{w}^T(n+1)\mathbf{x}(n) = \mathbf{w}^T(n)\mathbf{x}(n) - \eta\mathbf{x}^T(n)\mathbf{x}(n)$$

Case 2-2: If $\mathbf{x} \in C_1$ while $\mathbf{w}^T\mathbf{x} < 0$, then $e(n) = +1$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta\mathbf{x}(n) \quad \text{or}$$

$$\mathbf{w}^T(n+1)\mathbf{x}(n) = \mathbf{w}^T(n)\mathbf{x}(n) + \eta\mathbf{x}^T(n)\mathbf{x}(n)$$

Single-Layer Perceptron (SLP)

- Remark: convergence of the learning algorithm has been proved.
- Relation of SLP to LDA (Bayes classifier):

$$y = \mathbf{w}^T \mathbf{x} + b,$$

$$\text{where } \mathbf{w} = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\right), b = \frac{1}{2}\left(\boldsymbol{\mu}_l^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_l^T - \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k^T\right).$$

- Reading assignment: Chapter 5 (no * sections).
- Extension of SLP to multiclass classification?

  Reading paper by Khan et al.