Artificial Intelligence

Linear Models for Classification

Donghui Wang
AI Institute@ZJU

2015.4

Contents

- Basic Concepts
- Discriminant Functions
- Probabilistic Generative Models
- Probabilistic Discriminative Models
- The Laplace Approximation
- Bayesian Logistic Regression

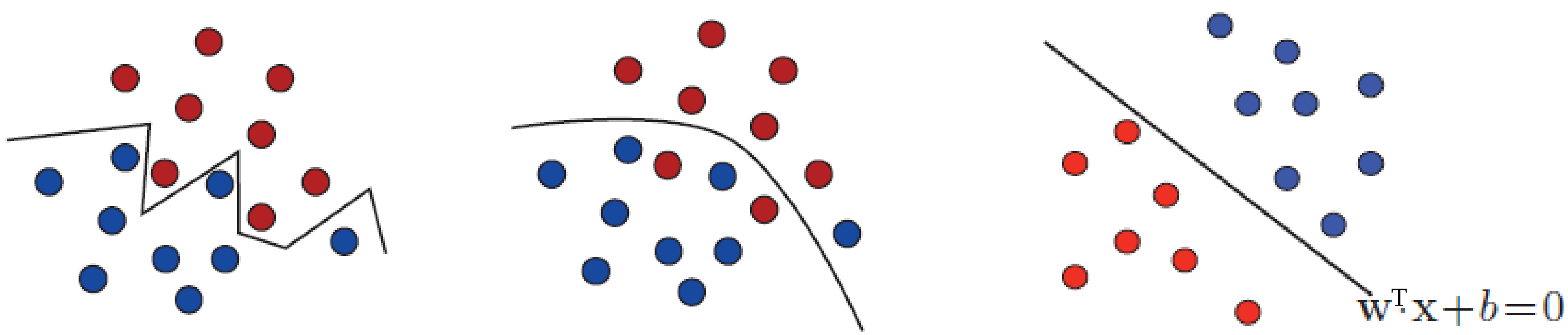
References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 4. 2006.*

Basic Concepts

Linearly separable

- Decision regions:
  - Input space is divided into several regions
- Decision boundaries (surfaces):
  - Under linear models, it's a linear function of the input vector x
  - (D-1)-dimensional hyper-plane within the D-dimensional input space
- Data sets whose classes can be separated exactly by linear decision surfaces are said to be *linear separable*.



Representation of Class Labels

- Two classes (K=2):
  - Target variable $t \in \{0,1\}$, $t=1$ represents class C_1, else class C_2
- K-classes (K>2):
  - 1-of-K coding scheme: $\mathbf{t} = (0, 1, 0, 0, 0)^T$

W : weight vector

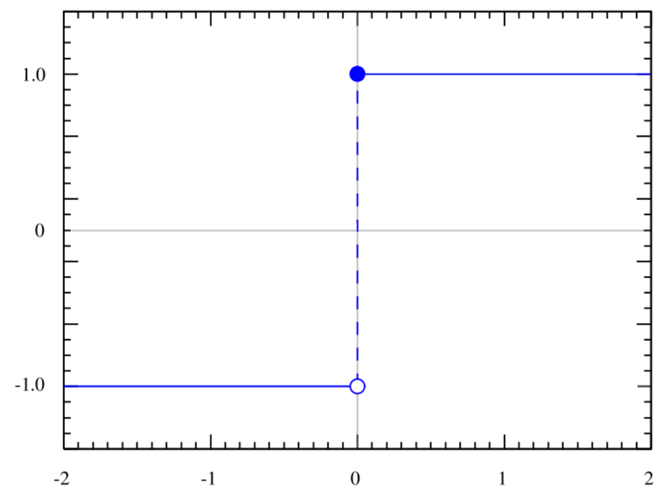
w_0 : bias/threshold

- Predict discrete class labels:
  - Linear model prediction (linear discriminant function): $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$
  - Nonlinear function $f(.) : R \rightarrow (0, 1)$
  - Generalized linear models:

$f(\cdot)$: activation function

$f^{-1}(\cdot)$: link function

$$y(\mathrm{x}) = f\left(\mathrm{w}^{\mathrm{T}}\mathrm{x} + w_0\right)$$



  - Decision surface:
    - $y(\mathbf{x}) = \text{constant} \rightarrow \mathbf{w}^T\mathbf{x} + w_0 = \text{constant}$

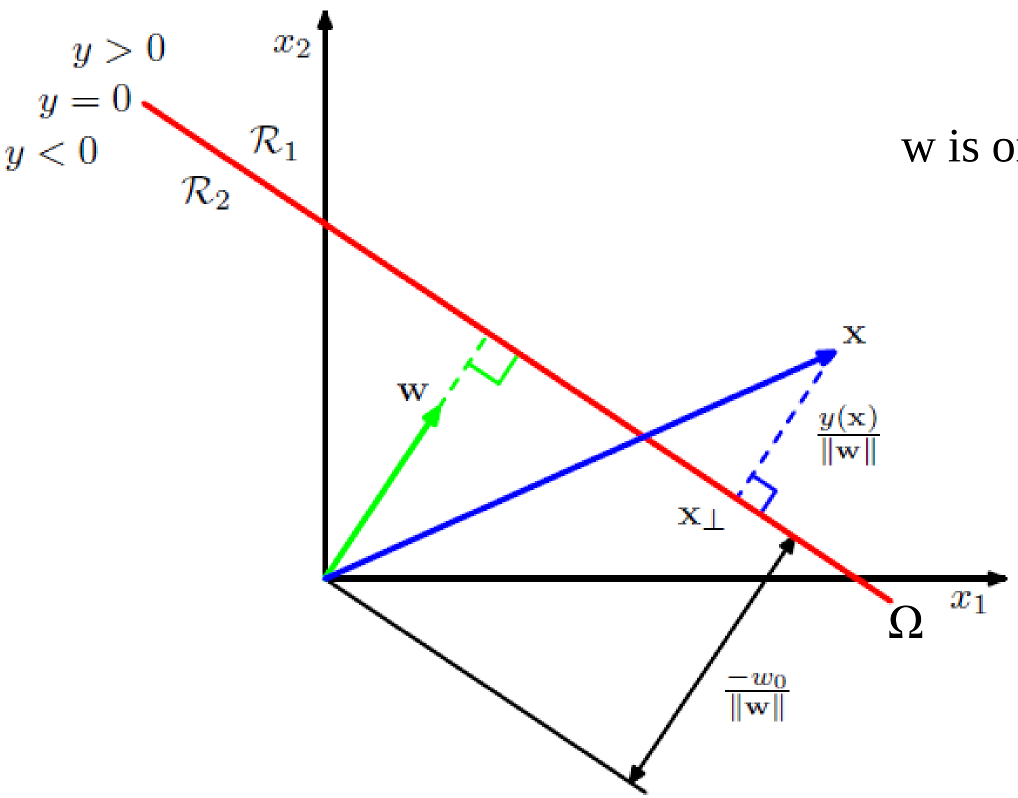
Three classification approaches

- Discriminant function:
  - *Least-squares approach*: making the model predictions as close as possible to a set of target values
  - *Fisher's linear discriminant*: maximum class separation in the output space
  - *The perceptron algorithm of Rosenblatt*: generalized linear model
- Generative approach:
  - Model the class-conditional densities and the class priors
  - Compute posterior probabilities through Bayes's theorem
- Discriminative approach:
  - Directly training posterior probabilities.

Discriminant Functions (nonprobabilistic methods)

Two classes

- Linear discriminant function: $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$
  - if $y(\mathbf{x}) \geq 0$, assign $\mathbf{x}$ to class C_1, else class C_2
  - decision surface Ω: $y(\mathbf{x}) = 0$
  - the normal distance from the origin to the decision surface: $\dfrac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\dfrac{w_0}{\|\mathbf{w}\|}$



x_A and x_B lie on the decision surface: $y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0$

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_A - \mathbf{x}_B) = 0$$

w is orthogonal to every vector lying within Ω

$\dfrac{\mathbf{w}}{\|\mathbf{w}\|}$ is the normal vector of Ω

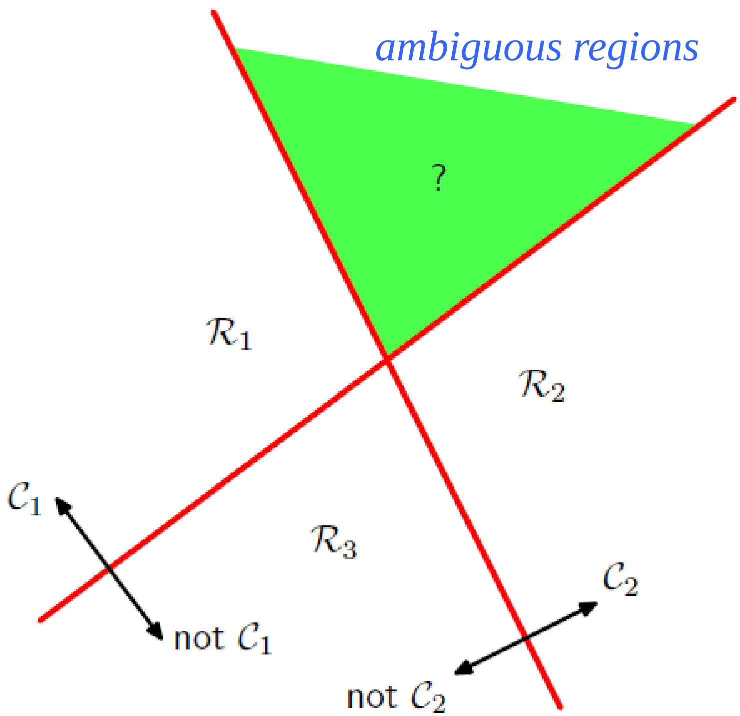
$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|} \Rightarrow r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$$

$$\widetilde{\mathbf{w}} = (w_0, \mathbf{w}) \qquad \widetilde{\mathbf{x}} = (x_0, \mathbf{x})$$

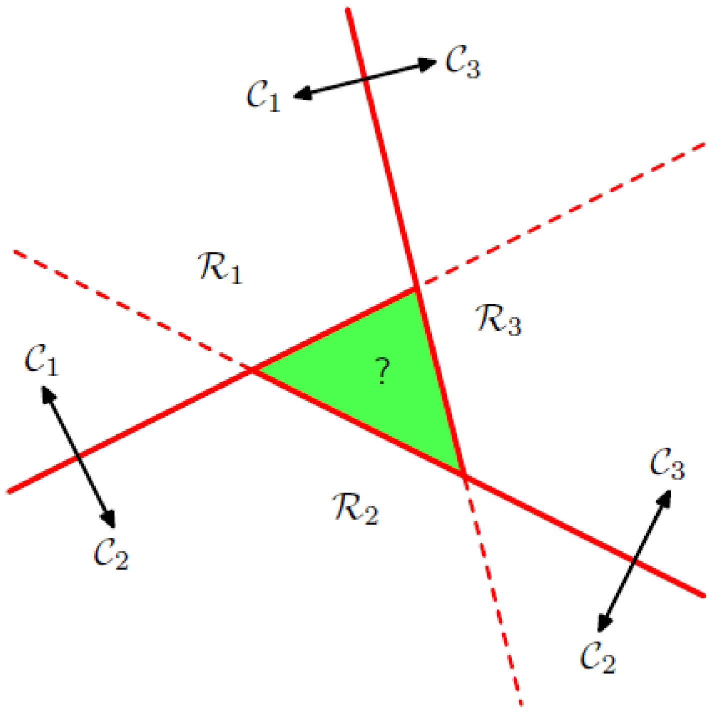
$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

Multiple classes

- How to build a K-class discriminant function?
  - *One-versus-the-rest classifier*
    - K-1 classifiers each of which solves a two-class problem
  - *One-versus-one classifier*
    - $K(K-1)/2$ binary discriminant functions



One-versus-the-rest classifier



One-versus-one classifier

Multiple classes

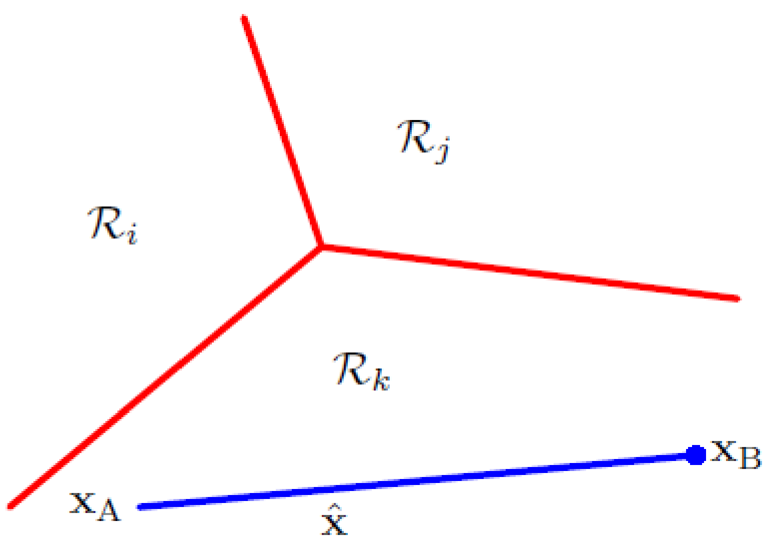
- Single K-class discriminant comprising K linear functions:

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

  - assigning a point $\mathbf{x}$ to class $\mathcal{C}_k$ if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$.
  - decision boundary between class $\mathcal{C}_k$ and class $\mathcal{C}_j$ is given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0$$

  - $\mathcal{R}_k$ is singly connected and convex.



$$\widehat{\mathbf{x}} = \lambda\mathbf{x}_A + (1-\lambda)\mathbf{x}_B \qquad \text{where } 0 \leqslant \lambda \leqslant 1$$

$$y_k(\widehat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1-\lambda) y_k(\mathbf{x}_B)$$

Because both $\mathbf{x}_A$ and $\mathbf{x}_B$ lie inside $\mathcal{R}_k$, it follows that $y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A)$ $y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$, for all $j \neq k$, and hence $y_k(\widehat{\mathbf{x}}) > y_j(\widehat{\mathbf{x}})$, and so $\widehat{\mathbf{x}}$ also lies inside $\mathcal{R}_k$.

Learning the parameters of LDF

- Three approaches:
  - *Least-squares approach*:
    - making the model predictions as close as possible to a set of target values
  - *Fisher's linear discriminant*:
    - maximum class separation in the output space
  - *The perceptron algorithm of Rosenblatt*:
    - generalized linear model

Least squares for classification

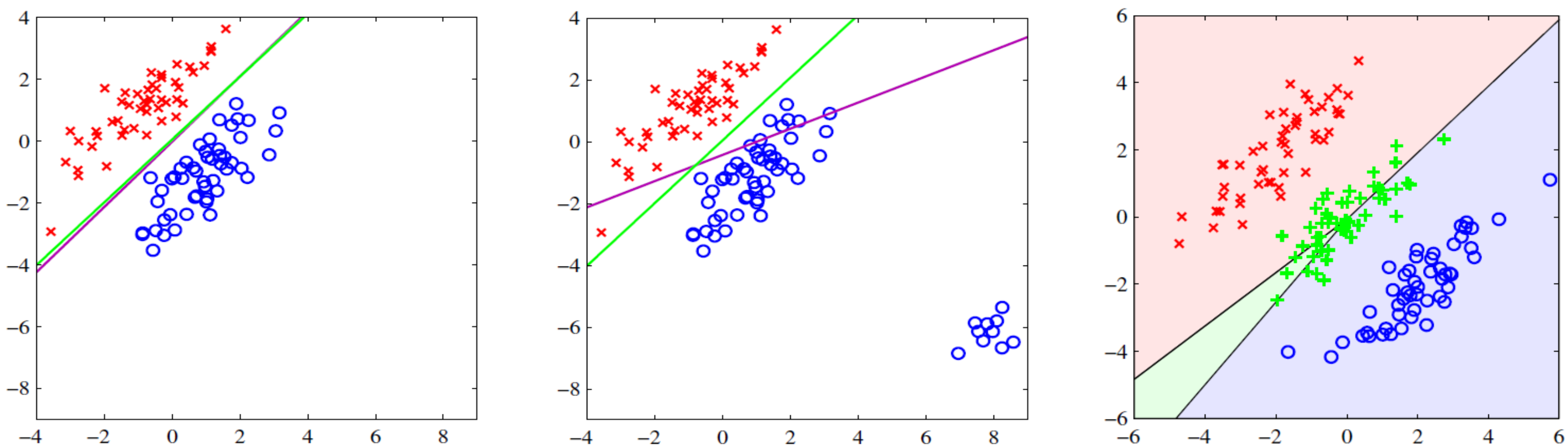
- Problem:
  - Each class $\mathcal{C}_k$ is described by its own linear model:

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \qquad \text{where } k = 1, \ldots, K$$

  - group together: $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} \qquad \widetilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^{\mathrm{T}})^{\mathrm{T}} \qquad \widetilde{\mathbf{x}} = (1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$
  - new input $\mathbf{x}$ is then assigned to the class for which the output $y_k = \widetilde{\mathbf{w}}_k^{\mathrm{T}}\widetilde{\mathbf{x}}$ is largest.
- Learning $\widetilde{\mathbf{W}}$ with training data set: $\{\mathbf{x}_n, \mathbf{t}_n\}$ where $n = 1, \ldots, N$
  - SSE function:

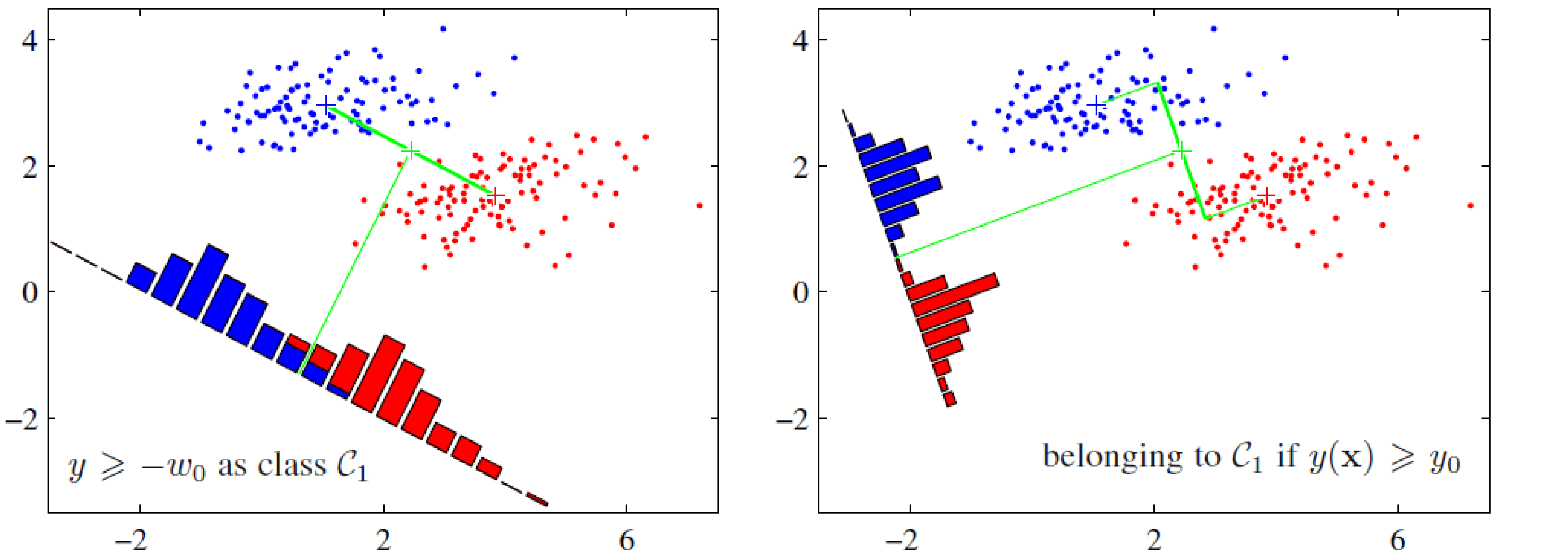
$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\} \Rightarrow \widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T}$$

- Discriminant function: $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}}\widetilde{\mathbf{x}}$



Fisher's linear discriminant

- From the view of dimensionality reduction:



- The simplest measure of the separation of the classes is the separation of the projected class means:

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} \mathbf{x}_n \quad \xrightarrow[m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k]{y = \mathbf{w}^{\mathrm{T}}\mathbf{x}} \quad m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1)$$

- Problem: we can increase the magnitude of w to make ($m_2 - m_1$) arbitrarily large!

$$\sum_i w_i^2 = 1 \quad \longrightarrow \quad \mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$$

Fisher's linear discriminant

- *The Fisher's criterion:* maximize the separation between the projected class means as well as the inverse of the total within-class variance.

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \qquad y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k$$

$$\Rightarrow \quad J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

Generalized Rayleigh quotient

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

Between-class covariance matrix

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}$$

Within-class covariance matrix

- *Fisher's linear discriminant:*

$$\nabla J(\mathbf{w}) = 0 \quad \Rightarrow \quad (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w} \quad \Rightarrow \quad \boxed{\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)}$$

Relation to least squares

- The Fisher criterion can be obtained as a special case of least squares if we consider following target coding scheme:
  - The target for class C_1 to be N/N_1, for class C_2 to be $-N/N_2$
  - The sum-of-squares error function:

$$E = \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n\right)^2$$

$$\frac{\partial E}{\partial w_0} = \sum_{n=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n\right) = 0$$

$$\sum_{n=1}^{N} t_n = N_1\frac{N}{N_1} - N_2\frac{N}{N_2} = 0$$

$$\Rightarrow \quad w_0 = -\mathbf{w}^{\mathrm{T}}\mathbf{m}$$

$$\mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n = \frac{1}{N}(N_1\mathbf{m}_1 + N_2\mathbf{m}_2)$$

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n\right)\mathbf{x}_n = 0$$

$$\Rightarrow \quad \left(\mathbf{S}_{\mathrm{W}} + \frac{N_1 N_2}{N}\mathbf{S}_{\mathrm{B}}\right)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2)$$

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

$$\Rightarrow \quad \boxed{\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)}$$

Fisher's discriminant for multiple classes

- Assume input space dimensionality $D > K$ (number of classes, $K>2$):

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \qquad y_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$$

covariance matrices defined in the original x-space

- Total covariance matrix: $\mathbf{S}_{\mathrm{T}} = \mathbf{S}_{\mathrm{W}} + \mathbf{S}_{\mathrm{B}}$

$$\mathbf{S}_{\mathrm{T}} = \sum_{n=1}^{N}(\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^{\mathrm{T}} \qquad \mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n = \frac{1}{N}\sum_{k=1}^{K}N_k\mathbf{m}_k \qquad N = \textstyle\sum_k N_k$$

- The generalization of the within-class covariance matrix:

$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K}\mathbf{S}_k \qquad \mathbf{S}_k = \sum_{n\in\mathcal{C}_k}(\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \qquad \mathbf{m}_k = \frac{1}{N_k}\sum_{n\in\mathcal{C}_k}\mathbf{x}_n$$

- The generalization of the between-class covariance matrix:

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K}N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^{\mathrm{T}}$$

Fisher's discriminant for multiple classes

- Assume input space dimensionality $D > K$ (number of classes, $K>2$):

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \qquad y_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$$

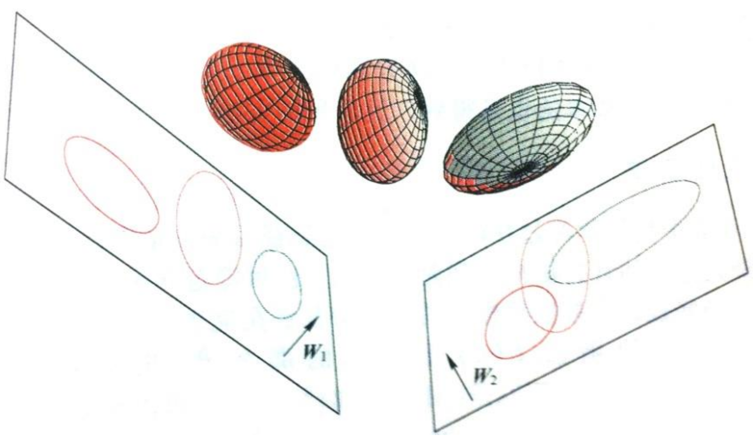
covariance matrices defined in the projected y-space

- The generalization of the within-class and between-class covariance matrix:

$$\mathrm{s}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \qquad \mathrm{s}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^{\mathrm{T}}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k$$

- *The Fisher's criterion:* $J(\mathbf{W}) = \mathrm{Tr}\left\{ \mathrm{s}_{\mathrm{W}}^{-1} \mathrm{s}_{\mathrm{B}} \right\} = \mathrm{Tr}\left\{ (\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{W}} \mathbf{W})^{-1} (\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{B}} \mathbf{W}) \right\}$



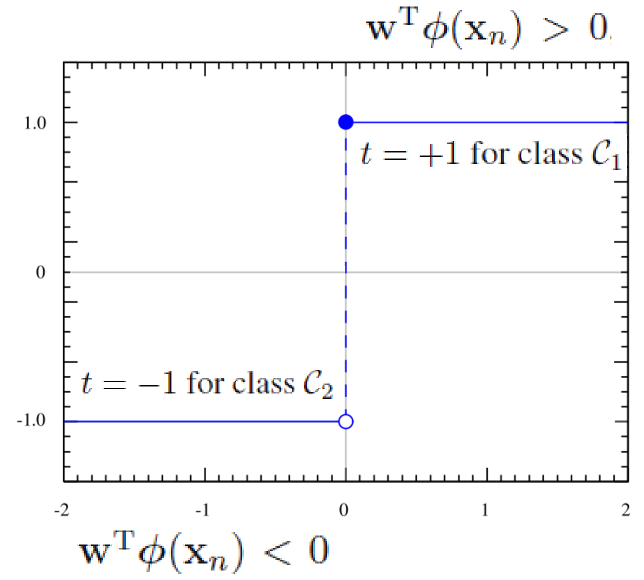
$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}}$$

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^{\mathrm{T}}$$

The perceptron algorithm

- Construct a generalized linear model:

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right) \qquad f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases}$$



- Perceptron criterion (need to be minimized):

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^{\mathrm{T}}\phi_n t_n$$

- Stochastic gradient descent algorithm:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta\phi_n t_n$$

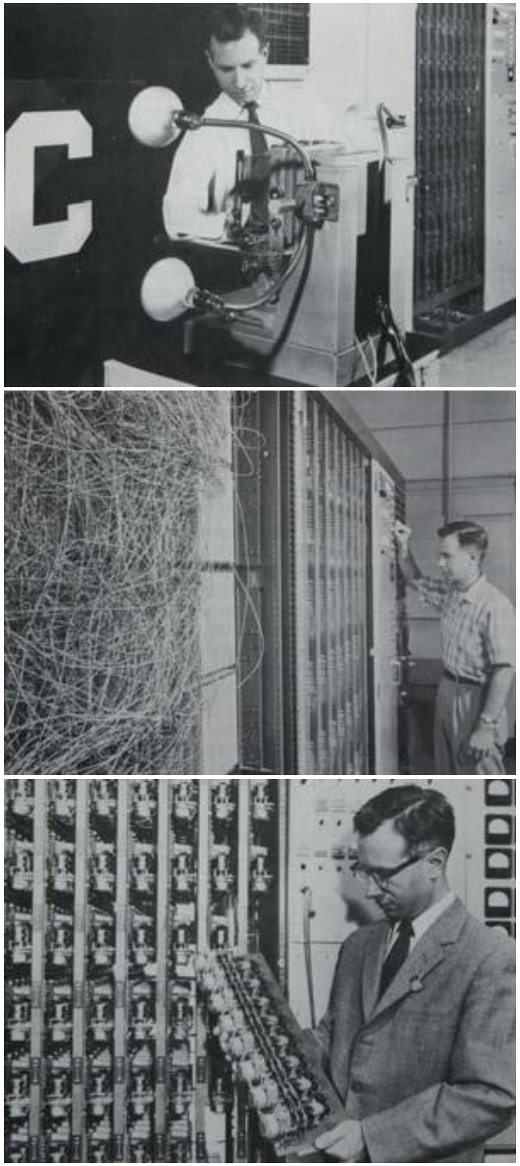
$\eta = 1$

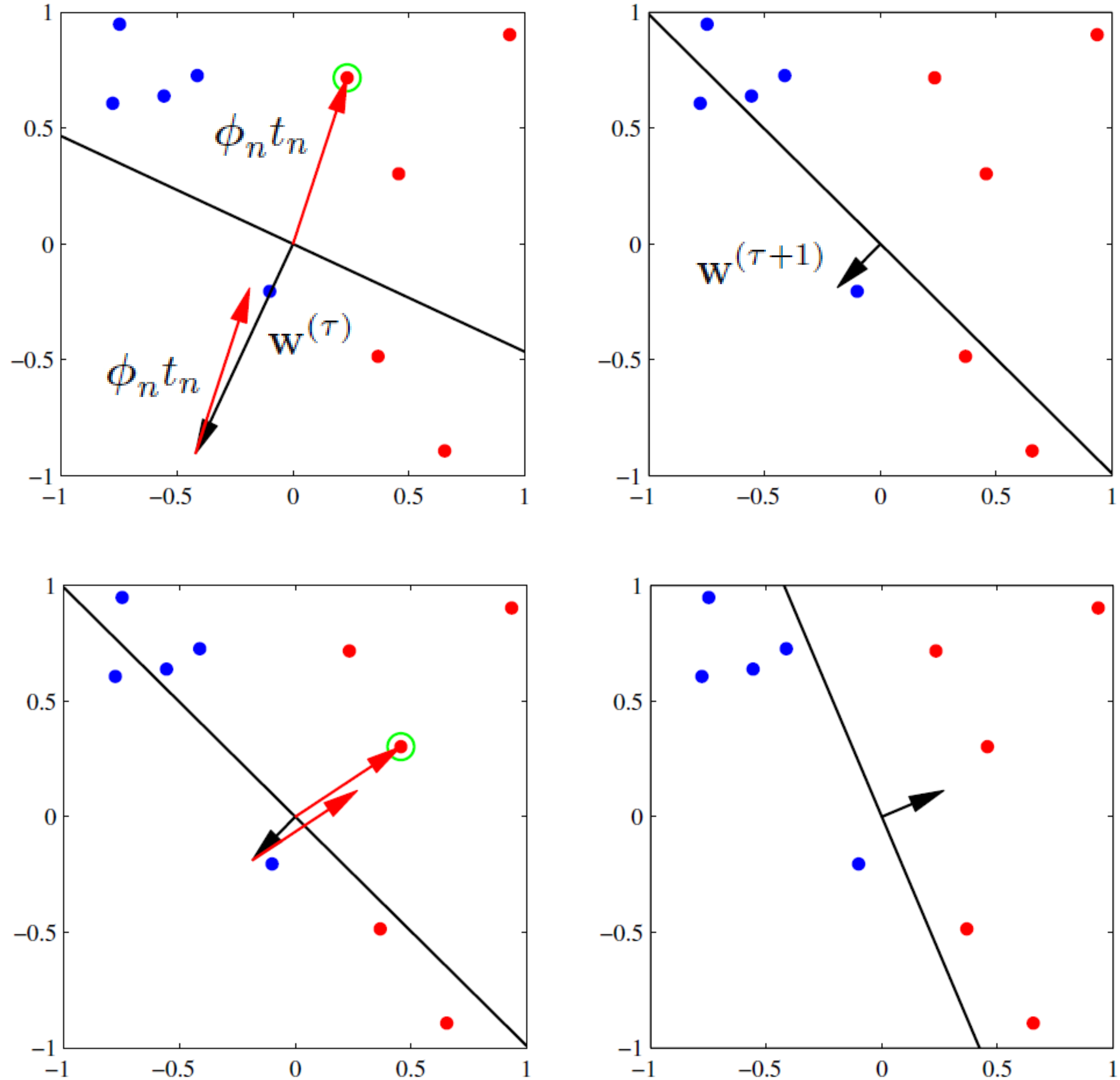
$$-\mathbf{w}^{(\tau+1)\mathrm{T}}\phi_n t_n = -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n - (\phi_n t_n)^{\mathrm{T}}\phi_n t_n < -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n$$

- Perceptron convergence theorem:
  - *If there exists an exact solution (in other words, if the training data set is linearly separable), then the perceptron learning algorithm is guaranteed to find an exact solution in a finite number of steps.*

The perceptron algorithm

- Analogue hardware implementations (Mark 1):





Probabilistic Generative Models

Probabilistic Generative Models

- Compute posterior probabilites by the class-conditional densities and class priors:

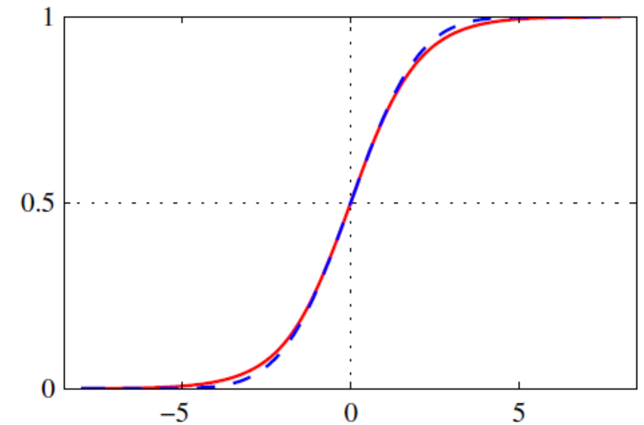
$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)+p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a)$$

logistic sigmoid function

$$a = \ln\frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \qquad \sigma(a) = \frac{1}{1+\exp(-a)}$$

$$\sigma(-a) = 1-\sigma(a) \qquad a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

logit function



- K>2 classes:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

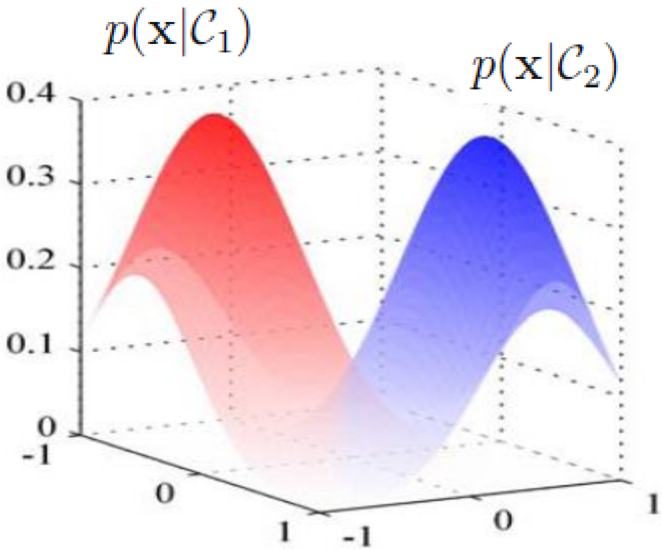
softmax function (normalized exponential)

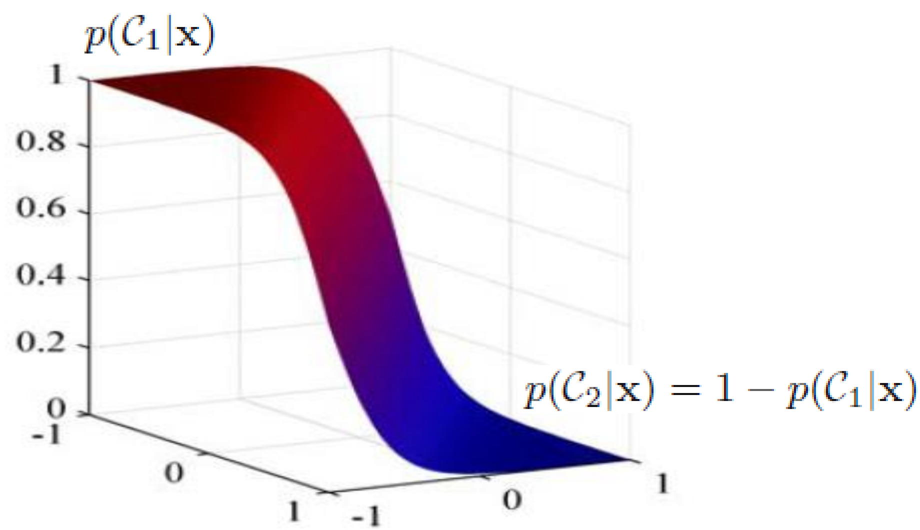
Continuous inputs

- *Assume:* $$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- *2 classes:* $$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \qquad \mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$





- *K classes:*

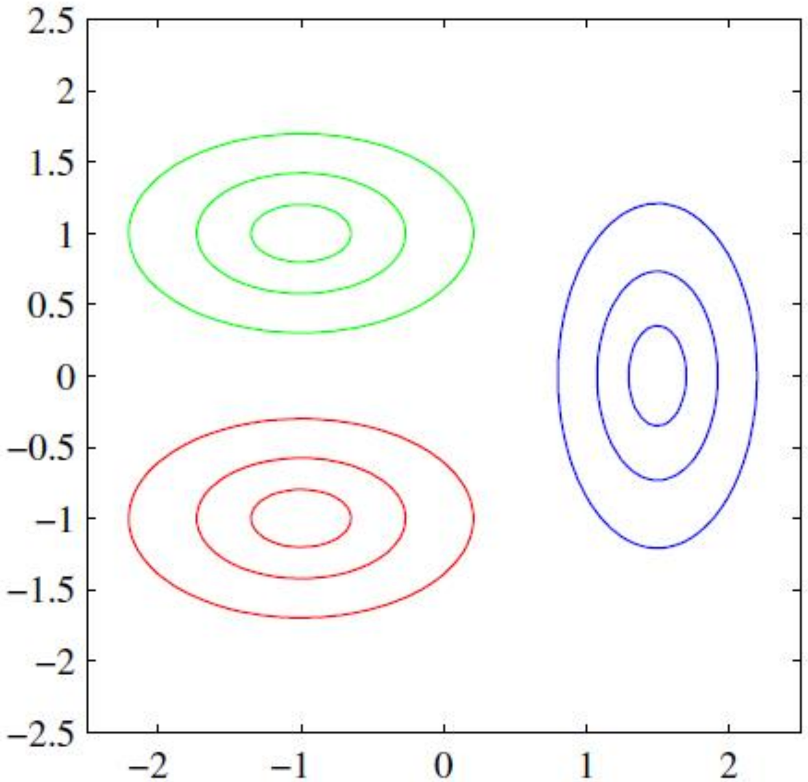
$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \qquad \mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \qquad w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)$$

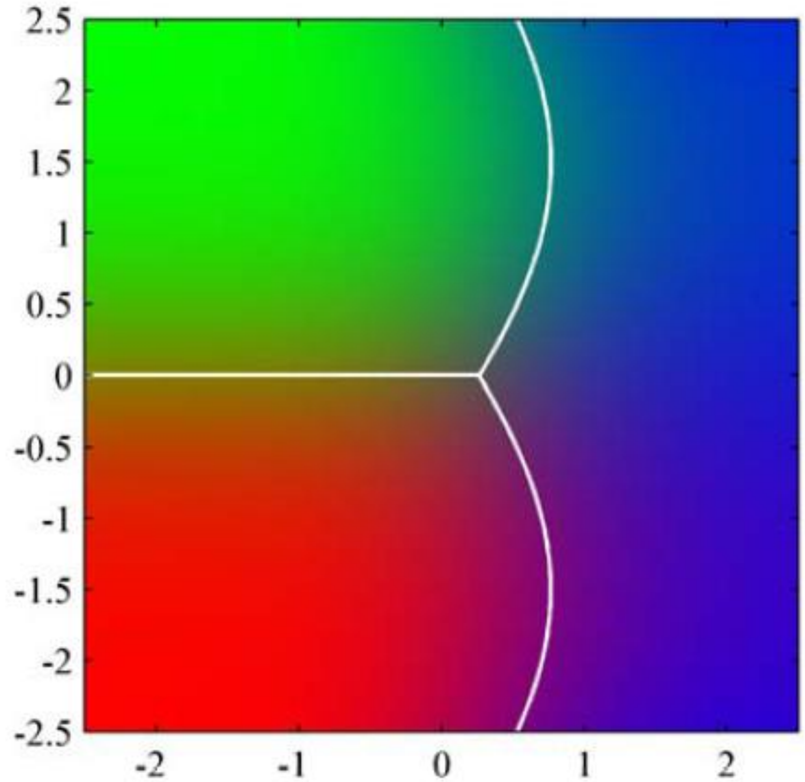
Continuous inputs

- *Assume:* $$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- *K classes with its own covariance matrix (quadratic discriminant):*

$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \qquad \mathbf{w}_k = \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k \qquad w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)$$





Maximum likelihood solution for two classes

- *We have assumed (shared covariance matrix):*

$$p(\mathbf{x}_n|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \Rightarrow p(\mathbf{x}_n|\mathcal{C}_1) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \quad p(\mathbf{x}_n|\mathcal{C}_2) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

- *And denote the prior:* $p(\mathcal{C}_1) = \pi, \quad p(\mathcal{C}_2) = 1 - \pi$
- *Hence:*

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1 - \pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

- *Now we have an input data set:*

$\{\mathbf{x}_n, t_n\}$ where $n = 1, \ldots, N$. $t_n = 1$ denotes class $\mathcal{C}_1$ and $t_n = 0$ denotes class $\mathcal{C}_2$

- *Then we estimate the parameters of above model by ML.*
- *The likelihood function:*

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n} \qquad \mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$$

- *The log likelihood:*

$$\ln p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \{t_n \ln \pi + (1 - t_n)\ln(1 - \pi) + t_n \ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1 - t_n)\ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\}$$

Maximum likelihood solution for two classes

$$\ln p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \{t_n \ln \pi + (1-t_n)\ln(1-\pi) + t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-t_n)\ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\}$$

- *Solve π:*

$$\sum_{n=1}^{N} \{t_n \ln \pi + (1-t_n)\ln(1-\pi)\} \Rightarrow \pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1+N_2}$$

- *Solve $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$:*

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N} t_n(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{const}$$

$$\Rightarrow \boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n \qquad \boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N}(1-t_n)\mathbf{x}_n$$

- *Solve $\boldsymbol{\Sigma}$:*

$$-\frac{1}{2}\sum_{n=1}^{N} t_n \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} t_n(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1)$$

$$-\frac{1}{2}\sum_{n=1}^{N}(1-t_n)\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(1-t_n)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_2) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\mathbf{S}\right\}$$

$$\mathbf{S} = \frac{N_1}{N}\overbrace{\boxed{\frac{1}{N_1}\sum_{n\in\mathcal{C}_1}(\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}}}^{S_1} + \frac{N_2}{N}\overbrace{\boxed{\frac{1}{N_2}\sum_{n\in\mathcal{C}_2}(\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}}}}^{S_2} \Rightarrow \boldsymbol{\Sigma} = \mathbf{S}$$

Maximum likelihood solution for K-classes

- *The likelihood function:* $p(\{\phi_n, \mathrm{t}_n\}|\{\pi_k\}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\{p(\phi_n|\mathcal{C}_k)\pi_k\}^{t_{nk}}$ $\quad \sum_k \pi_k = 1$
- *The log likelihood:* $\ln p(\{\phi_n, \mathrm{t}_n\}|\{\pi_k\}) = \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\{\ln p(\phi_n|\mathcal{C}_k) + \ln \pi_k\}$
- *Introduce a Lagrange multiplier λ:* $\ln p(\{\phi_n, \mathrm{t}_n\}|\{\pi_k\}) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$

- *Solve π_k:* $\pi_k = \frac{N_k}{N}$

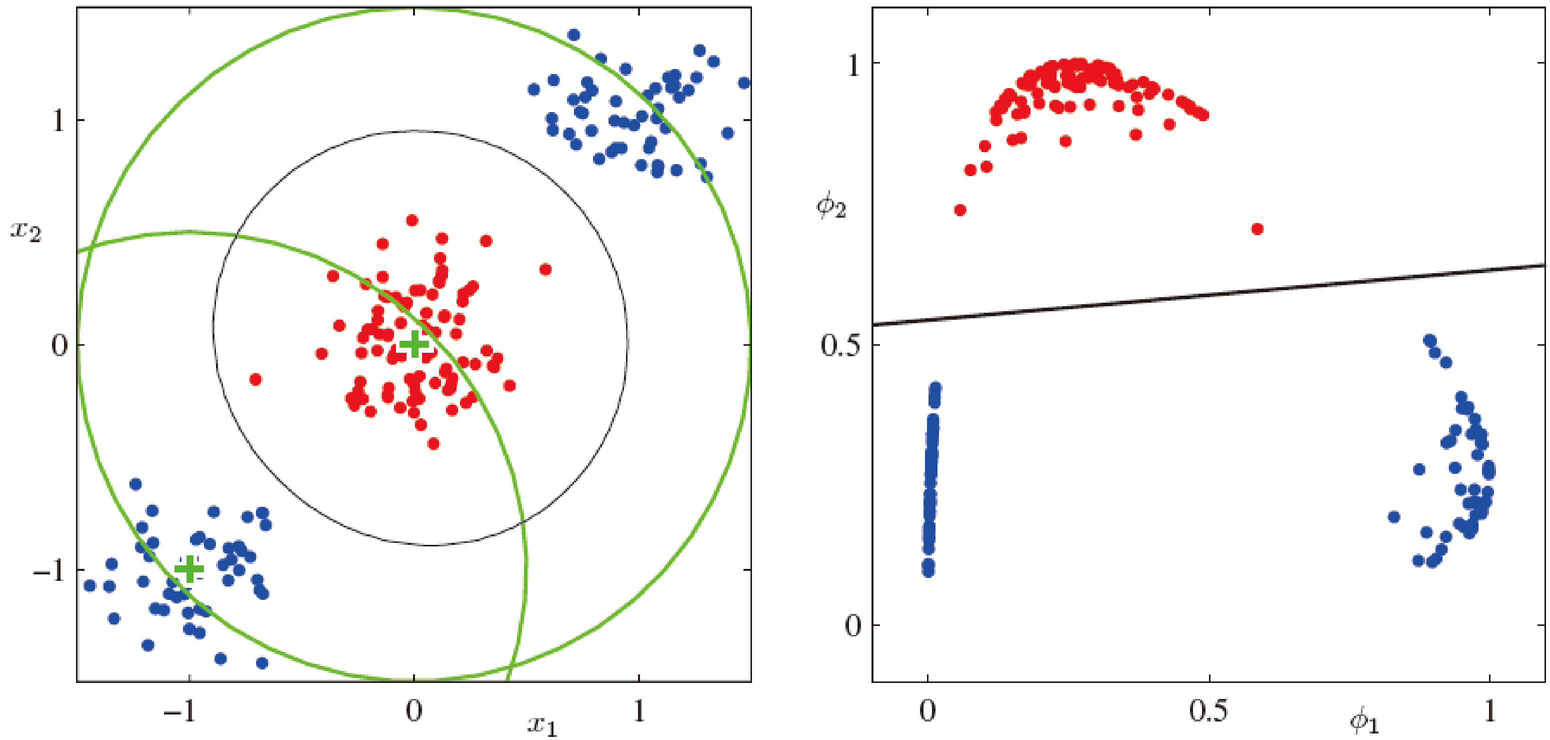
Assumption: Each class-conditional density is Gaussian with a shared covariance matrix.

- *Solve $\boldsymbol{\mu}_k$:* $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} t_{nk}\phi_n$
- *Solve $\boldsymbol{\Sigma}$:* $\boldsymbol{\Sigma} = \sum_{k=1}^{K}\frac{N_k}{N}\mathbf{S}_k$ $\qquad \mathbf{S}_k = \frac{1}{N_k}\sum_{n=1}^{N} t_{nk}(\phi_n - \boldsymbol{\mu}_k)(\phi_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$

Probabilistic Discriminative Models

Fixed basis functions

- Classification models work on feature space instead of original input space by nonlinear basis functions:



Logistic regression

- Logistic regression model:
  - Only M parameters need to be estimated.

logistic sigmoid function

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right) \qquad p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi) \qquad \sigma(a) = \frac{1}{1+\exp(-a)}$$

  - For a data set $\{\phi_n, t_n\}$, where $t_n \in \{0,1\}$ and $\phi_n = \phi(\mathbf{x}_n)$, $n = 1, \ldots, N$, the likelihood function can be written

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}\left\{1 - y_n\right\}^{1-t_n}$$ where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $y_n = p(\mathcal{C}_1|\phi_n)$.

  - Cross-entropy error function:

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\left\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\right\}$$

$$y_n = \sigma(a_n)$$

$$a_n = \mathbf{w}^{\mathrm{T}}\phi_n$$

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

$$\left\{\begin{aligned} \frac{\partial E}{\partial y_n} &= \frac{1-t_n}{1-y_n} - \frac{t_n}{y_n} = \frac{y_n(1-t_n) - t_n(1-y_n)}{y_n(1-y_n)} = \frac{y_n - t_n}{y_n(1-y_n)} \\ \frac{\partial y_n}{\partial a_n} &= \frac{\partial \sigma(a_n)}{\partial a_n} = \sigma(a_n)\left(1-\sigma(a_n)\right) = y_n(1-y_n) \end{aligned}\right.$$

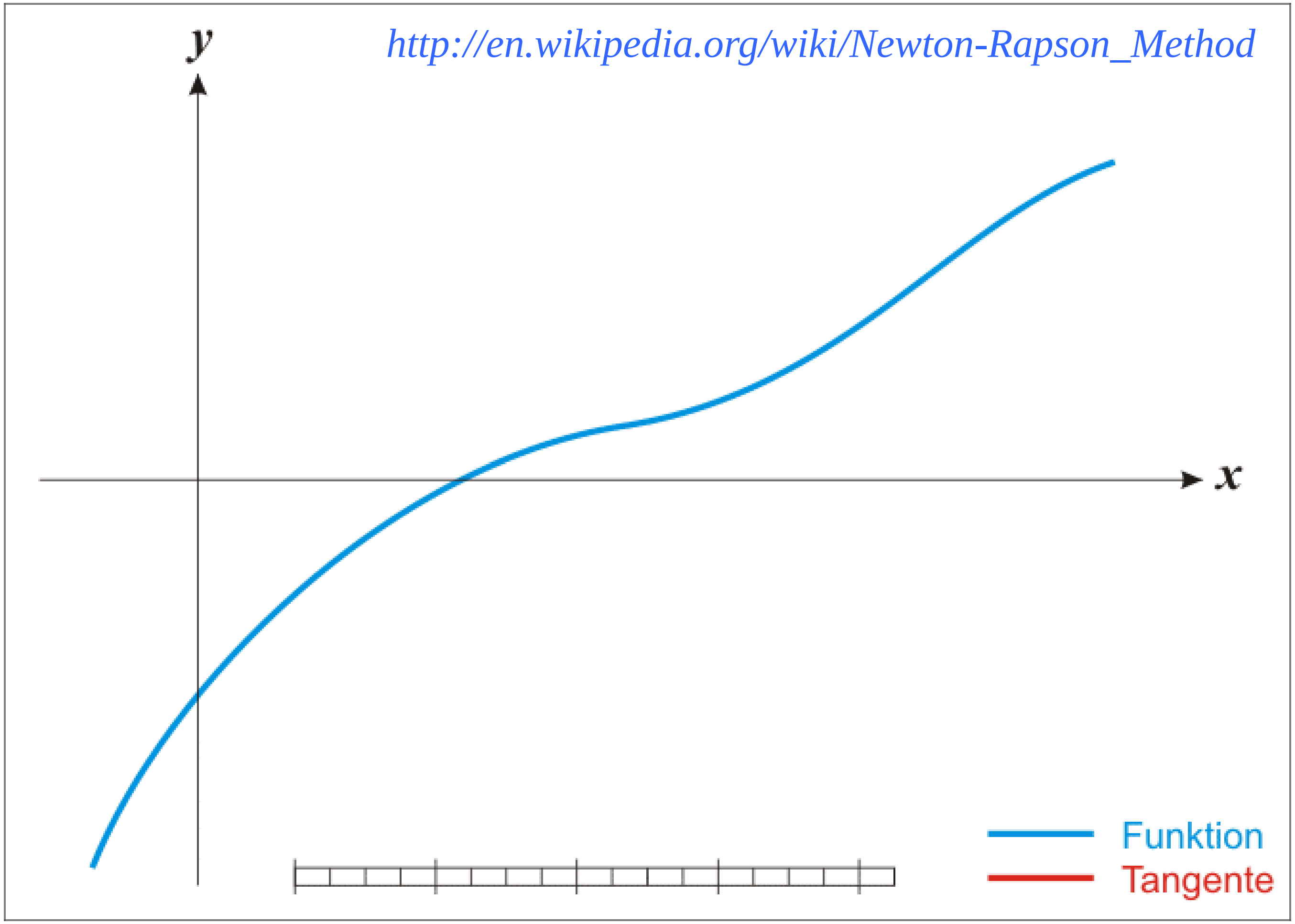
No closed-form solution

$$\boxed{\nabla E(\mathbf{w}) = \sum_{n=1}^{N}\frac{\partial E}{\partial y_n}\frac{\partial y_n}{\partial a_n}\nabla a_n = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \sum_{n=1}^{N}\left(\sigma(\mathbf{w}^{\mathrm{T}}\phi_n) - t_n\right)\phi_n}$$

Iterative reweighted least squares (IRLS) algorithm

- *Newton-Raphson* iterative optimization scheme:

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$$



Iterative reweighted least squares (IRLS) algorithm

- *Newton-Raphson* iterative optimization scheme: $\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$

- For linear regression model with the sum-of-squares error function:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\phi_n - t_n)\phi_n = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

$$\boxed{\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N}\phi_n\phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}}$$

$$\Rightarrow \mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\left\{\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\right\} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

- For logistic regression model with cross-entropy error function:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \mathbf{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t})$$

$$\boxed{\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N}y_n(1-y_n)\phi_n\phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi}}$$

$$\Rightarrow \mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t}) = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\left\{\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t})\right\}$$
$$= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{z}$$

$$\mathbf{z} = \mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$$

$$\boxed{\begin{aligned} a_n(\mathbf{w}) &\simeq a_n(\mathbf{w}^{(\text{old})}) + \left.\frac{\mathrm{d}a_n}{\mathrm{d}y_n}\right|_{\mathbf{w}^{(\text{old})}}(t_n - y_n) \\ &= \phi_n^{\mathrm{T}}\mathbf{w}^{(\text{old})} - \frac{(y_n - t_n)}{y_n(1-y_n)} = z_n. \end{aligned}}$$

local linear approximation to the logistic sigmoid function around the current operating point w(old)

Multiclass logistic regression

- Models:

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right) \Longrightarrow p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \mathbf{w}_k^{\mathrm{T}}\phi$$

- Likelihood function (1-of-K coding scheme):

$$p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}} \qquad y_{nk} = y_k(\phi_n)$$

- Cross-entropy error function:

$$E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_{nk}$$

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$

$$\Longrightarrow \nabla_{\mathbf{w}_j} E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = \sum_{n=1}^{N}(y_{nj} - t_{nj})\,\phi_n \qquad \sum_k t_{nk} = 1$$

Exercise 4.18

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_n \Longrightarrow \mathbf{w}_1,\ldots,\mathbf{w}_K$$

Sequential learning algorithm to update w one by one

- Solve by IRLS algorithm:

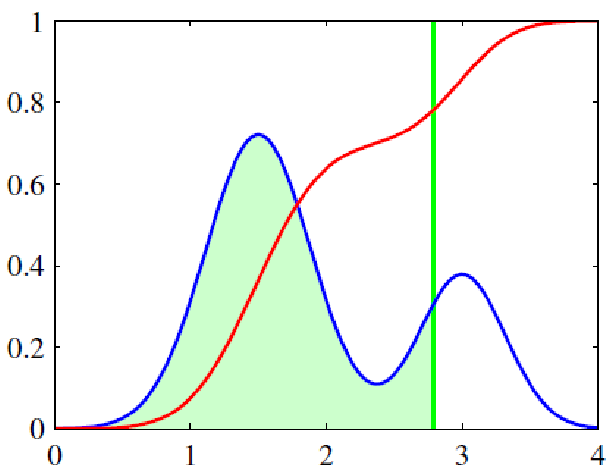
$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$$

*Block j,k of Hessian matrix (comprise blocks of size M*M)*

$$\boxed{\nabla_{\mathbf{w}_k}\nabla_{\mathbf{w}_j} E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = \sum_{n=1}^{N} y_{nk}(I_{kj} - y_{nj})\phi_n\phi_n^{\mathrm{T}}}$$

Another PDM: Probit regression

- Use CDF to construct an activation function:

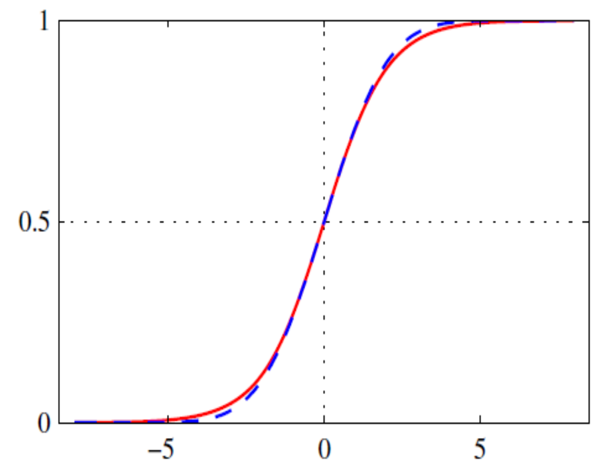


Logistic sigmoid function

$$p(\mathcal{C}_1|\phi) = \frac{1}{1+\exp(\mathbf{w}^{\mathrm{T}}\phi)}$$

$$p(\mathcal{C}_1|\phi) = \int_{-\infty}^{\mathbf{w}^{\mathrm{T}}\phi} \mathcal{N}(\theta|0,1)\,\mathrm{d}\theta$$

Inverse probit function



Canonical link functions

- Canonical link function:

$$p(t|\eta, s) = \frac{1}{s} h\left(\frac{t}{s}\right) g(\eta) \exp\left\{\frac{\eta t}{s}\right\} \qquad y \equiv \mathbb{E}[t|\eta] = -s\frac{d}{d\eta}\ln g(\eta)$$

- Generalized linear model: $y = f(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi})$
  - f( ) is the activation function and f^{-1}( ) is the link function
- Log likelihood function:

$$\ln p(\mathbf{t}|\eta, s) = \sum_{n=1}^{N} \ln p(t_n|\eta, s) = \sum_{n=1}^{N} \left\{\ln g(\eta_n) + \frac{\eta_n t_n}{s}\right\} + \text{const}$$

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\eta, s) = \sum_{n=1}^{N} \left\{\frac{d}{d\eta_n}\ln g(\eta_n) + \frac{t_n}{s}\right\} \frac{d\eta_n}{dy_n}\frac{dy_n}{da_n}\nabla a_n = \sum_{n=1}^{N} \frac{1}{s}\{t_n - y_n\}\,\psi'(y_n) f'(a_n)\phi_n$$

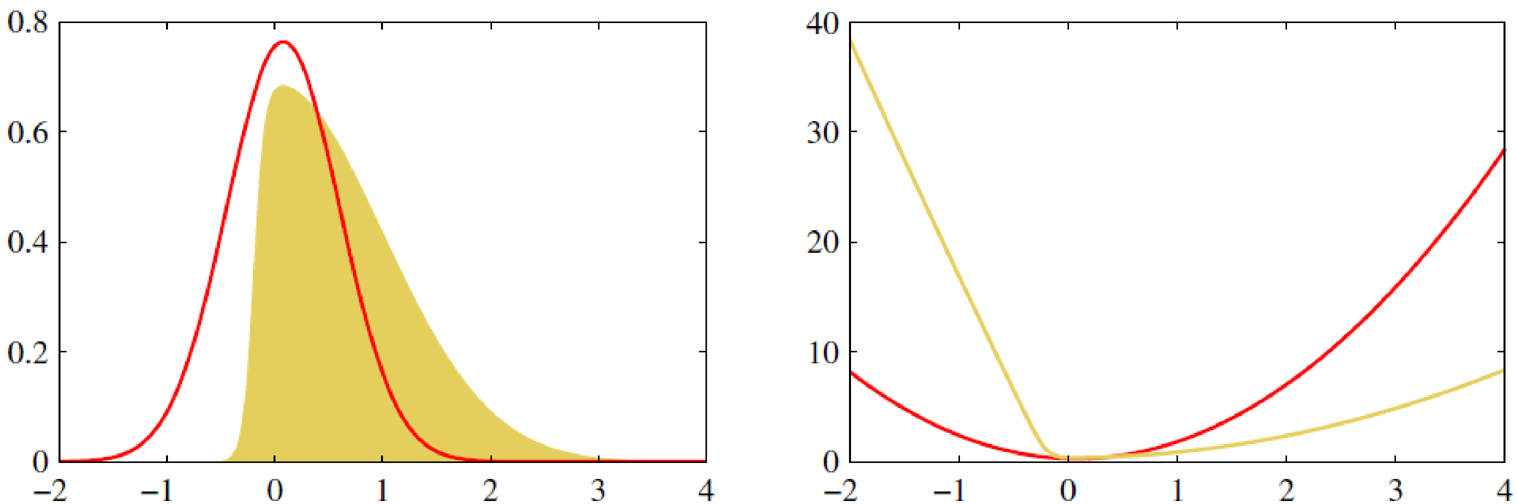
$$\xrightarrow[f^{-1}(y) = \psi(y)]{} \quad \nabla E(\mathbf{w}) = \frac{1}{s}\sum_{n=1}^{N}\{y_n - t_n\}\phi_n$$

For the Gaussian $s = \beta^{-1}$,
for the logistic model $s = 1$.

The Laplace Approximation

The Laplace Approximation

- Find a Gaussian approximation q(z) which is centred on a mode of the distribution p(z):



$$p(z) = \frac{1}{Z} f(z) \qquad \ln f(z) \simeq \ln f(z_0) - \frac{1}{2} A (z - z_0)^2 \qquad f(z) \simeq f(z_0) \exp\left\{ -\frac{A}{2} (z - z_0)^2 \right\}$$

$$Z = \int f(z) \, \mathrm{d}z \qquad A = - \frac{d^2}{dz^2} \ln f(z) \bigg|_{z = z_0} \qquad q(z) = \left(\frac{A}{2\pi} \right)^{1/2} \exp\left\{ -\frac{A}{2} (z - z_0)^2 \right\}$$

The Laplace Approximation

$$p(z) = \frac{1}{Z} f(z) \qquad f(z) \simeq f(z_0) \exp\left\{-\frac{A}{2}(z - z_0)^2\right\} \qquad Z = \int f(z)\,\mathrm{d}z$$

$$q(z) = \left(\frac{A}{2\pi}\right)^{1/2} \exp\left\{-\frac{A}{2}(z - z_0)^2\right\} \qquad A = -\left.\frac{d^2}{dz^2} \ln f(z)\right|_{z=z_0}$$

- Extend to M-dimensional space: $p(\mathbf{z}) = f(\mathbf{z})/Z$

$$Z = \int f(\mathbf{z})\,\mathrm{d}\mathbf{z} \simeq f(\mathbf{z}_0) \int \exp\left\{-\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^{\mathrm{T}} \mathbf{A} (\mathbf{z} - \mathbf{z}_0)\right\} \mathrm{d}\mathbf{z} = f(\mathbf{z}_0) \frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}}$$

$$\ln f(\mathbf{z}) \simeq \ln f(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^{\mathrm{T}} \mathbf{A} (\mathbf{z} - \mathbf{z}_0)$$

$$f(\mathbf{z}) \simeq f(\mathbf{z}_0) \exp\left\{-\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^{\mathrm{T}} \mathbf{A} (\mathbf{z} - \mathbf{z}_0)\right\} \qquad \mathbf{A} = -\nabla\nabla \ln f(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_0}$$

$$\Rightarrow \quad q(\mathbf{z}) = \frac{|\mathbf{A}|^{1/2}}{(2\pi)^{M/2}} \exp\left\{-\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^{\mathrm{T}} \mathbf{A} (\mathbf{z} - \mathbf{z}_0)\right\} = \mathcal{N}(\mathbf{z}|\mathbf{z}_0, \mathbf{A}^{-1})$$

Bayesian Logistic Regression

Bayesian Logistic Regression

- *In logistic regression model, we have:*
  - For a data set $\{\phi_n, t_n\}$, where $t_n \in \{0,1\}$ and $\phi_n = \phi(\mathbf{x}_n)$, $n = 1, \ldots, N$, the likelihood function can be written
  $$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n}$$ where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $y_n = p(\mathcal{C}_1|\phi_n)$.
  - *Now assume prior is Gaussian:* $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$
  - *Then posterior distribution over w (Obviously it's not a Gaussian, but we can find its Gaussian approximation by Laplace approximation framework):*
  $$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{w})p(\mathbf{t}|\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0) \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n}$$
  $$y_n = \sigma(a_n)$$
  $$a_n = \mathbf{w}^{\mathrm{T}}\phi_n$$
  $$\Rightarrow \ln p(\mathbf{w}|\mathbf{t}) = -\frac{1}{2}(\mathbf{w} - \mathbf{m}_0)^{\mathrm{T}}\mathbf{S}_0^{-1}(\mathbf{w} - \mathbf{m}_0) + \sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\} + \mathrm{const}$$
  - *Maximize above posterior distribution to give the MAP solution* w_{MAP}*, which defines the mean of the Gaussian.*
  - *The covariance is then give by the inverse of the matrix of second derivatives of the negative log likelihood:*
  $$\mathbf{S}_N = -\nabla\nabla \ln p(\mathbf{w}|\mathbf{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^{\mathrm{T}}$$
- *So, the Gaussian approximation:*
$$\boxed{q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_{\mathrm{MAP}}, \mathbf{S}_N^{-1})}$$

Bayesian Logistic Regression

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_{\mathrm{MAP}}, \mathbf{S}_N^{-1})$$

- *For new input vector x, corresponding feature vector is φ(x)*
- *Then the predictive distribution for class C_1:*

$$p(\mathcal{C}_1|\phi, \mathbf{t}) = \int p(\mathcal{C}_1|\phi, \mathbf{w})p(\mathbf{w}|\mathbf{t})\,\mathrm{d}\mathbf{w} \simeq \int \sigma(\mathbf{w}^{\mathrm{T}}\phi)q(\mathbf{w})\,\mathrm{d}\mathbf{w} \qquad p(\mathcal{C}_2|\phi, \mathbf{t}) = 1 - p(\mathcal{C}_1|\phi, \mathbf{t})$$

- *We can rewrite (Mathematics structural trick):*

$$\sigma(\mathbf{w}^{\mathrm{T}}\phi) = \int \delta(a - \mathbf{w}^{\mathrm{T}}\phi)\sigma(a)\,\mathrm{d}a \qquad \delta(\cdot) \text{ is the Dirac delta function}$$

$$\Rightarrow \int \sigma(\mathbf{w}^{\mathrm{T}}\phi)q(\mathbf{w})\,\mathrm{d}\mathbf{w} = \int \sigma(a)p(a)\,\mathrm{d}a \quad \textit{where} \quad p(a) = \int \delta(a - \mathbf{w}^{\mathrm{T}}\phi)q(\mathbf{w})\,\mathrm{d}\mathbf{w}$$

$$\mu_a = \mathbb{E}[a] = \int p(a)a\,\mathrm{d}a = \int q(\mathbf{w})\mathbf{w}^{\mathrm{T}}\phi\,\mathrm{d}\mathbf{w} = \mathbf{w}_{\mathrm{MAP}}^{\mathrm{T}}\phi$$

$$\sigma_a^2 = \mathrm{var}[a] = \int p(a)\left\{a^2 - \mathbb{E}[a]^2\right\}\mathrm{d}a = \int q(\mathbf{w})\left\{(\mathbf{w}^{\mathrm{T}}\phi)^2 - (\mathbf{m}_N^{\mathrm{T}}\phi)^2\right\}\mathrm{d}\mathbf{w} = \phi^{\mathrm{T}}\mathbf{S}_N^{-1}\phi$$

$$\Rightarrow p(\mathcal{C}_1|\mathbf{t}) = \int \sigma(a)p(a)\,\mathrm{d}a = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\,\mathrm{d}a$$

apply the approximation $\sigma(a) \simeq \Phi(\lambda a)$ $\qquad \lambda^2 = \pi/8 \qquad \kappa(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2}$

$$\Rightarrow p(\mathcal{C}_1|\mathbf{t}) \simeq \int \Phi(\lambda a)\mathcal{N}(a|\mu, \sigma^2)\,\mathrm{d}a = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{1/2}}\right) \simeq \sigma\left(\kappa(\sigma^2)\mu\right) = \sigma\left(\kappa(\sigma_a^2)\mu_a\right)$$

Next: Kernel Methods and SVM

- HW4:
  - 4.4, 4.5, 4.8, 4.17, 4.18, 4.19
  - Programming work: see website for details.