Home Page

Title Page

Contents

Page 11 of 30

Go Back

Full Screen

Close

Quit

Moore-Penrose Pseudo-Inverse

- $\mathbf{Bx} = \mathbf{c}$, $\mathbf{B}$: non-square. $\mathbf{x} = \mathbf{B}^{-1}\mathbf{c}$?
- $\mathbf{B}^T\mathbf{Bx} = \mathbf{B}^T\mathbf{c}$: $\boxed{\mathbf{x} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{c}}$
  Moore-Penrose pseudo-inverse, do not compute inverse: the SVD does it algorithmically
- Another derivation: $f(\mathbf{x}) = ||\mathbf{Bx} - \mathbf{c}||^2$, $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = ?$
- Method 1: $f(\mathbf{x}) = ||\mathbf{Bx} - \mathbf{c}||^2 = (\mathbf{Bx} - \mathbf{c})^T(\mathbf{Bx} - \mathbf{c})$,
  $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{B}^T(\mathbf{Bx} - \mathbf{c}) = \mathbf{0}$ (vector calculus rules)
  $2\mathbf{B}^T\mathbf{Bx} = 2\mathbf{B}^T\mathbf{c}$, get the pseudo-inverse above
- Method 2: $f(\mathbf{x}) = (\mathbf{Bx} - \mathbf{c})^T(\mathbf{Bx} - \mathbf{c})$
  $= \mathbf{x}^T\mathbf{B}^T\mathbf{Bx} - \mathbf{x}^T\mathbf{B}^T\mathbf{c} - \mathbf{c}^T\mathbf{Bx} + \mathbf{c}^T\mathbf{c}$. Middle terms scalars, equal! $(\mathbf{x}^T\mathbf{B}^T\mathbf{c})^T = \mathbf{c}^T\mathbf{Bx}$. $f(\mathbf{x}) = \mathbf{x}^T\mathbf{B}^T\mathbf{Bx} - 2\mathbf{c}^T\mathbf{Bx} + \mathbf{c}^T\mathbf{c}$. Differentiate!

Home Page

Title Page

Contents

Page 12 of 30

Go Back

Full Screen

Close

Quit

Aside: Specific role of the bias w_0

- log-lh$=-\frac{N}{2}\log 2\pi - \frac{N}{2}\log\sigma^2 - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}\{t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)\}^2$
- 3rd term has $\mathbf{w}$: $-\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}\{t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)\}^2 = -\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}\{t_i - (w_0\phi_0(\mathbf{x}_i) + \ldots + w_{M-1}\phi_{M-1}(\mathbf{x}_i))\}^2 = -\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}\{t_i - w_0\phi_0(\mathbf{x}_i) - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_i)\}^2$
- $\frac{\partial \text{log-lh}}{\partial w_0} = \frac{1}{\sigma^2}\frac{2}{2}\sum_{i=1}^{N}\{t_i - w_0\phi_0(\mathbf{x}_i) - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_i)\} = 0 \implies \sum_{i=1}^{N} t_i - w_0 N - \sum_{i=1}^{N}\sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_i) = 0$
- Interchange the two summations: $w_{0_{ML}} = \frac{\sum_{i=1}^{N} t_i}{N} - \sum_{j=1}^{M-1} w_j \frac{\sum_{i=1}^{N}\phi_j(\mathbf{x}_i)}{N} \implies w_{0_{ML}} = \bar{t} - \sum_{j=1}^{M-1} w_j\overline{\phi_j}$
- The bias compensates for the difference between the (av target) & the weighted sum of av basis fns

Home Page

Title Page

Contents

Page 13 of 30

Go Back

Full Screen

Close

Quit

Regularised Least Squares

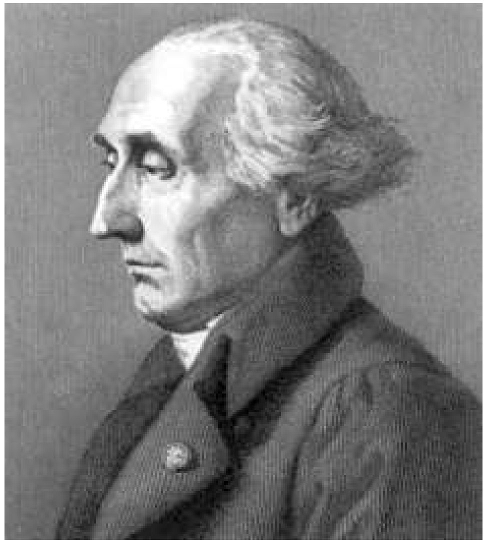
- Why? An otherwise nice model with nice properties, but gives infinite/trivial solutions
- To control overfitting
- Start: $E_D(\mathbf{w}) \stackrel{\triangle}{=} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2$ (data fidelity)
- To minimise $E_D(\mathbf{w}) + \lambda E_w(\mathbf{w})$ Fidelity, weights param $\lambda : E_w(\mathbf{w}) = 0$. $E_w(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}\sum_{j=0}^{M-1} w_j^2$
- Advantage: Quadratic in $\mathbf{w}$: closed-form solution
- (ML): 'weight decay': weights $\downarrow$ 0 unless supported by the data. (Stat): 'param shrinkage'
- $E \stackrel{\triangle}{=} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2 + \lambda \frac{1}{2} \mathbf{w}^T \mathbf{w}$. $\frac{\partial E}{\partial \mathbf{w}} = 0 \implies$ $\frac{-2}{2} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i\} \boldsymbol{\phi}^T(\mathbf{x}_i) + \frac{2\lambda}{2} \mathbf{w}^T = 0 \implies$ $\sum_{i=1}^{N} t_i \boldsymbol{\phi}^T(\mathbf{x}_i) = \sum_{i=1}^{N} \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) \boldsymbol{\phi}^T(\mathbf{x}_i) + \lambda \mathbf{w}^T \implies$

- $\sum_{i=1}^{N} t_i \boldsymbol{\phi}^T(\mathbf{x}_i) = \mathbf{w}^T \left(\sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}^T(\mathbf{x}_i) + \lambda \mathbf{I}\right) \implies$

  $= \mathbf{w}^T \left(\quad + \lambda \mathbf{I}\right) \implies$

  $\mathbf{t}^T \boldsymbol{\Phi} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda \mathbf{I})^T \mathbf{w} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda \mathbf{I})\mathbf{w} \implies$
- $\boxed{\mathbf{w} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda \mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{t}}$
- Note about the $||\mathbf{w}^T\mathbf{w}||$: May actually be implemented numerically as $||\mathbf{w}^T\mathbf{w}|| - c$, small c

Intertwined Histories



J.-L. Lagrange
[1736-1813]

A. Lavoisier
[1743-1794]

J.-B. J. Fourier
[1768-1830]

https://upload.wikimedia.org/wikipedia/commons/1/19/Lagrange_portrait.jpg

https://upload.wikimedia.org/wikipedia/commons/4/44/Lavoisier-statue.jpg

https://upload.wikimedia.org/wikipedia/commons/a/aa/Fourier2.jpg

Home Page

Title Page

Contents

Page 16 of 30

Go Back

Full Screen

Close

Quit

Classification

- $\mathbf{x} \rightarrow$ [Classifier] $\rightarrow \mathscr{C}_j$
- Three approaches to Classification:
  1. Simplest: Discriminant Functions:
     Functions which directly assign a class to $\mathbf{x}$.
     Linear Discriminant: the discriminant fns are lines/linear/hyperplanes
  2. Model them directly: e.g., Mixture of Gaussians. Represent as parametric models, optimise params using a training set
  3. Toughest: Generative Approach: Find $P(\mathscr{C}_j|\mathbf{x})$
     Find $P(\mathscr{C}_j|\mathbf{x})$ using the Bayes' Theorem:
     $P(\mathscr{C}_j|\mathbf{x}) = P(\mathbf{x}|\mathscr{C}_j)P(\mathscr{C}_j)/P(\mathbf{x})$. Models for:
     $P(\mathbf{x}|\mathscr{C}_j)$: class cond densities; $P(\mathscr{C}_j)$: priors

Home Page

Title Page

Contents

Page 17 of 30

Go Back

Full Screen

Close

Quit

Men of God...

Thomas Bayes [1701-1761] | G. J. Mendel [1882-1884] | M. Mitra [1968-]

https://upload.wikimedia.org/wikipedia/commons/d/d4/Thomas_Bayes.gif

https://upload.wikimedia.org/wikipedia/commons/3/3d/Gregor_Mendel_oval.jpg

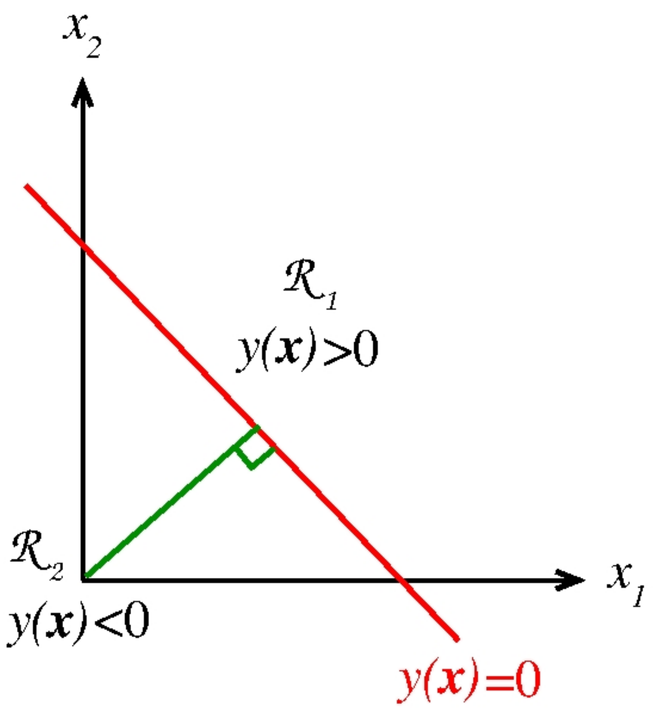
http://iseeindia.com/wordpress/wp-content/uploads/2011/11/Ramkrishna_Miss11736-290x290.jpg

Mahan Maharaj/Swami Vidyanathananda
2011 Shanti Swarup Bhatnagar Award in Math Sciences
Infosys Prize 2015 for Mathematical Sciences

Useful Generalisations of Linearity

- Linearity: Written equivalently in two ways:
  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$, for $(D+1) = M-$ dim data, $x_0 = 1$, or
  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$, for $D = (M-1)-$ dim data
- $y(\mathbf{x}, \mathbf{w}) = w_0 x_0 + \ldots w_{M-1} x_{M-1} = \sum_{j=0}^{M-1} w_j x_j$
- Model useful for Regression: linear comb of basis fns (lin/non-lin) $\boxed{y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})}$
- Generalising lin to scalar basis functions $\phi_j(\mathbf{x})$:
- $y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = w_o \phi_0(\mathbf{x}) + \ldots w_{M-1} \phi_{M-1}(\mathbf{x})$
- Model useful for Classification: fns (lin/non-lin) of the linear $\mathbf{w}^T \mathbf{x}$ (or $\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$) $\boxed{y(\mathbf{x}, \mathbf{w}) = f(\mathbf{w}^T \mathbf{x})}$
- Examples: Linear Regression, Neural Networks

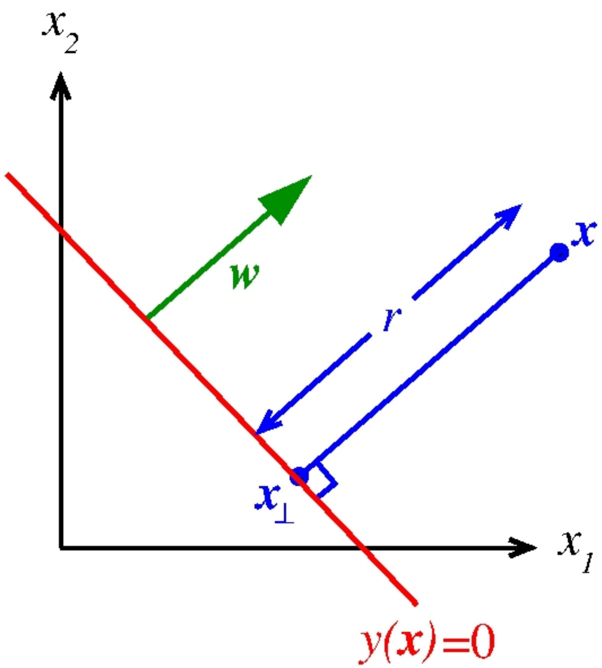
Discriminant Functions: 2 Classes



- $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$
- 2-D implicit from of eqn of a line is $ax+by+c=0$. Here, $y(\mathbf{x},\mathbf{w}) = w_2x_2 + w_1x_1 + w_0 = 0$
- $y(\mathbf{x}) = 0$: 1-D h'plane in 2-D
- Relative location of $\mathscr{R}_1$, $\mathscr{R}_2$ is immaterial: which is above/-below/to the left/to the right
- Physical Significance of w_0: measure of the dist from the origin Why? For $ax+by+c=0$, perp distance of (x_1,y_1) from the line is $\frac{|ax_1+by_1+c|}{\sqrt{a^2+b^2}}$ Perp dist of $y(\mathbf{x}) = 0$ from the origin $= \frac{|w_2(0)+w_1(0)+w_0|}{\sqrt{w_2^2+w_1^2}} = \frac{|w_0|}{||\mathbf{w}||}$, $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w} = \sum w_j^2$

Some more Physical Significance

- For two points $\mathbf{x}_A$ and $\mathbf{x}_B$ on the line $y(\mathbf{x}) = 0$:
  $y(\mathbf{x}_A) = 0 \implies \mathbf{w}^T\mathbf{x}_A + w_0 = 0$
  $y(\mathbf{x}_B) = 0 \implies \mathbf{w}^T\mathbf{x}_B + w_0 = 0$
  $\implies \mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0 \implies \mathbf{w} \perp$ line $y(\mathbf{x}) = 0$
- Phy Significance of perp dist of a point from a line

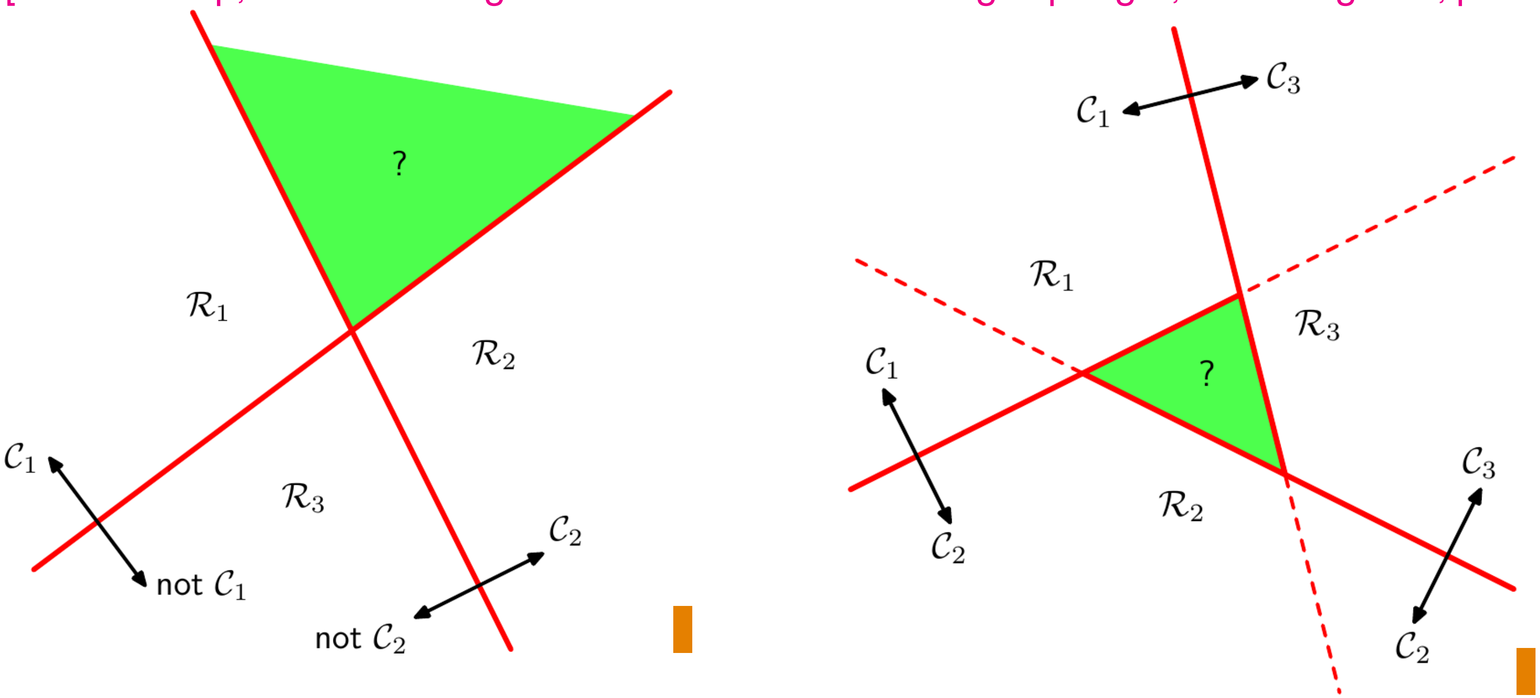


- $\mathbf{x} = \mathbf{x}_\perp + r\,\hat{\mathbf{w}} = \mathbf{x}_\perp + r\,\frac{\mathbf{w}}{||\mathbf{w}||}$
- Pre-multiply by $\mathbf{w}^T$ & add w_0:
- $\mathbf{w}^T\mathbf{x} + w_0 = \mathbf{w}^T(\mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||}) + w_0$
- $\implies y(\mathbf{x}) = (\mathbf{w}^T\mathbf{x}_\perp + w_0) + r\frac{||\mathbf{w}||^2}{||\mathbf{w}||}$
- $\implies \boxed{r = \frac{y(\mathbf{x})}{||\mathbf{w}||}}$
- Consistent with perp distance of (x_1, y_1) from line: $\frac{ax_1+by_1+c}{\sqrt{a^2+b^2}}$

Discriminant Functions: K Classes

- Building a $K-$ Classifier from 2-class ones

[C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006. Fig. 4.2, p. 183]



- One-versus-Rest
- $K-1$ classifiers, each of which solves the 2-class $\mathscr{C}_j$ vs. not $\mathscr{C}_j$

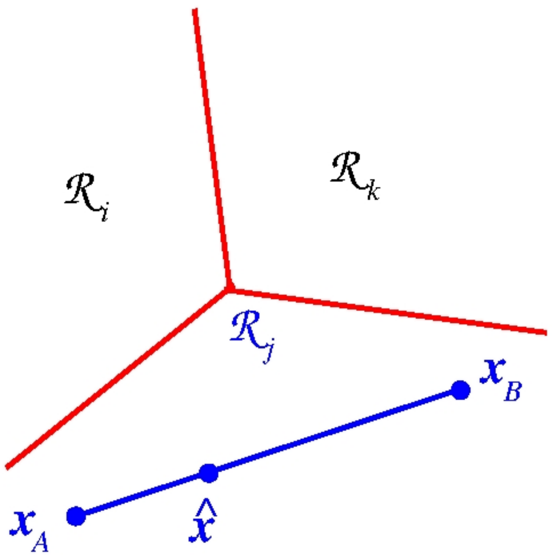
- One-versus-One
- ${}^K C_2$ 2-class classifiers
- Ambiguity here also!

- e.g., Tree-SVM? Explicitly define the hierarchy!

Elegant: $K-$ Class Discriminant!

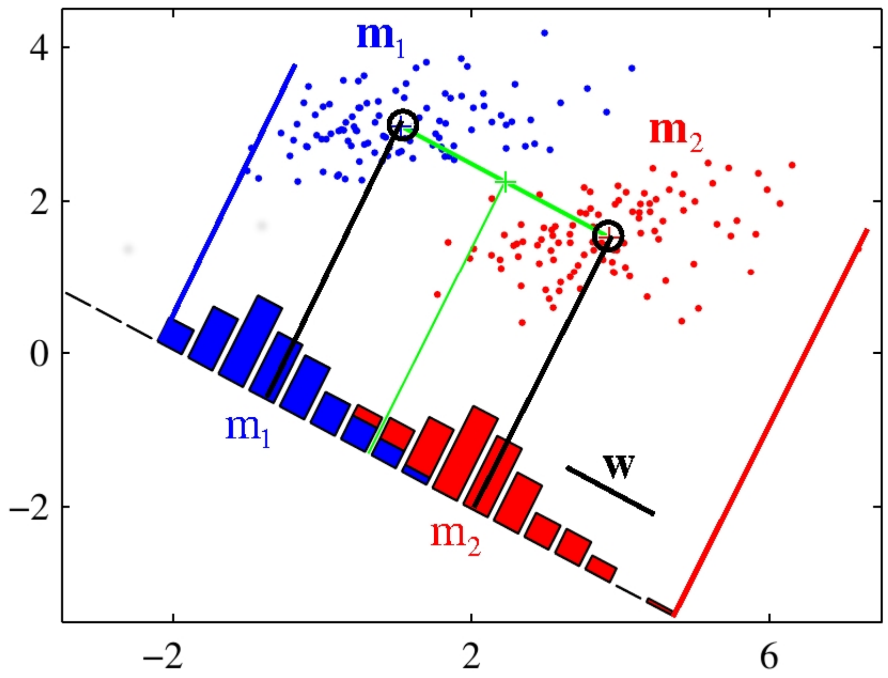
$$\boxed{y_j(\mathbf{x}) = \mathbf{w}_j^T \mathbf{x} + w_{j0}}$$

- Decision rule: Class $\mathscr{C}_j$ if $y_j(\mathbf{x}) > y_l(\mathbf{x}) \forall j \neq j$
- Decision boundary b/w $\mathscr{C}_j$ & $\mathscr{C}_k$: $y_j(\mathbf{x}) = y_k(\mathbf{x})$
- $\implies \mathbf{w}_j^T \mathbf{x} + w_{j0} = \mathbf{w}_k^T \mathbf{x} + w_{k0}$
- $\implies (\mathbf{w}_j - \mathbf{w}_k)^T \mathbf{x} + (w_{j0} - w_{k0}) = 0$
- $(D-1)-$ dim hyperplane, same form as the 2-class case: analogous properties
- The decision region for a multi-class linear discriminant must be convex & singly connected
- Enforced by the formulation! How?



- $\hat{\mathbf{x}}$ lies on the line b/w $\mathbf{x}_A$ & $\mathbf{x}_B$
- $\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1-\lambda)\mathbf{x}_B$
- Discriminant Fn Convexity
- $y_j(\hat{\mathbf{x}}) = \mathbf{w}_j^T \hat{\mathbf{x}} + w_{j0} = \mathbf{w}_j^T(\lambda \mathbf{x}_A + (1-\lambda)\mathbf{x}_B) + w_{j0} =$
- $\lambda \mathbf{w}_j^T \mathbf{x}_A + (1-\lambda)\mathbf{w}_j^T \mathbf{x}_B + w_{j0} = \lambda[\mathbf{w}_j^T \mathbf{x}_A + w_{j0}] + (1-\lambda)[\mathbf{w}_j^T \mathbf{x}_B + w_{j0}] + w_{j0} - \lambda w_{j0} - (1-\lambda) w_{j0}$
- $\Longrightarrow$ $\boxed{y_j(\hat{\mathbf{x}}) = \lambda y_j(\mathbf{x}_A) + (1-\lambda) y_j(\mathbf{x}_B)}$

Some Physical Significance



- Points $\mathbf{x}$ in 2-D space with means $\mathbf{m}_1$ & $\mathbf{m}_2$ (2 classes)
- A line: $w_2 x_2 + w_1 x_1 + w_0 = 0$: $\mathbf{w}^T\mathbf{x} + w_0 = 0$
- $\mathbf{w} \perp \mathbf{x}$, perp dist from [0, 0] = $\frac{w_0}{||\mathbf{w}^T\mathbf{w}||}$
- All 2-D spaces can be superimposed (Euclidean, here): $[x_2\ x_1]^T$ or $[w_2\ w_1]^T$
- $\mathbf{w}^T\mathbf{x}$: all points $\mathbf{x}$ are projected onto line $\mathbf{w}$
- Line-Point Duality. Line $\mathbf{w}$: by intercepts w_2 & w_1
- Means (2-D points) $\mathbf{m}_1$ & $\mathbf{m}_2$ are projected to 1-D projections m_1 & m_2. Each point $\mathbf{x}$ to x