Linear models for Classification

linear model for classification

- assign input to one of different classes
- Requirement: classes - disjoint
- I/p space divided -> decision regions whose boundaries are decision boundaries / decision surfaces
- decision surfaces - are linear functions of the i/p vector x => D-1 dimensional hyperplanes, - data sets - are linearly separable
- While linear regression - t vector of numbers whose values are to be predicted
- In Classification -> 2 or more class -> t = 0 (class c1) or 1 (class C2)
- value of t is probability of class C1
- W -> weight vector w_0 _bias
- $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ Negative of bias -> Threshold

- more than 2 -> vector of length k where J^{th} non zero => Cj

Approaches to classification

- classification: A) Discriminant function -> vector x to a class
- B) Probabilistic discriminative more powerful -> conditional probability distribution p(Ck/x) in inferences and use this to make optimal decisions <= separate inference and decision
- to determine conditional probability p(Ck/x) 2 ways: 1) represent parametrically and then optimize parameters (with training set)
- C) Probabilistic generative approach - model class conditional densities - p(x/Ck) with prior probabilities p(Ck) and calculate posterior probabilities p(Ck/x) using Bayes theorem
- $p(C_k|x) = p(x|C_k)p(C_k) / p(x)$
- D) Bayesian Logistic Regression

- Predict class labels - transform linear fn of w using a *non-linear* function $y(x) = f(w^Tx + w_0)$
- f is called as activation function (in statistics its inverse is called a link function)

- decision surfaces where "y(x) is constant" => $w^Tx + w_0$ is a constant
- so decision surfaces are linear (even though f is non-linear) So class of modes described by $y(x) = f (w^Tx + w_0)$ <= *generalized linear models*
- But as f is non-linear, analysis is more complex than linear regression models

- Discriminant functions: Input vector x is assigned to one of *K* classes
- linear discriminants - decision surface is a hyperplane (hyperplane delineates one class from another)
- Two classes:
- Linear function of I/p vector x $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$
- 2 points Xa and Xb on the decision surface. => y(Xa)=y(Xb) =0

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_{\mathrm{A}} - \mathbf{x}_{\mathrm{B}}) = 0$$

- W is orthogonal to every vector on decision surface=> w determines orientation of decision surface

- For point x on decision surface y(x)=0 => distance from origin to decision surface = $$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$

- Note: Bias parameter decides location of decision surface
- $\mathbf{x}_{\perp}$ - orthogonal projection of X on decision surface

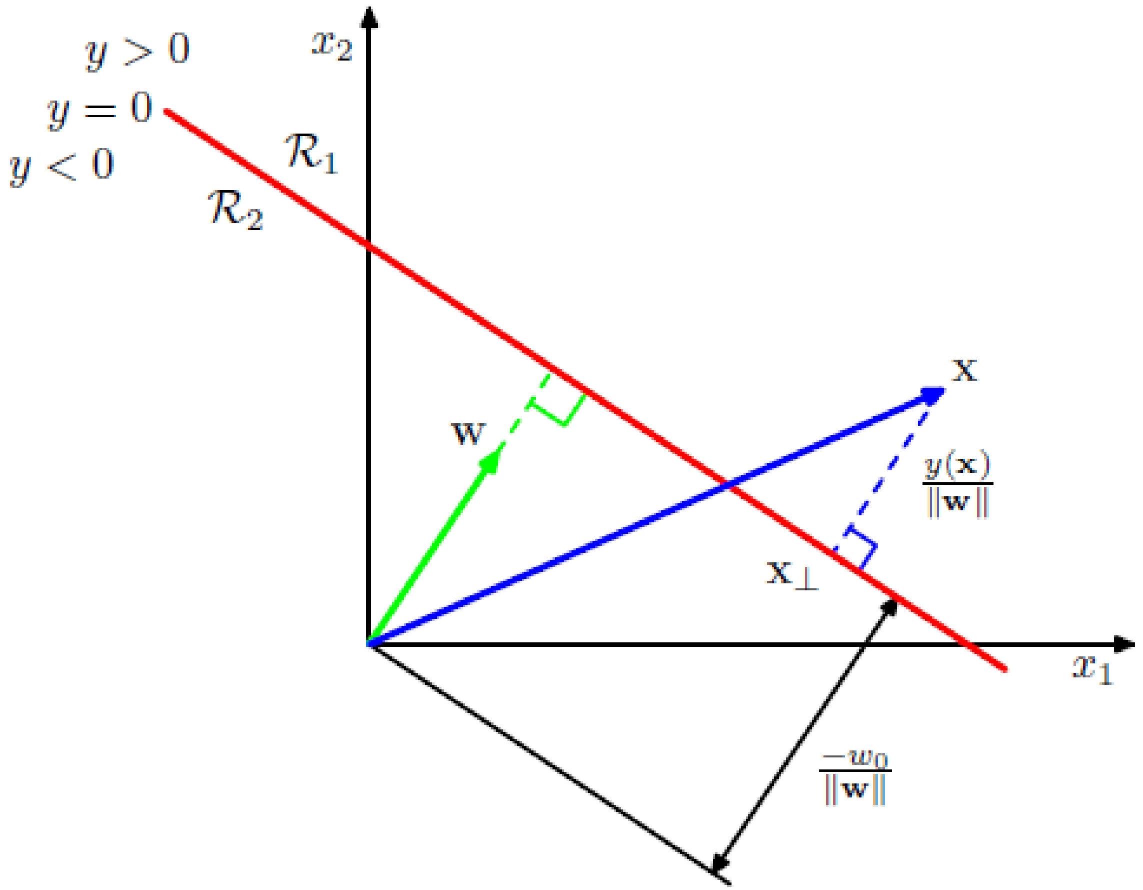
$$\mathbf{x} = \mathbf{x}_{\perp} + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

- Multiply by w^T and add w_0. Use y(x) = $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$ and
- $$y(\mathbf{x}_{\perp}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\perp} + w_0 = 0$$

Decision surface – red is perpendicular to w
displacement from origin – w0

x distance from decision surface

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$$



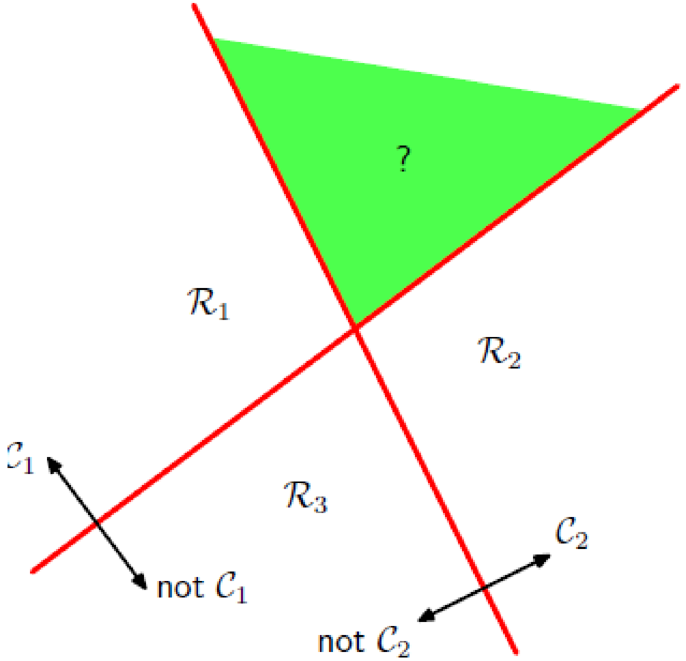
- For convenience : introduce dummy I/p: X_0 =1 and define
- $\widetilde{\mathbf{w}} = (w_0, \mathbf{w})$ and $\widetilde{\mathbf{x}} = (x_0, \mathbf{x})$
- Results in $$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}.$$

- Decision surfaces are D dimensional hyperplanes passing through origin of D+1 dimensional I/p space

Multiple classes

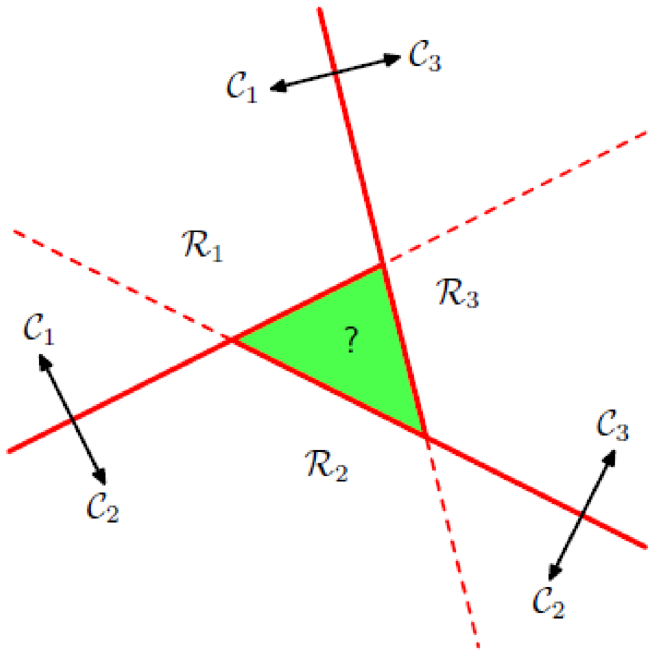
- Wrong approaches:
  - A) combine two class discriminants

  Use K-1 classifiers each solving a two class discrimination problem <- one vs rest classifier



Multiple classes – wrong approach

- One versus one classifier:
- K(K-1)/2 classifiers : one for every pair of classes



Multiple classes - correct approach

- Use Single K class discriminant using K linear functions

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

- Assign a point to a class C_k if $y_k(x) > y_j(x)$
- Decision boundary between C_k and C_j is $y_k(x) = y_j(x)$
- => (D-1) dimensional hyperplane given by:

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

- Above is similar to 2 class equation

- Decision regions for such discriminants are Singly connected and convex

Learn parameters of linear discriminants

- 3 approaches exist:
- a) Least squares b) Fishers discriminant c) Peceptron
- <u>Least Squares:</u>
- Situation: K classes with 1 of K binary coding scheme for target vector t
- Using least squares, approximates conditional expectation $\mathbb{E}[\mathbf{t}|\mathbf{x}]$
- This is = vector of posterior probabilities <- pbm as probabilities are difficult to correctly approximate

- Each class is $y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$
- Same as $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$
- $\widetilde{\mathbf{W}}$ k^th column corresponds to (D+1 dimensional vector)
- $\widetilde{\mathbf{x}}$ Is augmented i/p vector $(1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$ with dummy i/p x_0 =1
- X is member of class where $y_k = \widetilde{\mathbf{w}}_k^{\mathrm{T}}\widetilde{\mathbf{x}}$ has largest value
- <u>To do:</u> Determine matrix $\widetilde{\mathbf{W}}$ by minimizing sum of squares error fn.
- Given: T matrix whose n^th row is vector $\mathbf{t}_n^{\mathrm{T}}$
- $\widetilde{\mathbf{X}}$ matrix whose n^th row is $\widetilde{\mathbf{x}}_n^{\mathrm{T}}$

- Sum of squares error function = $E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$

- Make derivative w.r.t. to W, as zero => solution for $\widetilde{\mathbf{W}}$
- Is $\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T}$

- $\widetilde{\mathbf{X}}^{\dagger}$ Is pseudo inverse of X

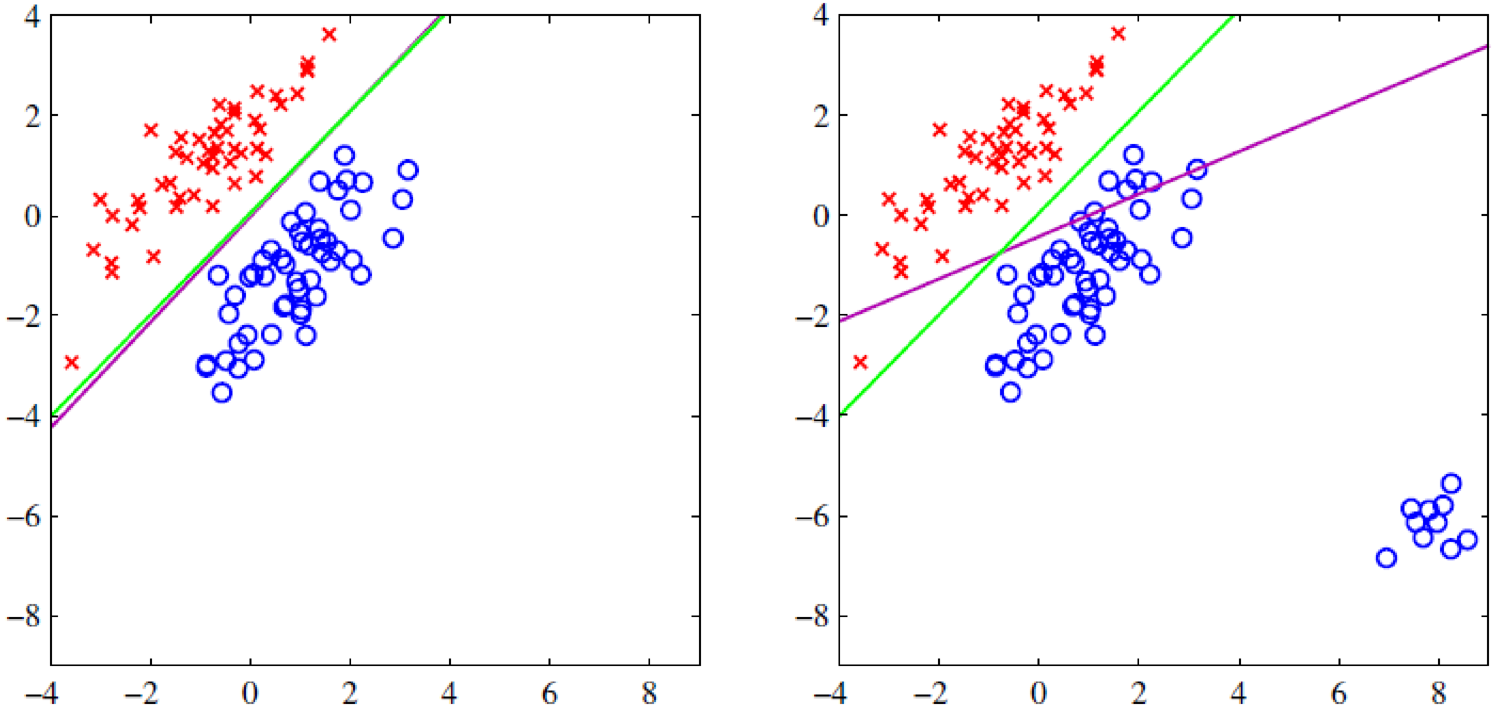
- Therefore Discriminant function is = $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}}\widetilde{\mathbf{x}}.$
- ---
- If every target vector in training set satisfies linear constraint $\mathbf{a}^{\mathrm{T}}\mathbf{t}_n + b = 0$
- Then model prediction for any x will satisfy same constraint =>

$$\mathbf{a}^{\mathrm{T}}\mathbf{y}(\mathbf{x}) + b = 0.$$

- Using 1-of-k coding scheme => (x) will sum to1 in all predictions

- **Pbm.** in using least squares approach-> 1) outliers not handled

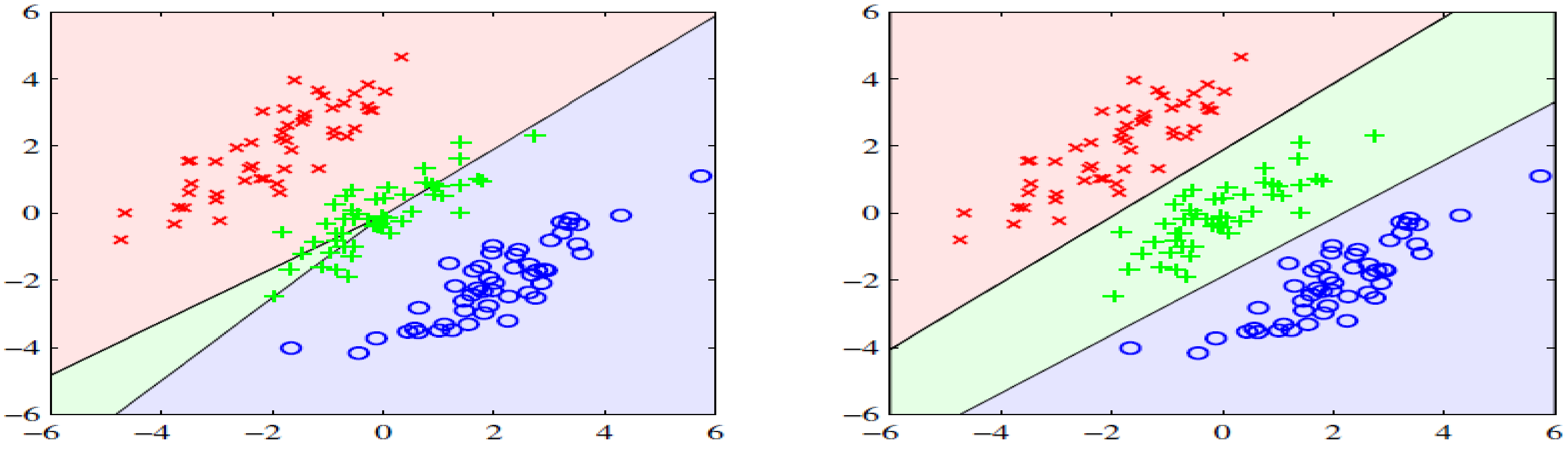
-



"Too correct" Penalized

- Adding data on right bottom skews least squares solution: (violet) vs logistic regression

- 2nd problem in using least squares:

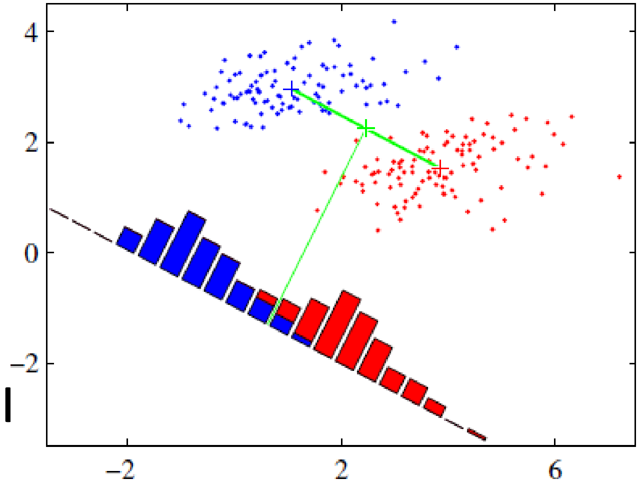


- Least squares – green class not covered   Logistic regression - correct

- (Reason – least squares assumes Gaussian distribution, which is not the case always)

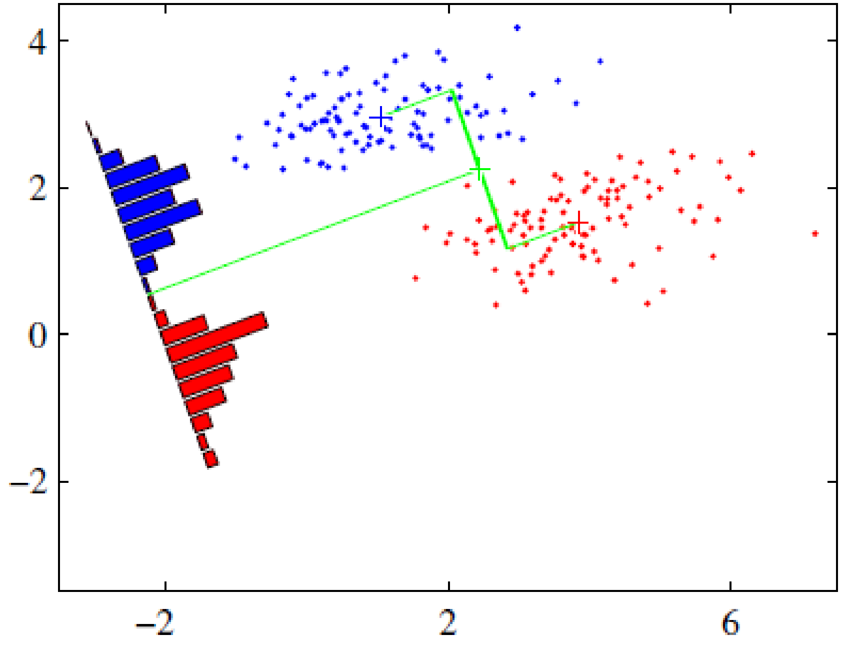
Fishers linear discriminant

- Dimensionality reduction used for linear classification.
- 2 dimensional to one dimension using $y=w^T x$
- Threshold on y => standard linear classifier ($y >= -w_0$ => class C_1)
- Pbm: Mapping to 1 dimension loses information, classes overlap. Handle by adjusting components of weight vector W (which maximizes separation between classes)
- Example: 2 classes C1 and C2 with N1 and N2 points respectively
- Mean vector is $$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$

- Measure of separation between 2 classes: Distance between projected Class means
- Choose w to maximize $(m_2-m_1) = w^T(m_2-m_1)$
- M_k -> mean of projected class k, is equal to $w^T m_k$
- 

- Sa > histogram of projection of means

- Fishers linear discriminant: Maximize a function that will give large separation between projected class means and small variance within class => Minimize class overlap



- The 'Within class variance' for C_k is

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$

- Total within class variance = $s_1^2 + s_2^2$ for 2 classes
- Fisher criterion = Between class variance / Within class variance
- =

$$J(\mathrm{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

= $$J(\mathrm{w}) = \frac{\mathrm{w}^{\mathrm{T}}\mathrm{S}_{\mathrm{B}}\mathrm{w}}{\mathrm{w}^{\mathrm{T}}\mathrm{S}_{\mathrm{W}}\mathrm{w}}$$ S_B is between class variance and S_W is within class variance

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

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

- Differentiate J(w) w.r.t w => J(w) is maximized when

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

- Multiply by S_w^{-1}

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

- Fishers linear discriminant: Above
- Choose a threshold y_0: class C_1 if $y(x) >= y_0$ otherwise C_2
- *Relation to least square: Pl read*

Fisher discriminant for multiple classes

- K>2,
- Assume Dimensionality D is > K
- D' => Linear features y_k - ($y = w_k^t x$) : features grouped as vector Yand weight vectors w_k – columns of W
- $Y = W^T x$

$$S_W = \sum_{k=1}^{K} S_k \qquad S_k = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$$

- S_w is within class covariance

$$m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$$

- N_k -> number of patterns in C_k

- Total covariance matrix S_T =
- Within class covariance matrix S_W+ between class covariance matrix S_B

$$\mathbf{S}_{\mathrm{T}} = \mathbf{S}_{\mathrm{W}} + \mathbf{S}_{\mathrm{B}}$$

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

- <u>On D' dimensional y-space</u>

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k$$

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

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

- Find value that is large when "between class covariance is large" and "within class covariance is small"
- Example: $J(W) = \mathrm{Tr}\left\{ s_W^{-1} s_B \right\}$
- Rewritten as: $J(\mathbf{w}) = \mathrm{Tr}\left\{ (\mathbf{W} \mathbf{S}_W \mathbf{W}^T)^{-1} (\mathbf{W} \mathbf{S}_B \mathbf{W}^T) \right\}$
- Above is to be maximized

Perceptron

- Rosenblat - 1962
- Two class linear discriminant
- Input vector X becomes a feature vector $\phi(\mathbf{x})$ by a fixed non - linear transformation
- Create a generalized linear model $y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right)$

- f is nonlinear activation function = +1 for x >= 0 , -1 for x < 0
- $\phi(\mathbf{x})$ Will include bias component $\phi_0(\mathbf{x}) = 1$

- Cannot use number of misclassified patterns as error function as:
- Error is piece wise constant function of W. Has discontinuities wherever a new data point happens due to change in W.
- Gradient becomes zero
- Alternative: Perceptron criterion
- Patterns X_n in Class C_1 => $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) > 0$
- and patterns X_n in Class C_2=> $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) < 0$
- With t in range +1 to –1 all patterns -> $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n > 0$
- Zero error for correct classification
- Minimize $-\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n$

- Gives Perceptron criterion as:

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

- M – set of all misclassified patterns

- Apply stochastic gradient descent algorithm to get
- Change in weight vector =

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

- Learning rate η

- In every iteration error becomes less:
  - Set learning rate to 1 and use $\|\phi_n t_n\|^2 > 0$

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

  - Issues: Change in weight causes previously correct to become misclassified
  - Nonlinearly separable data sets perceptron approach will not converge

  - Similar to Perceptron is ADALINE.

Probabilistic Generative Model

--> Classification using ideas from the distribution of data

Model class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ and class priors $p(\mathcal{C}_k)$.

Then compute posterior probabilities through Bayes theorem

=> we will see that **Posterior linear probabilities are = Generalized Linear models with logistic sigmoid (for k=2) or softmax (for k>2)**

Case of 2 classes:

Posterior probability for Class $_1$:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

Define:

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

- Posterior probability for class C_1 becomes:

$$\frac{1}{1+\exp(-a)}$$

- Now define $\sigma(a)$ the Logistic Sigmoid Function as $\sigma(a) = \frac{1}{1+\exp(-a)}$

- Posterior probability for C_1 = $\sigma(a)$

- Sigmoid: S - shaped (squashing function)
- Inverse of Sigmoid function is Logit function = $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$

- Logit function: represents the log of the ratio of probabilities for the 2 classes:
- ln [p(C1|X)/p(C2|X)] - <-- called as "log odds"
- ==
- K > 2 classes:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

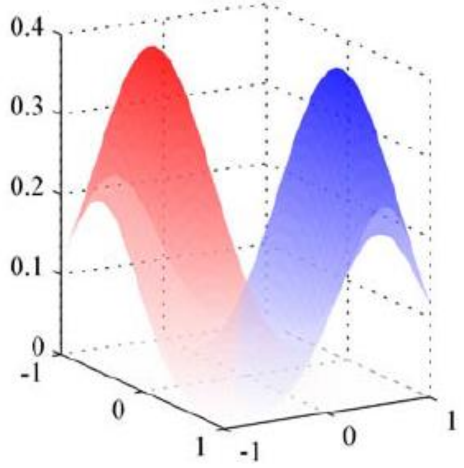
- $= \frac{\exp(a_k)}{\sum_j \exp(a_j)}$ <--Normalized exponential <- SoftMax fn

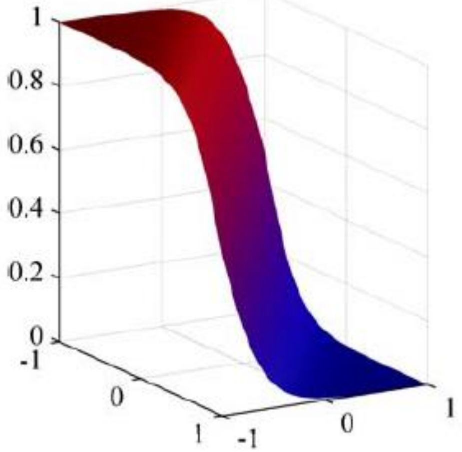
- ($a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$)

Class conditional densities – continuous, discrete

- Class conditional density -> $p(x/C_k)$
- Continuous input:
- Assume Gaussian and all classes share same covariance matrix.
- Density for class: C_k is = $$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$
- For Two classes: Already seen (s:29) $$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$
- ($\mathbf{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ $\quad w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$)

- Quadratic terms in x from exponents of Gaussian distribution are cancelled (<-- common variance matrices)
- Result is linear function of x in argument of logistic sigmoid.
- Illustration:
- 2 Dim I/p space
- LHS: 2 classes
- red , blue
- RHS: Posterior probability
- $p(C_1/x)$ is logistic sigmoid of linear fn (x)
- Red = $p(C_1/x)$ Blue = $p(C_2/x)$= 1-$p(C_1/x)$
- Decision boundaries => surfaces where p(Ck/x) are constant => linear fn(x) => decision boundaries are linear in I/p space





- Prior probabilities only lined to W_0=>changes in prior only shift decision boundary

- For general K classes: $$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

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

- Again linear functions of X
- ---
- If shared covariance matrix is not mandatory, then cancellation will not happen => quadratic functions of x => quadratic discriminant

Maximum likelihood solutions

- To use Maximum likelihood:
  - Put class densities $p(c_k/x)$ in a parameterized functional manner
  - Use prior probabilities
  - Data set of observations of x with class labels
- Two classes: Data set= $\{X_n, t_n\}$ $t_n=1 \Rightarrow$ Class C_1 $t_n=0 \Rightarrow C_2$
- Prior class probability $p(C_1) = \pi$ $p(C_2) = 1-\pi$
- Xn belonging to C_1 , $t_n = 1 \Rightarrow$ $p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$
- For C_2, $t_n=0 \Rightarrow$ $p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$
- Likelihood fn. = $p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$

- Maximize log of the likelihood function:

1) Maximization w.r.t π

Terms dependent on π are

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

Set derivative w.r.t π = 0, results in:

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

=> Maximization of log likelihood function is = Fraction of points in C_1

2) Maximize w.r.t to $\boldsymbol{\mu}_1$

- Terms depending on $\boldsymbol{\mu}_1$ are

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

- =

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

- Setting derivative w.r.t $\boldsymbol{\mu}_1$ as zero gives

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathbf{x}_n$$

- This is mean of all input vectors assigned to C_1
- Correspondingly $$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n$$

- 3) Maximum likelihood for shared covariance matrix
- Elements that depend on covariance are

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

- Rewriting this in terms of second mean

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

- Define $\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1}(\mathbf{x}_n-\boldsymbol{\mu}_1)(\mathbf{x}_n-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2}(\mathbf{x}_n-\boldsymbol{\mu}_2)(\mathbf{x}_n-\boldsymbol{\mu}_2)^{\mathrm{T}}.$$

- Above equation becomes $-\frac{N}{2}\ln|\mathbf{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\mathbf{\Sigma}^{-1}\mathbf{S}\right\}$

- As it is a Gaussian distribution $\Sigma = \mathbf{S}$

- This is a weighted average of the covariance matrices of each class separately
- Above can be generalized to multiple classes
- Note : outliers are not handled as Max Likelihood estimation of Gaussian outlier is not handled

Discrete Features

- Binary feature values: $x_i \in \{0, 1\}$

- D inputs => distribution is table of 2^D number for each class
- Containing 2^D-1independent variables
- Exponential growth. To handle this use naïve Bayes (feature values are kept independent, as per class C_k
- Class conditional distributions become $$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1-\mu_{ki})^{1-x_i}$$
- This has D independent parameters for eacn cıass

- So $a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$ becomes

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k)$$

Exponential family

- **Posterior linear probabilities are = Generalized Linear models with logistic sigmoid (for k=2) or softmax (for k>2)**
- More generally assume $p(x/C_k)$ belong to exponential family of distributions
- Distribution of x can be $$p(\mathbf{x}|\boldsymbol{\lambda}_k) = h(\mathbf{x})g(\boldsymbol{\lambda}_k)\exp\left\{\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$
- Now consider u(x) = x. Introduce a scaling parameter 's'
- Exponential family - class conditional densities become

$$p(\mathbf{x}|\boldsymbol{\lambda}_k, s) = \frac{1}{s} h\left(\frac{1}{s}\mathbf{x}\right) g(\boldsymbol{\lambda}_k) \exp\left\{\frac{1}{s}\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{x}\right\}$$

- Each class has its own parameter vector $\boldsymbol{\lambda}_k$, but all classes use same scaling parameter 's'

- For 2 class substitute above in $a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$

- This gives: Posterior class probability is = logistic sigmoid on a linear function a(x) where a(x) is =

$$a(\mathbf{x}) = (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)^{\mathrm{T}}\mathbf{x} + \ln g(\boldsymbol{\lambda}_1) - \ln g(\boldsymbol{\lambda}_2) + \ln p(\mathcal{C}_1) - \ln p(\mathcal{C}_2)$$