H1 Lecture 5: Classification, Decisions and Discriminants - 10/02/20

H2 2-Class Classification

We will mainly focus on classfication between two classes C_1 and C_0. e.g. *yes* or *no*, *true* or *false*, *good* or *bad*, *present* or *absent*

Similar to regression, our data consists of vector inputs, $\mathbf{x} \in \mathbb{R}^D$ and targets $t = 1$ for C_1 and $t = 0$ for C_0. Can think of t as the probability of class C_1.

> The task is to ***learn a mapping*** $y : \mathbb{R}^D \to \{0, 1\}$ ***that can accurately predict class for new data (unseen at training time)***

H2 1-of-K Classification

For example, predict digit from image. For more than two classes ($K > 2$), convenient to define targets as a ***one-hot-vector*** - a single element 1 and all others 0, i.e., if $\mathbf{t}$ represents class C_j, then $t_j = 1$ and all other $t_k = 0$.

For example, if $K = 5$ classes, to encode class 2, we have:

$$\mathbf{t} = (0, 1, 0, 0, 0)^T$$

> Again, we can interpret t_k as the probability that the class is C_k

H2 Example: X-Ray Diagnosis

Say we have encoded an X-ray image as input-vector $\boldsymbol{x}$, and want to use this to determine whether the patient has cancer or not.

Classes are:

- absent, $\mathcal{C}_0$ $(t = 0)$
- present, $\mathcal{C}_1$ $(t = 1)$.

General **inference** problem:

- determine $p(\boldsymbol{x}, \mathcal{C}_k)$ for all k (or equivalently $p(\boldsymbol{x}, t)$)

Ultimately, though, we must decide whether to give treatment or not.

- Our **decision step**

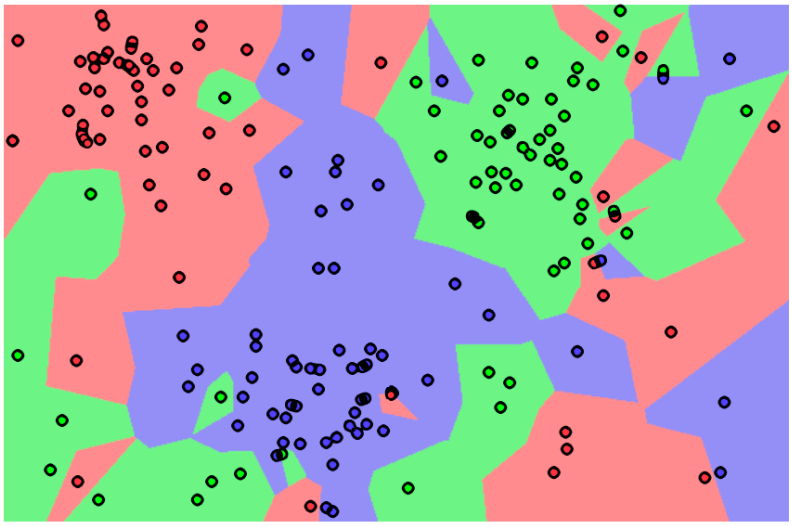
Decision is straightforward, if we solve the inference problem.

H2 Decision Regions and Boundaries

Given any input $\mathbf{x}$ we want to predict its class.

- Need rule to assign each $boldx$ to an available class, C_k

- Rule divides input space into ***decision regions*** , R_k
- Points $\mathbf{x} \in R_k$ assigned to Ck
- ***Decision boundaries*** separate decision regions
- Decision regions need not be connected



Example decision regions for 1NN classifier

H2 Using Probabilities to Minimise Misclassification

Assume we have solved the inference problem. We are interested in the probability of class, C_k , given input, $\mathbf{x}$:

$$p(C_K|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

- $p(C_k)$ is ***prior*** for class k
- $p(C_k|\mathbf{x})$ is ***posterior*** for class k given input
- For our X-ray example:
  - $p(C_1)$ - probability that ***typical patient has cancer***
  - $p(C_1|\mathbf{x})$ - ***revised probability*** in light of the X-ray image, $\mathbf{x}$
  - $p(C_0|\mathbf{x}) + p(C_1|\mathbf{x}) = 1$

To minise the probability of incorrect classification, we ***choose the class with the highest probability***.

For 2-classes, a misclassification occurs, when we assign input $\mathbf{x}$ to class C_1 when it should be C_0 (and vice versa).

- If we pick a data-point, $\mathbf{x}$, at random:

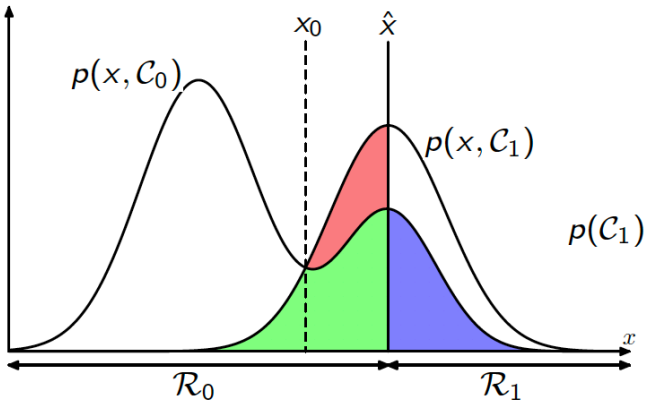
$$\begin{aligned} p(mistake) &= p(\mathbf{x} \in R_0, C_1) + p(\mathbf{x} \in R_1, C_0) \\ &= \int_{R_0} p(\mathbf{x}, C_1) d\mathbf{x} + \int_{R_1} p(\mathbf{x}, C_0) d\mathbf{x} \end{aligned}$$

- To minimise $p(mistake)$, we choose decision region so that:

$$p(\mathbf{x}, C_1) > p(\mathbf{x}, C_0) \implies \mathbf{x} \in R_1$$

> *Multivariate Integration*:
>
> - Non-square region, R, delt with by varying bounds
>
> $$\int_R f(x,y)dx\,dy = \int_{l_x}^{u_y}\int_{l_x(y)}^{u_x(y)} f(x,y)dx\,dy$$
>
> - Higher dimensions extend this idea

Minimising Misclassification: 1d Example



$$p(\mathcal{C}_1) = \int_{\mathcal{R}_1} p(x, \mathcal{C}_1)\, dx$$

If we choose $\mathcal{R}_0$ and $\mathcal{R}_1$ as above:

- $\hat{x}$ is decision boundary
- Error probability: area of green, red and blue region
- For $x < \hat{x}$, class 1 misclassified (green+red).
- For $x \geq \hat{x}$, class 0 misclassified (blue)
- To minimise error, set $\hat{x} = x_0$ (minimise red region)

For K classes, the probability of a mistake:

$$\begin{aligned} p(mistake) &= \sum_k \sum_{j \neq k} p(\mathbf{x} \in R_j, C_k) \\ &= \sum_k \sum_{j \neq k} \int_{R_j} p(\mathbf{x}, C_k) d\mathbf{x} \end{aligned}$$

Labels independently assigned, and so best choice for $\mathbf{x}$ is:

$$y^* = \arg\max_j p(\mathbf{x}, C_j) = \arg\max_j p(C_j|\mathbf{x})$$

> Minimum Misclassification Label for $\mathbf{x}$:
>
> $$y(\mathbf{x}) = \arg\max_j p(C_j|\mathbf{x})$$

Asymmetric Losses

In many cases, some misclassfications are less important than others. For our X-ray example:

- false postive - classifying a healthy patient as having cancer
- false negative - classifying cancer patient as healthy
- Althought undesirable, false positives may have less damaginhg consequences than false negatives (or vice versa)
- Can encode this in a ***loss matrix*** , e.g.

	predict cancer	**predict normal**
true cancer	0	1000
true normal	1	0

H3 Minimising Expected Loss

Given loss matrix L, we wish to minise the expected loss

$$E[L] = \sum_j \sum_k \int_{R_j} L_{kj} p(\mathbf{x}, C_k) d\mathbf{x}$$

where L_{kj} is loss for predicting class j for data-point from class k

Labels independently assigned, and so best choice for $\mathbf{x}$ is:

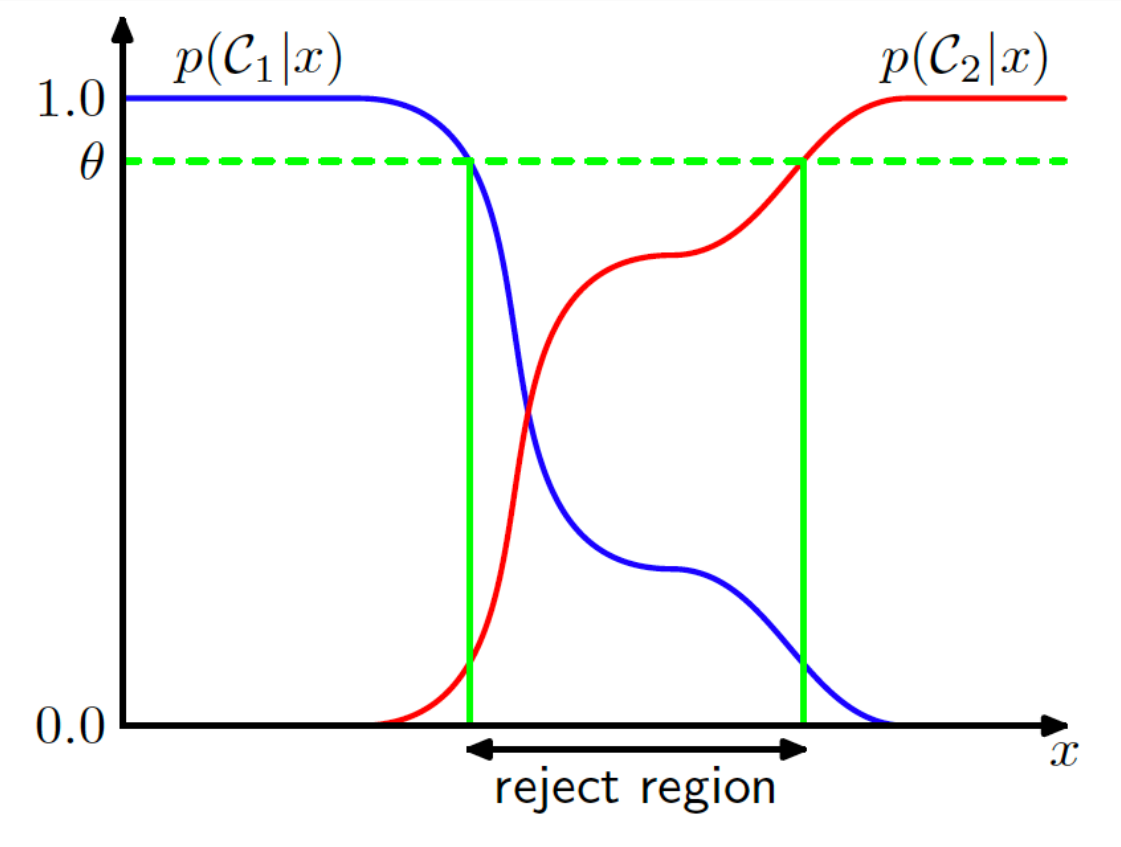
$$y^* = \arg\max_j \sum_k L_{kj} p(\mathbf{x}, C_k) = \arg\max_j \sum_k L_{kj} p(C_j|\mathbf{x})$$

> Minimum Expected-Loss Label for $\mathbf{x}$:
>
> $$y(\mathbf{x}) = \arg\min_j \sum_k L_{kj} p(C_k|\mathbf{x})$$

H3 The Reject Option

Within some regions of data-space:

- we are uncertain about class membership
- happens where the ***largest*** $p(C_k|\mathbf{x})$ is ***significantly less than 1***
- points in these regions have high classification error
- may prefer to ***reject*** (not classify) a point when all $p(C_k|\mathbf{x})$ are below a threshold value



Generative Models

Infer the joint probability $p(\mathbf{x}, C_k)$ *(or* $p(\mathbf{x}|C_k)$ *and* $p(C_k)$ *separately)*, use ***Bayes Rule*** to calculate posterior class probabilities:

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

and then use decision theory to determine class membership

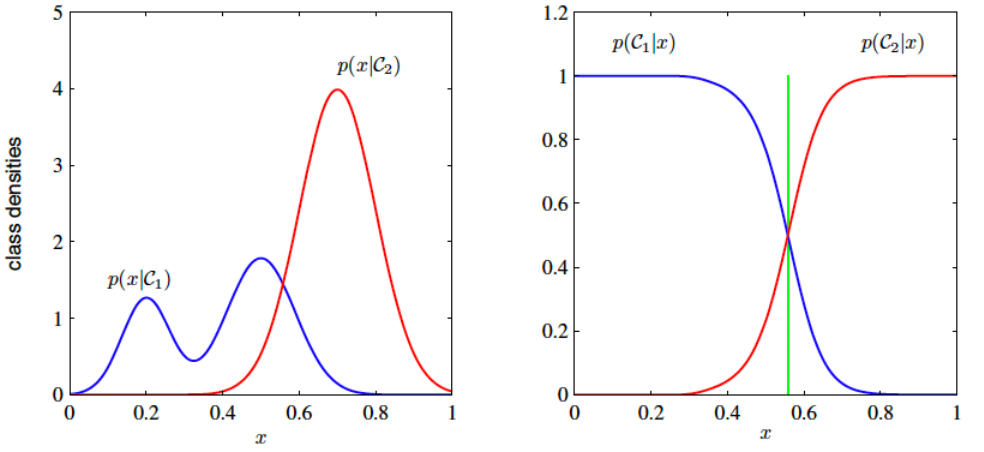
Advantages:

- We can ***generate*** synthetic data by sampling from model
- Either learn joint distribution $p(\mathbf{x}, C_k)$ or class-prior $p(C_k)$ and class-conditional $p(\mathbf{x}|C_k)$ separately
- Can calculate the marginal density of data, $p(\mathbf{x})$, useful for detecting outliers in new data (poor predictive performance)

Disadvantages

- Demanding in terms of data and computation, particularly if inputs, $\mathbf{x}$, have high dimension
- Class conditionals may have unnessessary detail, much cheaper to compute class-posterior directly $p(C_k|\mathbf{x})$

Generative vs Discriminative



- Generative approach (left) may model complexities that do not influence class predictions
- Posterior class probabilities of probabilistic discriminative approach (right)
- Non-probabilistic discriminative approach captured by class boundary (green vertical)

Probabilistic discriminative models can still

- ***minimise expected loss*** flexibly - if loss matrix changes, we do not need to relearn the full classification probram afresh
- have a ***reject option*** - requires an estimate of the misclassification rate for each input $\mathbf{x}$
- compensate for ***class prior*** - when training data has different class mixture from deployment, e.g. X-ray diagnosis classifier trained from hopsital dta, deployed at clinic
- ***combine models*** - break into sub-problems, then combine independent predictions

H2 Discriminative Models (Probabilistic)

Directly infer the posterior probabilities, $p(C_k|\mathbf{x})$, then use decision theory to determine class membership

H2 Discriminative Models (Non-Probabilistic)

Find a discriminant function $f(\mathbf{x})$ which ***directly maps each input, $\mathbf{x}$, onto a class label.***

H2 Discriminant Function

A ***discriminant*** is a function that takes input $\mathbf{x}$ and directly assigns it to one of K classes

- Restrict attention to ***linear discriminants*** , i.e. decision surfaces are hyperplanes
- Do this in two stages:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

for weight vector $\mathbf{w}$ and bias w_0

- We assign $\mathbf{x}$ to class C_1 if $y(\mathbf{x}) \geq 0$ and to C_0 otherwise
- Decision boundary defined by $y(\mathbf{x}) = 0$

H3 Linear Discriminants, 2 Classes

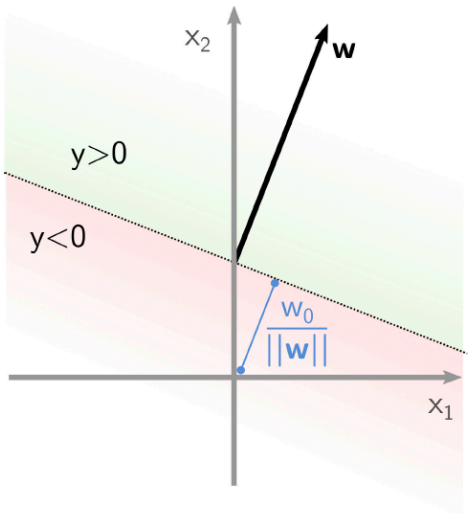
If $y(\boldsymbol{x}) \geq 0$ assign to class $\mathcal{C}_1$, or equivalently $\boldsymbol{x} \in \mathcal{R}_1$

If $y(\boldsymbol{x}) < 0$ assign to class $\mathcal{C}_0$, or equivalently $\boldsymbol{x} \in \mathcal{R}_0$

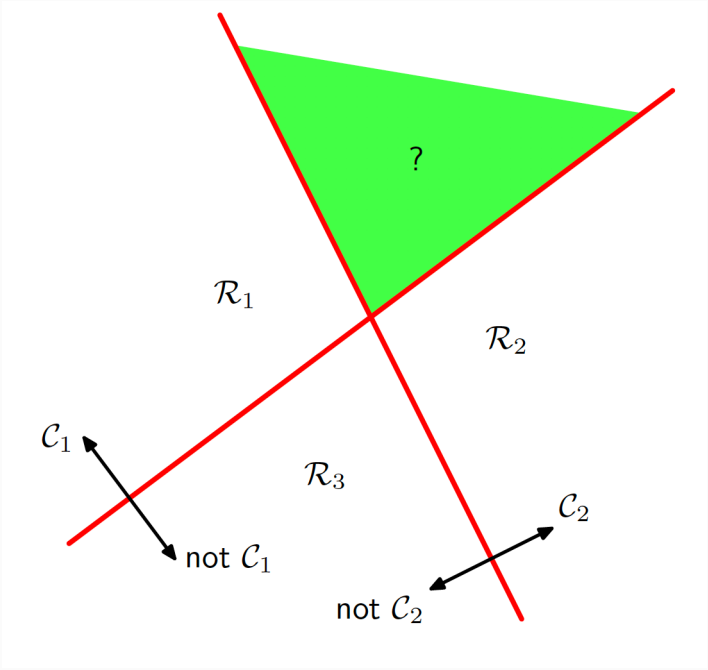
As we discussed before:

$$y(\boldsymbol{x}) = 0 \quad \Leftrightarrow \quad \frac{\boldsymbol{w}^T\boldsymbol{x}}{\|\boldsymbol{w}\|} = -\frac{w_0}{\|\boldsymbol{w}\|}$$

Signed perpendicular distance from decision plane given by $y(\boldsymbol{x}) = r$.



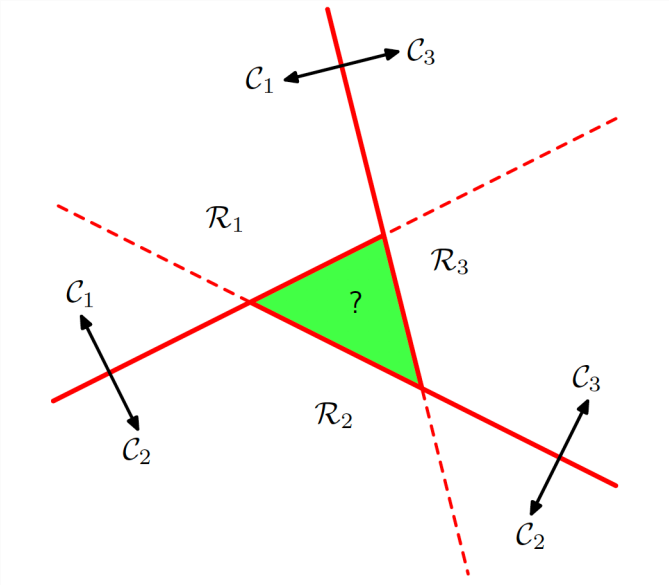
H3 Multiple Classes: One-versus-the-rest



Extend 2-class discriminants to K-class discriminants for ***one-versus-the-rest*** approach:

- Create $K-1$ classifiers, each separating two classes: C_k from points not in C_k
- Problems in regions ***assigned to more than one class***

H3 Multiple Classes: One-versus-one



Extend 2-class dirscrimnants to K-class discriminants fro ***one-versus-one*** approach:

- Create $\frac{K(K-1)}{2}$ binary discriminants, one for each pair of classes
- Problems in regions ***with no dominant class***

H3 A better Discriminant fro K-Classes

- Define a single K -class discriminant from K functions:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_k0$$

- assign $\mathbf{x}$ to class C_k if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$
- Boundary between R_k and R_j given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$ corresponds to $(D-1)$ -dimensional hyperplane

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

so same geometric properties as before

- But ***no ambiguous*** regions

H2 Loss Functions for Discriminants

H3 Least Squares for Classification

Consider a general classification problem with K classes

- Each class described by separate ***linear model***

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

- Together this forms a vector of outputs

$$\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_k(\mathbf{x}))^T$$

- Targets are ***one-hot*** vectors

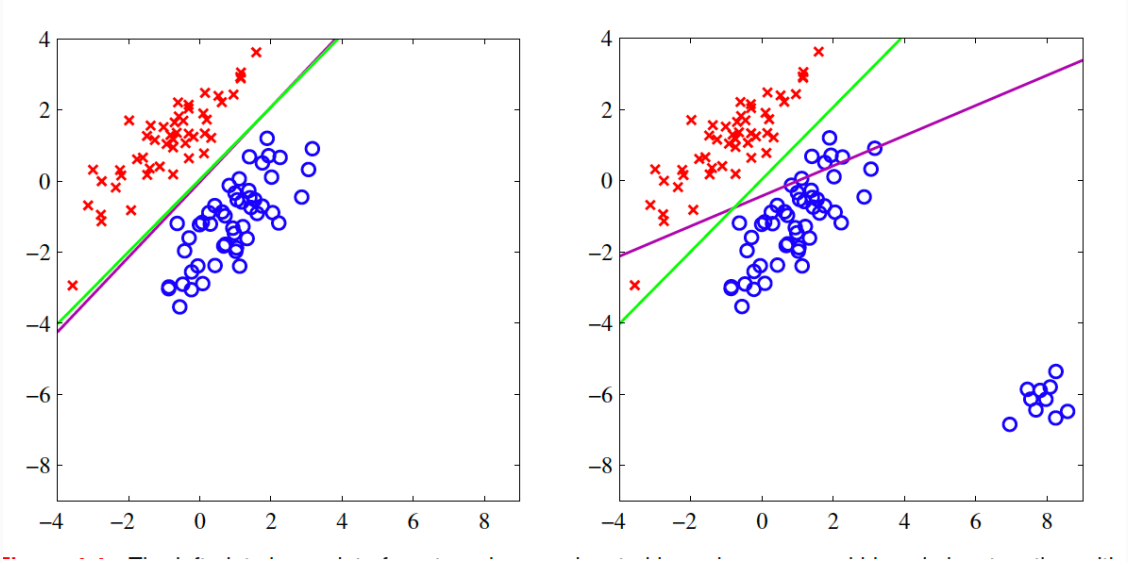
$$\mathbf{t}_n = (0, \ldots, 0, 1, 0, \ldots, 0)^T$$

- Least-squares minimises the average squared distance between vector prediction

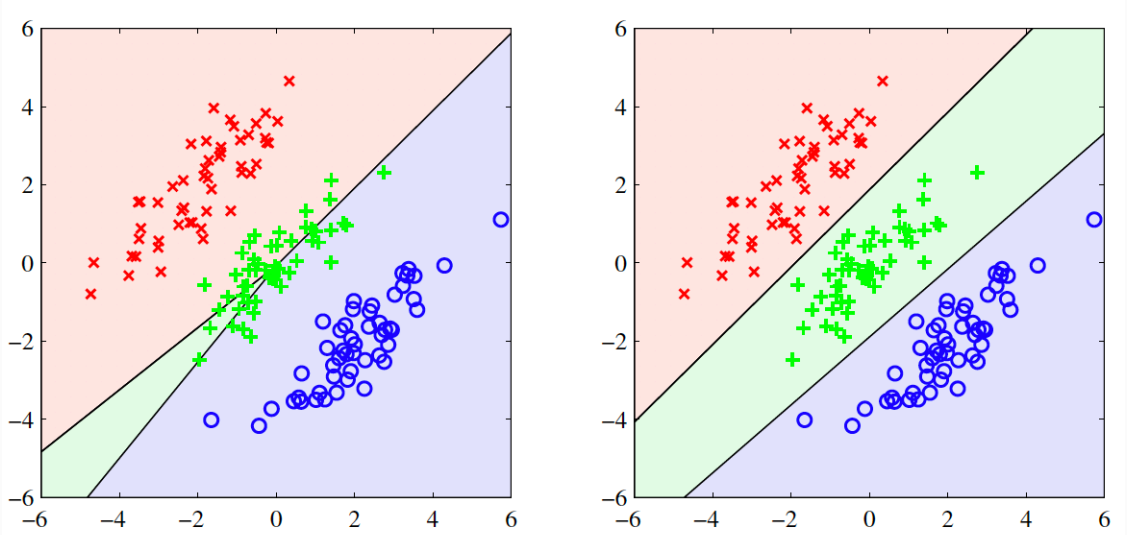
and vector target

$$\sum_{n-1}^{N}(\mathbf{y}(\mathbf{x}_n)-\mathbf{t}_n)^T(\mathbf{y}(\mathbf{x}_n)-\mathbf{t}_n)$$

H3 Problems with Least Squares



- Decision boundary least-squares (magenta)
- Decision boundary logistic regression† (green)
- Least-squares sensitive to outliers/points that are **too correct**



- Training points in 3 classes: ×, + and ○.
- Lines show decision boundaries: Left – least-squares , Right – logistic-regression (Lecture 6)

H3 Fisher's Linear Discriminant: Intuition

For 2-classes, we can view a linear classification model in terms of ***dimensionality reduction***.

- First project onto a single dimension:

$$y = \mathbf{w}^T \mathbf{x}$$

- Then place a ***threshold*** on y and classify as class C_1 if $y \geq -w_0$ and as calss C_0 otherwise.
- Projection leads to a ***loss of information*** : classes well separated in D-dimensions may overlap in one dimension
- But, we can choose $\mathbf{w}$ to optimise that separation

H3 Example: Maximise Projected Mean Distance

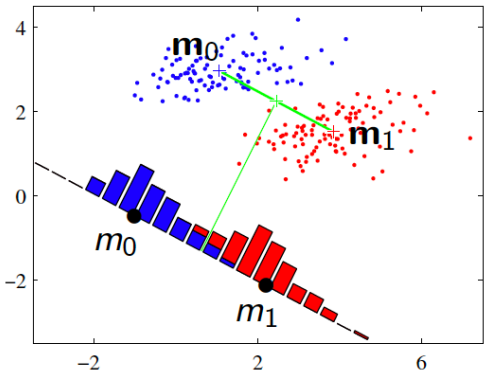
- Mean of class k is:

$$\boldsymbol{m}_k = \frac{\sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n}{N_k}$$

- Choose $\boldsymbol{w}$ to maximise distance between projected means:

$$m_1 - m_0 = \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_0)$$

where, $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$



- To avoid the magnitude of $\boldsymbol{w}$ affecting the result, we must restrict $\sum_i w_i^2 = 1$
- We still need to choose a threshold

> ***Notes***:
>
> A dot product between data-point, $\mathbf{x}_n \in \mathbb{R}^D$, and weight vector $\mathbf{w}$ can be thought of as projecting the D-dimensional data down onto a signle dimension, say $\mathbf{x}^T \mathbf{w} = \mathbf{w}^T \mathbf{x} = x_n \in \mathbb{R}$. The projected mean, m_k of the data-points in class C_k, can be therefore be thought of in two ways.
>
> First, as the projection of the D-dimensional mean, i.e.:
>
> $$m_k = \mathbf{w}^T \mathbf{m}_k = \mathbf{w}^T \left(\frac{\sum_{n \in C_K} \mathbf{x}_n}{N_k} \right)$$
>
> Second as the mean of the projected points, i.e.:
>
> $$m_k = \frac{\sum_{n \in C_K} \mathbf{w}^T \mathbf{x}_n}{N_k}$$
>
> where $x_n = \mathbf{w}^T \mathbf{x}_n$. Both are equivalent.
>
> Maiximising the distance between projected means $m_1 - m_0$ while simultaneously ensuring that the projection vector, $\mathbf{w}$, has unit length ($||\mathbf{w}|| = \sqrt{\mathbf{w}^T \mathbf{w}} = 1$) is an example of constrained optimisation.

> The diagram shows an example dataset with two classes, C_0 (blue) and C_1 (red). The projection vector $\mathbf{w}$ runs from $\mathbf{m}_0$ to $\mathbf{m}_1$ (along the green line). A histogram of the projected datapoints is shown with x-axis parallel to $\mathbf{w}$. Note that ***any resulting decision boundary will be orthogonal (at right angles to) the project vector.*** One example decision boundary would be along the green line running from the histogram to the projection vector.

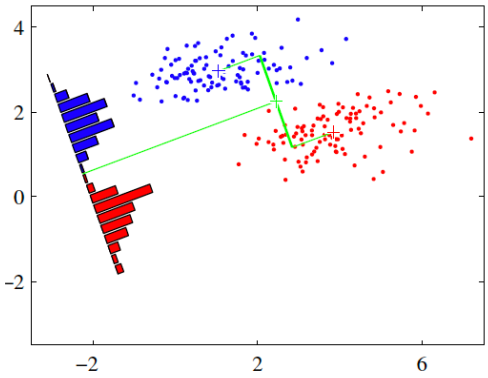
H3 Class Separation: A Better Way

A better way to maximise separation:

- Within-class variance of projected data is:

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

where $y_n = \boldsymbol{w}^T \boldsymbol{x}_n$



- Total within class variance $s_0^2 + s_1^2$
- Between class variance $(m_1 - m_0)^2$
- Instead maximise the Fisher criterion: $J(\boldsymbol{w}) = \dfrac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$

> ***Notes***:
>
> Maiximising Fisher's criterion now represents a trade-off between maximising the distance between the means and ensuring that the resulting projected data-points have a small ***within class variance***
>
> The diagram shows the same example dataset as the previous one, the new projection vector $\mathbf{w}$ points slightly more downwards (along the near vertical green line). A histogram of the projected datapoints is shown with x-axis parallel to $\mathbf{w}$.
>
> Any resulting decision boundary would again be orthogonal the projcetion vector, One example decision boundary would run along the near horizontal green line from histogram to data-points. This decision boundary shows ***a much better partition of the classes than was possible using the maximum mean distance approach***.