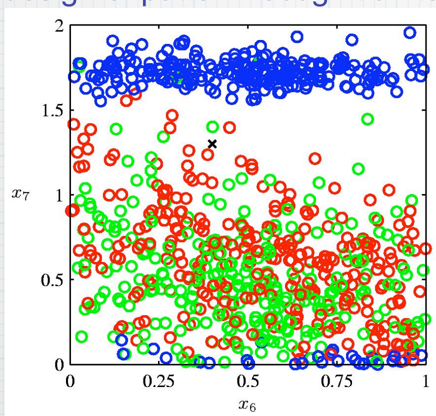
Curse of Dimensionality

- Real world applications deal with spaces with high dimensionality
- High dimensionality poses serious challenges for the design of pattern recognition techniques



Oil flaw data in two dimensions

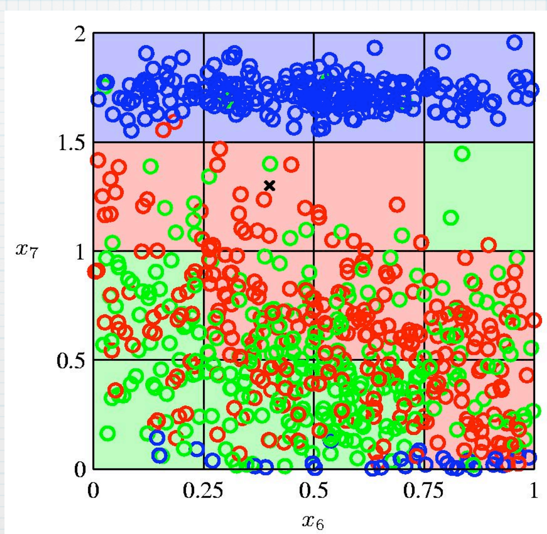
Red = homogeneous

Green = annular

Blue = laminar

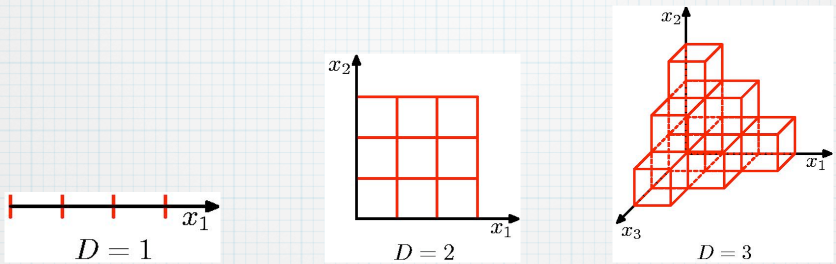
Curse of Dimensionality

- A simple classification approach



Curse of Dimensionality

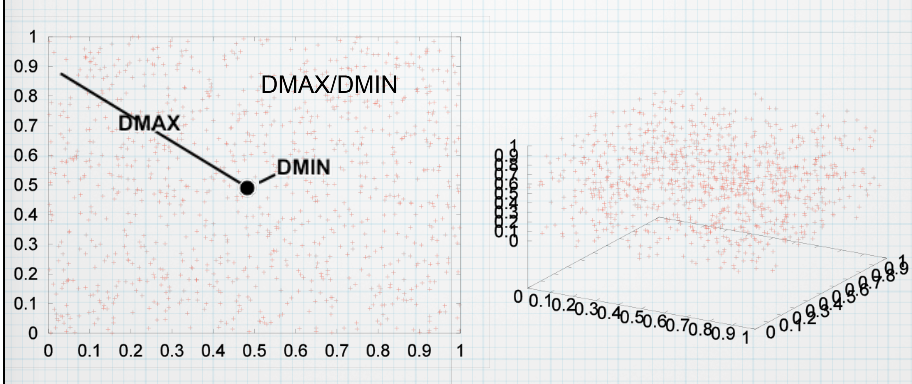
- Going higher in dimensionality…



3^1 3^2 3^3

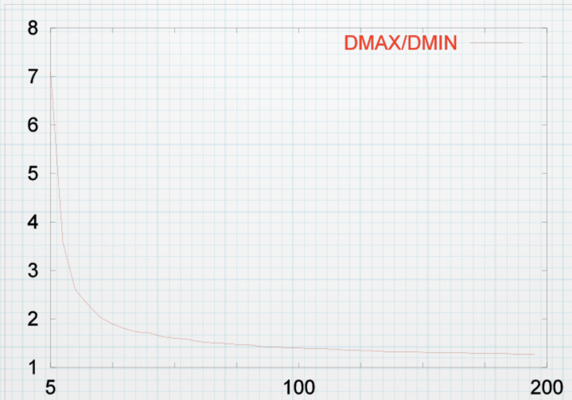
The number of regions of a regular grid grows exponentially with the dimensionality D of the space

Curse of Dimensionality



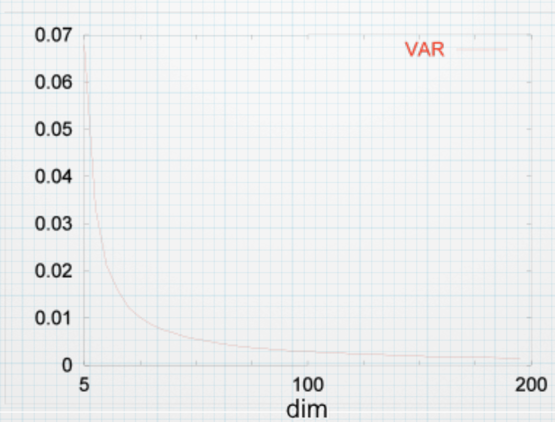
Sample of size *N=500* uniformly distributed in $[0,1]^q$

Curse of Dimensionality



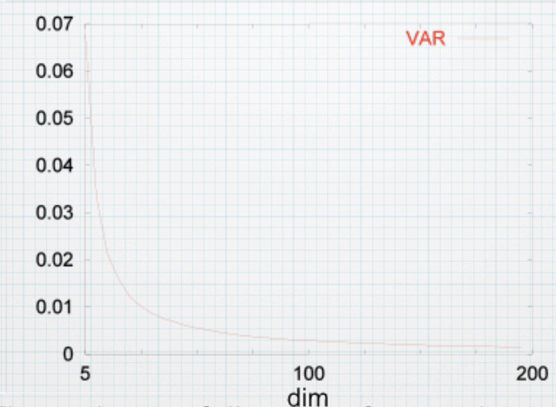
The distribution of the ratio **DMAX/DMIN** converges to **1** as the dimensionality increases

Curse of Dimensionality



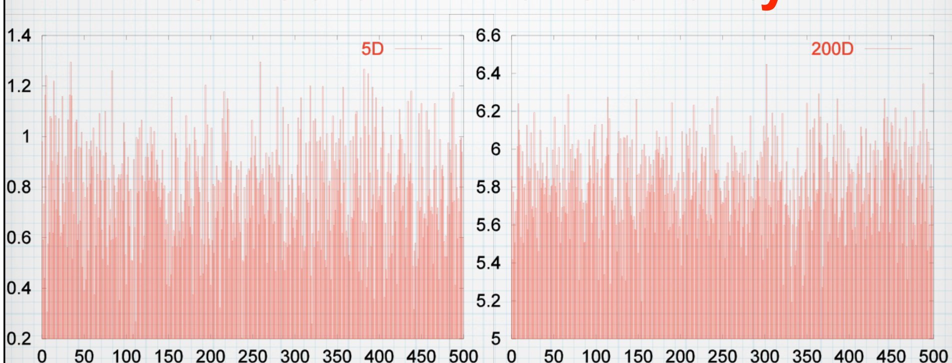
Variance of distances from a given point

Curse of Dimensionality



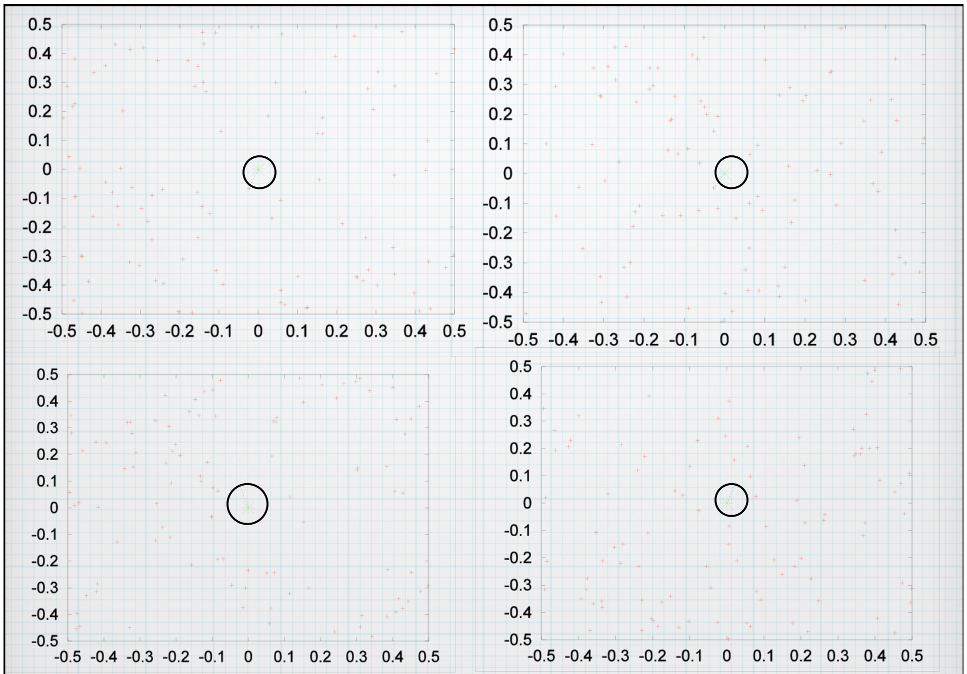
The variance of distances from a given point converges to **0** as the dimensionality increases

Curse of Dimensionality

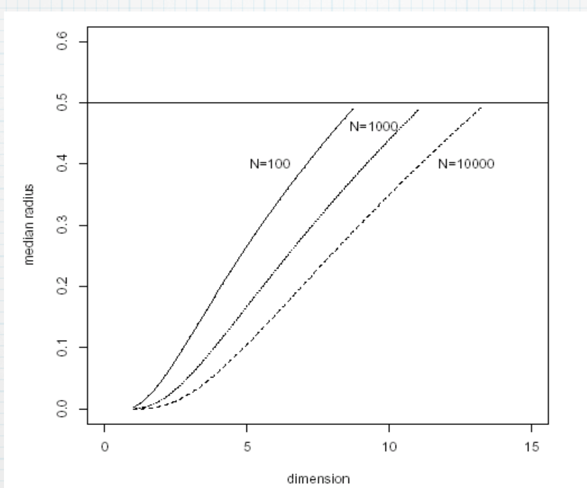


Distance values from a given point

Values flatten out as dimensionality increases



Computing radii of nearest neighborhoods



median radius of a nearest neighborhood

uniform distribution in the unit cube $\left[-.5,.5\right]^q$

Curse-of-Dimensionality

As dimensionality increases, the distance from the closest point increases faster

- Random sample of size $N \sim$ uniform distribution in the q-dimensional unit hypercube
- Diameter of a $K=1$ neighborhood using Euclidean distance: $d(q,N) = O(N^{-1/q})$

q	4	4	6	6	10	10	20	20	20
N	100	1000	100	1000	1000	10000	10000	10^6	10^{10}
d(q,N)	0.42	0.23	0.71	0.48	0.91	0.72	1.51	1.20	0.76

Large $d(q,N) \Rightarrow$ Highly biased estimations

Curse-of-Dimensionality

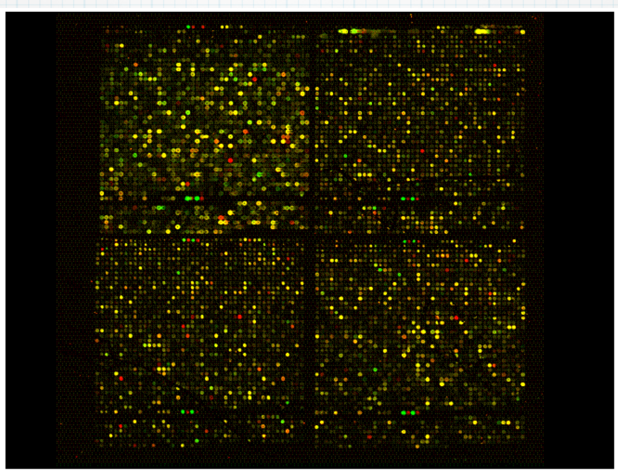
- In high dimensional spaces data become extremely sparse and are far apart from each other

- **The curse of dimensionality affects *any* estimation problem with high dimensionality**

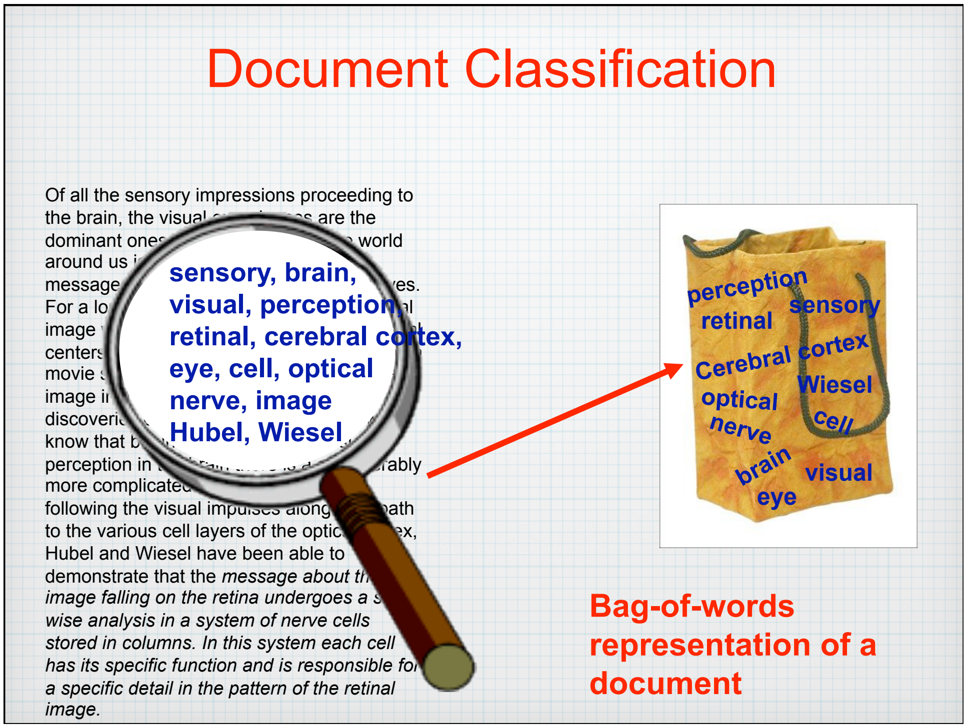
Curse-of-Dimensionality

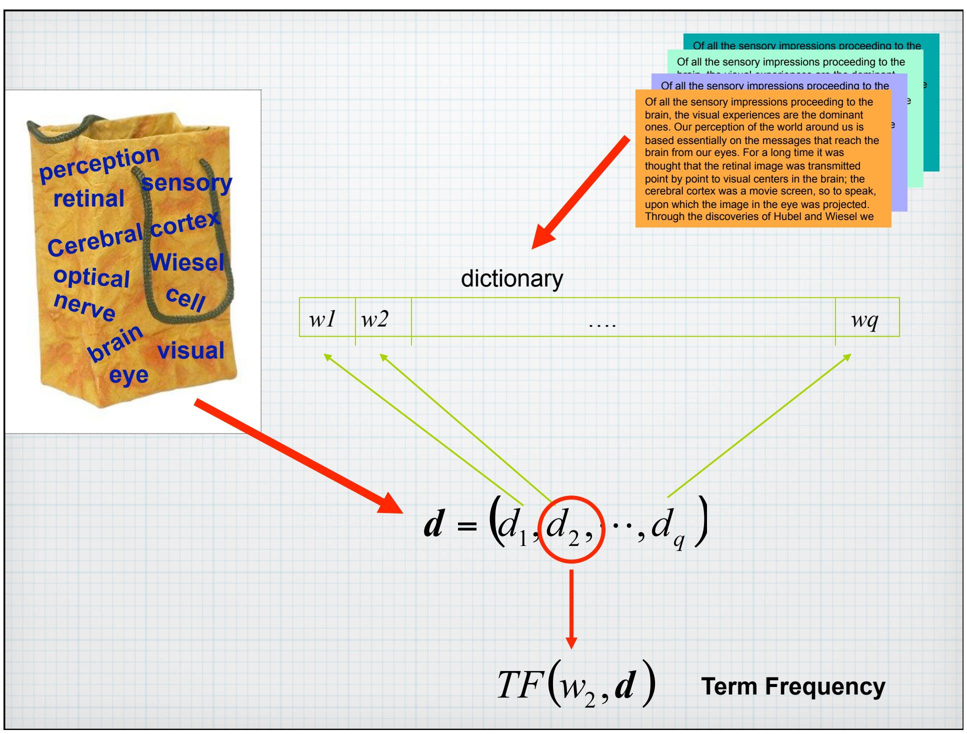
- It is a serious problem in many real-world applications
  - Microarray data: 3,000-4,000 genes;
  - Documents: 10,000-20,000 words in dictionary;
  - Images, face recognition, etc.

Microarray Data Analysis

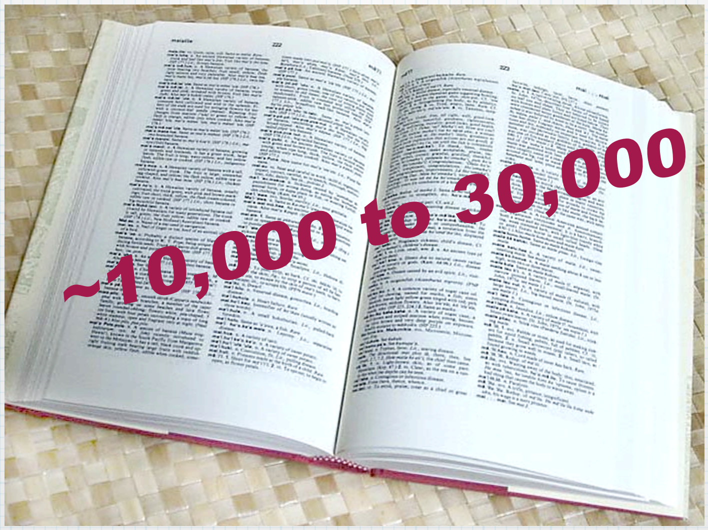


Problem: Which samples are most similar to each other, in terms of their expression profiles across genes


Document Classification
Of all the sensory impressions proceeding to
the brain, the visual
dominant ones
around us
message
For a lo
image
centers
movie
image i
discoveri
know that
perception in
more complicated
following the visual impulses along
to the various cell layers of the optic
Hubel and Wiesel have been able to
demonstrate that the message about th
image falling on the retina undergoes a s
wise analysis in a system of nerve cells
stored in columns. In this system each cell
has its specific function and is responsible fo
a specific detail in the pattern of the retinal
image.
sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel
perception
retinal
sensory
Cerebral cortex
optical
Wiesel
nerve
cell
brain
visual
eye
Bag-of-words representation of a document


Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the brain; the cerebral cortex was a movie screen, so to speak, upon which the image in the eye was projected. Through the discoveries of Hubel and Wiesel we
perception
retinal
sensory
Cerebral cortex
optical
Wiesel
nerve
cell
brain
visual
eye
dictionary
w1
w2
….
wq
$\boldsymbol{d} = (d_1, d_2, \cdots, d_q)$
$TF(w_2, \boldsymbol{d})$
Term Frequency

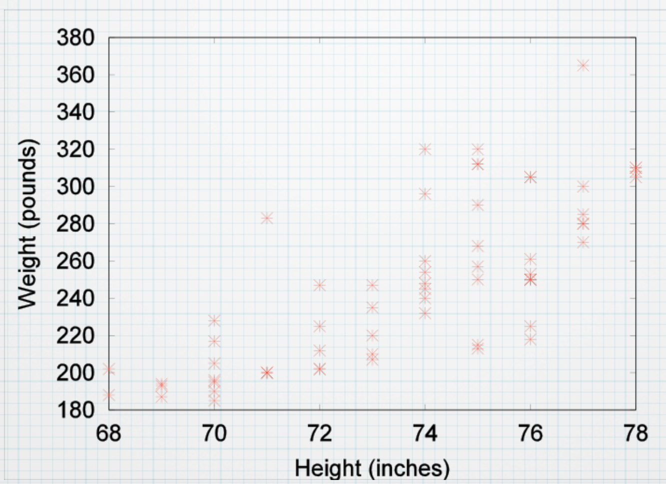
What's the size of the dictionary?



How can we deal with the curse of dimensionality?

Curse-of-Dimensionality

- Effective techniques applicable to high dimensional spaces exist.
- The reasons are twofold:
  - ✓ Real data are often confined to regions of *lower dimensionality*
  - ✓ Real data typically exhibit *smoothness properties* (at least locally). Local interpolation techniques can be used to make predictions



$$\begin{bmatrix} 7.68 & 92.2 \\ 92.2 & 1912.5 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

2×2 covariance matrix:

$$E\left[(\boldsymbol{x}-\mu)(\boldsymbol{x}-\mu)^T\right] =$$

$$E\left[\begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}(x_1-\mu_1, x_2-\mu_2)\right] =$$

$$E\begin{bmatrix} (x_1-\mu_1)^2 & (x_1-\mu_1)(x_2-\mu_2) \\ (x_1-\mu_1)(x_2-\mu_2) & (x_2-\mu_2)^2 \end{bmatrix} =$$

$$\frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} (x_1^i-\mu_1)^2 & (x_1^i-\mu_1)(x_2^i-\mu_2) \\ (x_1^i-\mu_1)(x_2^i-\mu_2) & (x_2^i-\mu_2)^2 \end{bmatrix}$$

$$\frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} (x_1^i-\mu_1)^2 & (x_1^i-\mu_1)(x_2^i-\mu_2) \\ (x_1^i-\mu_1)(x_2^i-\mu_2) & (x_2^i-\mu_2)^2 \end{bmatrix} =$$

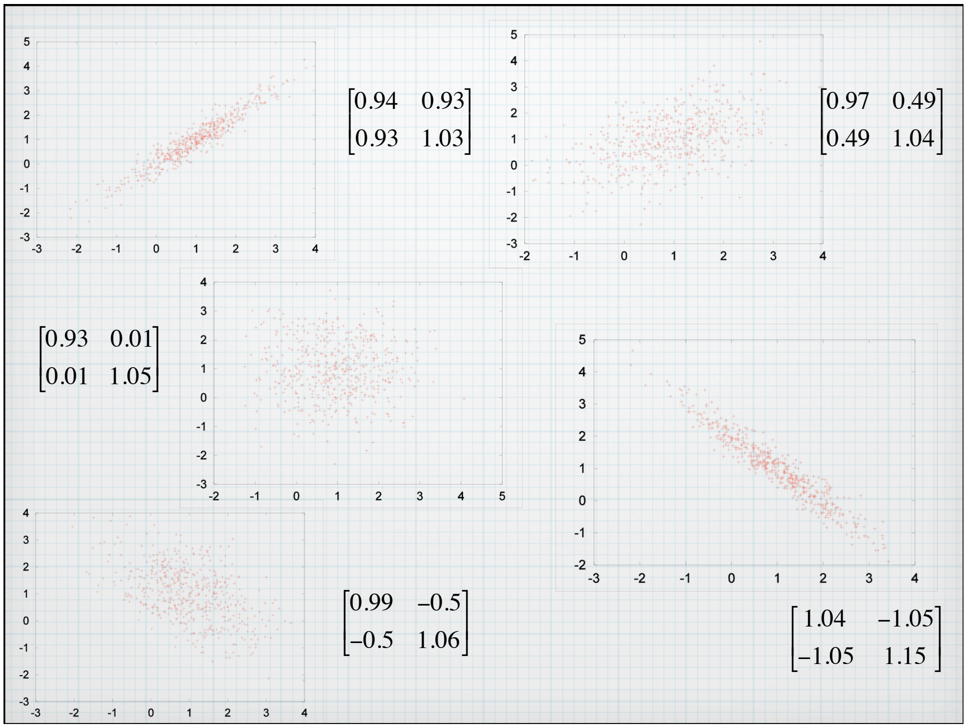
variance

covariance

$$\begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N}(x_1^i-\mu_1)^2 & \frac{1}{N}\sum_{i=1}^{N}\left[(x_1^i-\mu_1)(x_2^i-\mu_2)\right] \\ \frac{1}{N}\sum_{i=1}^{N}\left[(x_1^i-\mu_1)(x_2^i-\mu_2)\right] & \frac{1}{N}\sum_{i=1}^{N}(x_2^i-\mu_2)^2 \end{bmatrix}$$

covariance

variance



$$\begin{bmatrix} 0.94 & 0.93 \\ 0.93 & 1.03 \end{bmatrix} \quad \begin{bmatrix} 0.97 & 0.49 \\ 0.49 & 1.04 \end{bmatrix} \quad \begin{bmatrix} 0.93 & 0.01 \\ 0.01 & 1.05 \end{bmatrix} \quad \begin{bmatrix} 0.99 & -0.5 \\ -0.5 & 1.06 \end{bmatrix} \quad \begin{bmatrix} 1.04 & -1.05 \\ -1.05 & 1.15 \end{bmatrix}$$

Dimensionality Reduction

- Many dimensions are often interdependent (correlated);

We can:

- Reduce the dimensionality of problems;
- Transform interdependent coordinates into significant and independent ones;

Decision Theory

- *Decision theory*, when combined with *probability theory*, allows to make optimal decisions in situations involving uncertainty
- Training data: input vector $\boldsymbol{x}$, target vector $\boldsymbol{t}$
- Inference: joint probability distribution $p(\boldsymbol{x}, \boldsymbol{t})$
- Decision step: make optimal decision

Decision Theory

Classification example: medical diagnosis problem

- $\boldsymbol{x}$ set of pixel intensities in an image
- Two classes:
  - $C_1 = 0$ absence of cancer
  - $C_2 = 1$ presence of cancer
- Inference step: estimate $p(\boldsymbol{x}, C_k)$
- Decision step: given $\boldsymbol{x}$ predict C_k so that a measure of error is minimized according to the given probabilities

Decision Theory

How probabilities play a role in decision making?

- Decision step: given $\boldsymbol{x}$ predict C_k

Thus, we are interested in $p(C_k \mid \boldsymbol{x})$

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(\boldsymbol{x})}$$

Intuitively: we want to minimize the chance of assigning $\boldsymbol{x}$ to the wrong class. Thus, choose the class that gives the higher posterior probability

Minimizing the misclassification rate

- Goal: Minimize the number of misclassifications

We need to find a rule that assigns each input vector to one of the possible classes C_k

Such rule divides the input space into regions R_k so that all points in R_k are assigned to C_k

Boundaries between regions are called ***decision boundaries***

Minimizing the misclassification rate

- Goal: Minimize the number of misclassifications

$$p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$
$$= \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx$$

- Assign $\boldsymbol{x}$ to the class that gives the smaller value of the integrand:
  - Choose C_1 if $p(\boldsymbol{x}, C_1) > p(\boldsymbol{x}, C_2)$
  - Choose C_2 if $p(\boldsymbol{x}, C_2) > p(\boldsymbol{x}, C_1)$

Minimizing the misclassification rate

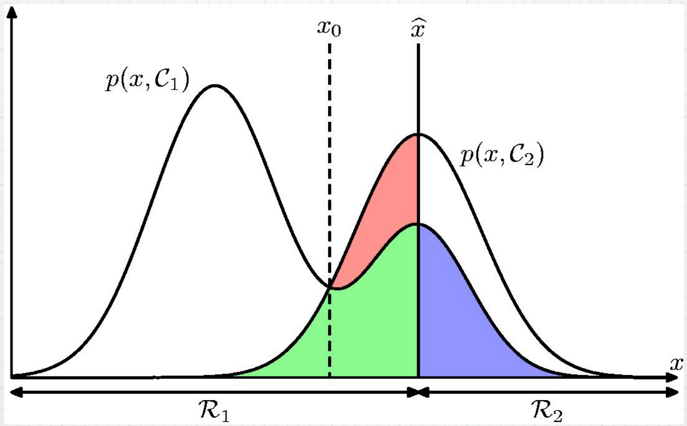
- Choose C_1 if $p(\boldsymbol{x}, C_1) > p(\boldsymbol{x}, C_2)$
- Choose C_2 if $p(\boldsymbol{x}, C_2) > p(\boldsymbol{x}, C_1)$

$$p(\boldsymbol{x}, C_k) = p(C_k \mid \boldsymbol{x})p(\boldsymbol{x})$$

Thus:

- Choose C_1 if $p(C_1 \mid \boldsymbol{x}) > p(C_2 \mid \boldsymbol{x})$
- Choose C_2 if $p(C_2 \mid \boldsymbol{x}) > p(C_1 \mid \boldsymbol{x})$

Minimizing the misclassification rate



Optimal decision boundary: $\hat{x} = x_0$

Minimizing the misclassification rate

General case of K classes:

$$p(correct) = \sum_{k=1}^{K} p(\mathbf{x} \in R_k, C_k) = \sum_{k=1}^{K} \int_{R_k} p(\mathbf{x}, C_k) d\mathbf{x}$$

Thus:

Choose C_k that gives the largest $p(C_k \mid x)$

Minimizing the expected loss

- Some mistakes are more costly than others.
- ***Loss function*** (***cost function***): overall measure of loss incurred in taking any of the available decisions

L_{kj} : loss incurred when we assign $\boldsymbol{x}$ to class C_j and the true class is C_k

$$\begin{array}{c|cc} & \text{cancer} & \text{normal} \\ \hline \text{cancer} & 0 & 1000 \\ \text{normal} & 1 & 0 \end{array}$$

The optimal solution is the one that minimizes the loss function

Minimizing the expected loss

- The loss function depends on the true class, which is unknown.
- The uncertainty of the true class is expressed through the joint probability $p(\boldsymbol{x}, C_k)$
- We minimize the expected loss:

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

- For each $\boldsymbol{x}$ we should minimize

$$\sum_k L_{kj}\, p(\boldsymbol{x}, C_k) = \sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})\, p(\boldsymbol{x})$$

Minimizing the expected loss

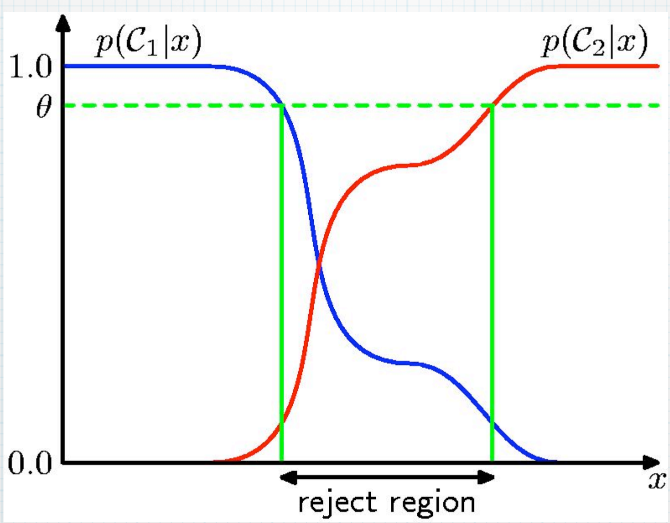
- For each $\boldsymbol{x}$ we should minimize

$$\sum_k L_{kj}\, p(\boldsymbol{x}, C_k) = \sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})\, p(\boldsymbol{x})$$

- Thus, to minimize the expected loss: Assign each $\boldsymbol{x}$ to the class j that minimizes

$$\sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})$$

The Reject Option



Inference and Decision

- ***Inference stage***: use the training data to learn a model for $p(C_k \mid \boldsymbol{x})$

- ***Decision stage***: use the given posterior probabilities to make optimal class assignments

Generative Methods

- Solve the inference problem of estimating the class-conditional densities $p(\boldsymbol{x} \mid C_k)$ for each class C_k
- Infer the prior class probabilities $p(C_k)$
- Use Bayes' theorem to find the class posterior probabilities:

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(\boldsymbol{x})}$$

where $$p(\boldsymbol{x}) = \sum_k p(\boldsymbol{x} \mid C_k)p(C_k)$$

- Use decision theory to determine class membership for each new input $\boldsymbol{x}$

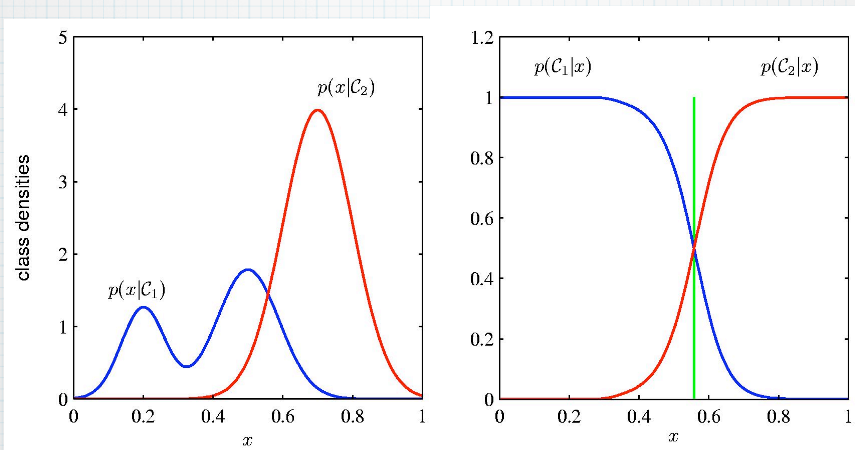
Discriminative Methods

- Solve directly the inference problem of estimating the class posterior probabilities $p(C_k \mid \boldsymbol{x})$
- Use decision theory to determine class membership for each new input $\boldsymbol{x}$

Discriminant Functions

- Find a function $f(\boldsymbol{x})$ which maps each input directly onto a class label. Probabilities play no role here.
- Use decision theory to determine class membership for each new input $\boldsymbol{x}$

Example



Linear Models for Classification

- ***Classification***: Given an input vector $\boldsymbol{x}$, assign it to one of K classes C_k where $k = 1, \ldots, K$

- The input space is divided in ***decision regions*** whose boundaries are called ***decision boundaries*** or ***decision surfaces***

- ***Linear models***: decision surfaces are linear functions of the input vector $\boldsymbol{x}$. They are defined by $(D-1)$-dimensional hyperplanes within the D-dimensional input space

Linear Models for Classification

- For regression: $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$

- For classification, we want to predict class labels, or more generally class posterior probabilities.

- We transform the linear function of $\boldsymbol{w}$ using a nonlinear function $f()$ so that

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

Generalized Linear Models

Linear Discriminant Functions

Two classes:

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

$$\begin{array}{ll} \textit{if} \;\; y(\boldsymbol{x}) \geq 0 & \textit{assign} \;\; \boldsymbol{x} \;\; \textit{to} \;\; C_1 \\ \textit{otherwise} & \textit{assign} \;\; \boldsymbol{x} \;\; \textit{to} \;\; C_2 \end{array}$$

Decision boundary: $y(\boldsymbol{x}) = 0$

Linear Discriminant Functions

Geometrical properties:

Decision boundary: $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0$

Let $\boldsymbol{x}_1, \boldsymbol{x}_2$ be two points which lie on the decision boundary

$$y(\boldsymbol{x}_1) = \boldsymbol{w}^T \boldsymbol{x}_1 + w_0 = 0, y(\boldsymbol{x}_2) = \boldsymbol{w}^T \boldsymbol{x}_2 + w_0 = 0$$
$$\Rightarrow \boldsymbol{w}^T (\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$$

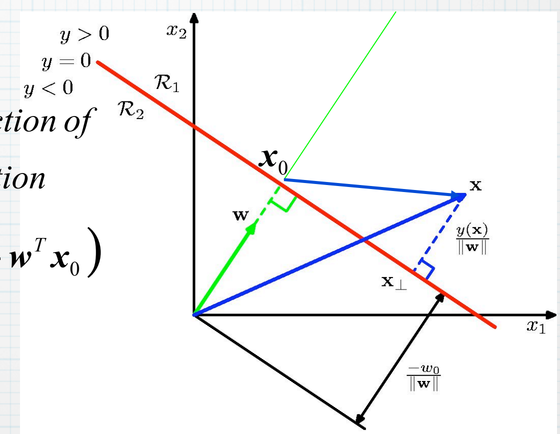
$\boldsymbol{w}$ represents the orthogonal direction to the decision boundary

Geometrical properties (con't)

$$\boldsymbol{w}^{*T} = \frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|}$$

$\boldsymbol{w}^{*T}(\boldsymbol{x} - \boldsymbol{x}_0)$ *is the projection of* $(\boldsymbol{x} - \boldsymbol{x}_0)$ *onto the* w^* *direction*

$$\frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|}(\boldsymbol{x} - \boldsymbol{x}_0) = \frac{1}{\|\boldsymbol{w}\|}\left(\boldsymbol{w}^T \boldsymbol{x} - \boldsymbol{w}^T \boldsymbol{x}_0\right)$$
$$= \frac{1}{\|\boldsymbol{w}\|}\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right) = \frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|}$$



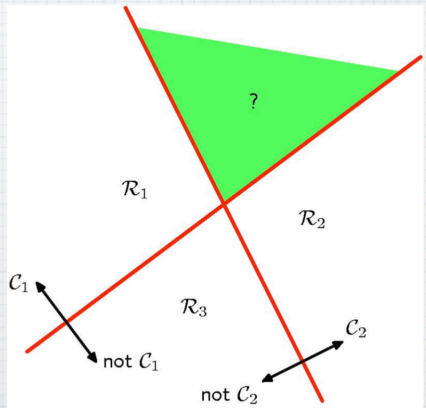
when $\boldsymbol{x} = \boldsymbol{0}$, $\frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|} = \frac{w_0}{\|\boldsymbol{w}\|}$

Signed orthogonal distance of the origin from the decision surface

Linear Discriminant Functions

Multiple classes

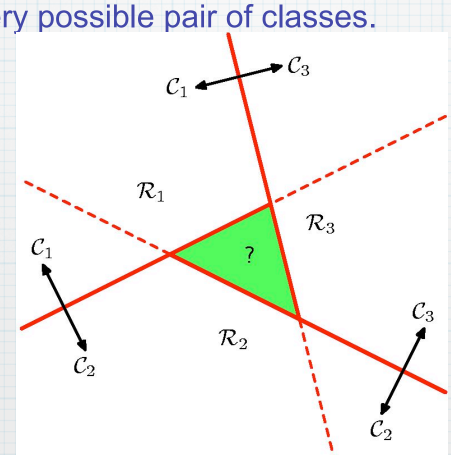
one-versus-the-rest: K-1 classifiers each of which solves a two-class problem of separating points of C_k from points not in that class



Linear Discriminant Functions

Multiple classes

one-versus-one: $K(K$-1)/2 binary discriminant functions, one for every possible pair of classes.



Linear Discriminant Functions

Multiple classes

Solution: consider a single *K-class* discriminant comprising *K* linear functions of the form

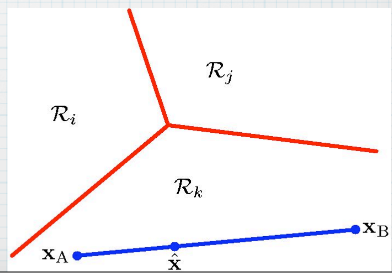
$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

Assign a point $\boldsymbol{x}$ to class C_k if $y_k(x) > y_j(x) \; \forall j \neq k$

The decision boundary between class C_k and class C_j is given by

$$y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$$

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



Linear Discriminant Functions

Two approaches:

- ***Fisher's linear discriminant***
- ***Perceptron algorithm***

Fisher's Linear Discriminant

One way to view a linear classification model is in terms of *dimensionality reduction*.

Two class case:

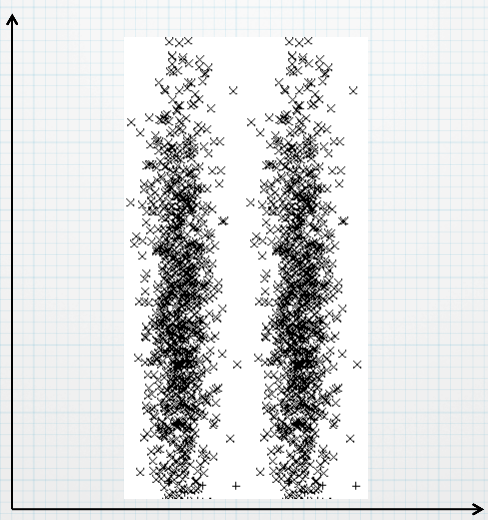
Suppose we project $\boldsymbol{x}$ onto one dimension:

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

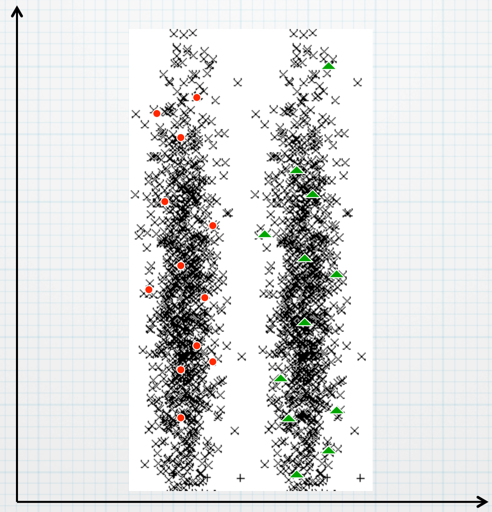
Set a threshold t

$$\begin{aligned} &if \;\; y \geq t \quad assign \;\; \boldsymbol{x} \;\; to \;\; C_1 \\ &otherwise \quad assign \;\; \boldsymbol{x} \;\; to \;\; C_2 \end{aligned}$$

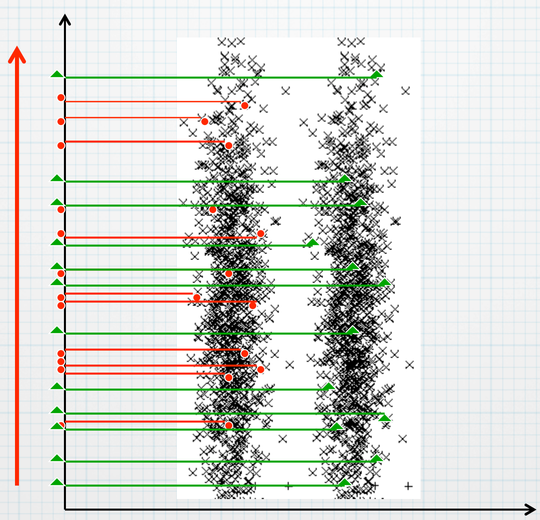
Example



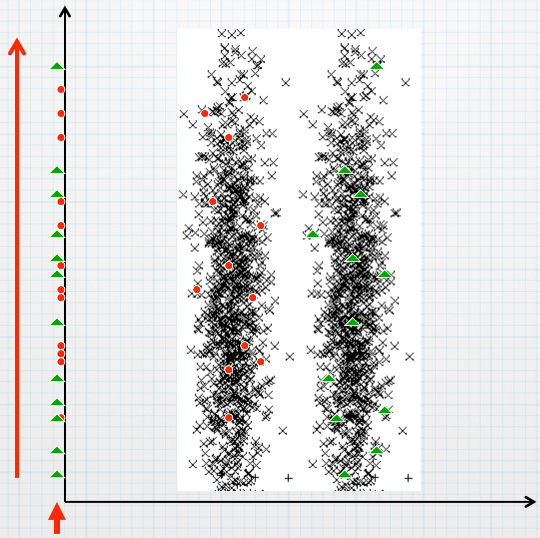
Example



Example

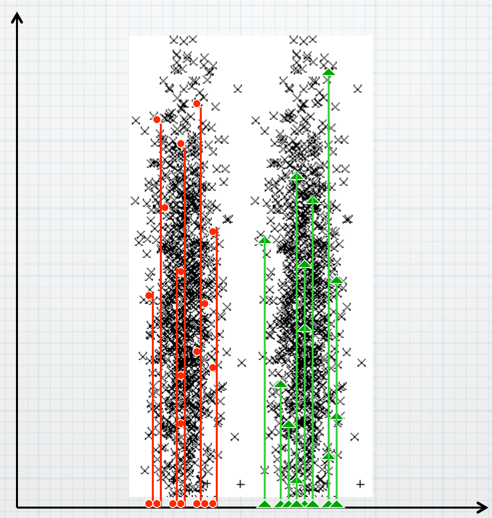


Example

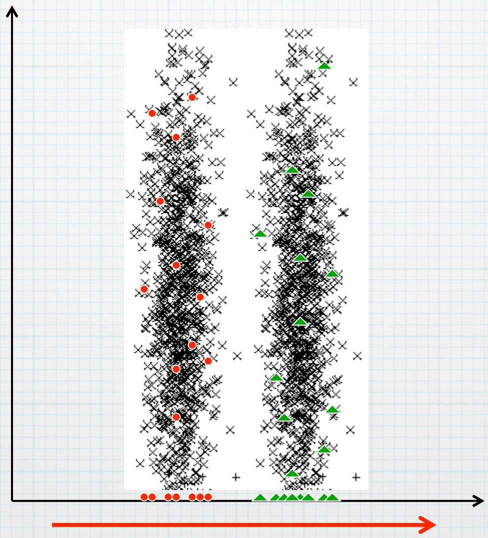


Confused mixture of samples from both classes

How about the horizontal dimension?



How about the horizontal dimension?



Projected samples are well separated now

Fisher's Linear Discriminant

- Find an orientation along which the projected samples are well separated;
- This is exactly the goal of linear discriminant analysis (LDA);
- In other words: we are after the linear projection that best separates the data, i.e. best discriminates data of different classes.

How can we find such discriminant direction?

LDA

$\{(\boldsymbol{x}_n, C_i)\}_{i=1}^{N}$ $\quad \boldsymbol{x}_n \in \Re^q$ $\quad C_i \in \{C_1, C_2\}$

- N_1 samples of class C_1
- N_2 samples of class C_2
- Consider $\boldsymbol{w} \in \Re^q$ with $\|\boldsymbol{w}\| = 1$
- Then: $\boldsymbol{w}^T \boldsymbol{x}$ is the projection of $\boldsymbol{x}$ along the direction of $\boldsymbol{w}$
- We want the projections $\boldsymbol{w}^T \boldsymbol{x}$ where $\boldsymbol{x} \in C_1$ separated from the projections $\boldsymbol{w}^T \boldsymbol{x}$ where $\boldsymbol{x} \in C_2$

LDA

- A measure of the separation between the projected points is the difference of the sample means:

$$\boldsymbol{m}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}$$ Sample mean of class C_i

$$m_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{w}^T \boldsymbol{x} = \boldsymbol{w}^T \boldsymbol{m}_i$$ Sample mean for the projected points

$$\Rightarrow \quad |m_1 - m_2| = |\boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)|$$

We wish to make the above difference as large as we can. In addition…

LDA

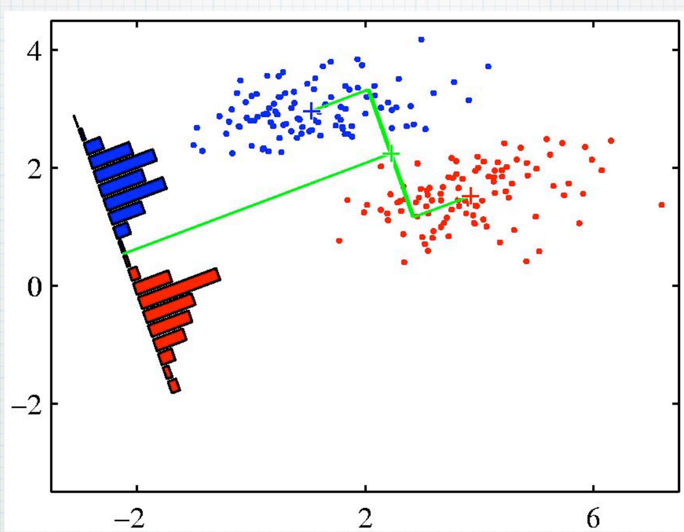
- To obtain good separation of the projected data we really want the difference between the means to be large relative to some measure of the standard deviation of each class:

$$s_i^2 = \sum_{x \in C_i} \left(\boldsymbol{w}^T \boldsymbol{x} - m_i \right)^2$$ Scatter for the projected samples of class C_i

$$s_1^2 + s_2^2$$ Total **within-class scatter** of the projected samples

$$\arg\max_{w} \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}$$ **Fisher linear discriminant analysis**

LDA



LDA

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

To obtain $J(\boldsymbol{w})$ as an explicit function of $\boldsymbol{w}$ we define the following matrices :

$$S_i = \sum_{x \in C_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

$S_W = S_1 + S_2$ **Within-class scatter matrix**

Then:

$$s_i^2 = \sum_{x \in C_i} \left(\boldsymbol{w}^T \boldsymbol{x} - m_i\right)^2 = \sum_{x \in C_i} \left(\boldsymbol{w}^T \boldsymbol{x} - \boldsymbol{w}^T \boldsymbol{m}_i\right)^2$$

$$= \sum_{x \in C_i} \boldsymbol{w}^T (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T \boldsymbol{w} = \boldsymbol{w}^T S_i \boldsymbol{w}$$

LDA

So : $s_1^2 = \boldsymbol{w}^T S_1 \boldsymbol{w}$ and $s_2^2 = \boldsymbol{w}^T S_2 \boldsymbol{w}$

Thus : $s_1^2 + s_2^2 = \boldsymbol{w}^T S_1 \boldsymbol{w} + \boldsymbol{w}^T S_2 \boldsymbol{w} =$

$$\boldsymbol{w}^T (S_1 + S_2) \boldsymbol{w} = \boldsymbol{w}^T S_W \boldsymbol{w}$$

Similarly :

$$(m_1 - m_2)^2 = \left(\boldsymbol{w}^T \boldsymbol{m}_1 - \boldsymbol{w}^T \boldsymbol{m}_2\right)^2 =$$

$$\boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} =$$

$$\boldsymbol{w}^T S_B \boldsymbol{w}$$

where $S_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$ **Between-class scatter matrix**

LDA

We have obtained :

$$s_1^2 + s_2^2 = \boldsymbol{w}^T S_W \boldsymbol{w}$$

$$(m_1 - m_2)^2 = \boldsymbol{w}^T S_B \boldsymbol{w}$$

$$\Rightarrow J(\boldsymbol{w}) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

$$\arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

LDA

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

$J(\boldsymbol{w})$ *is* max*imized when* $(\boldsymbol{w}^T S_B \boldsymbol{w}) S_W \boldsymbol{w} = (\boldsymbol{w}^T S_W \boldsymbol{w}) S_B \boldsymbol{w}$

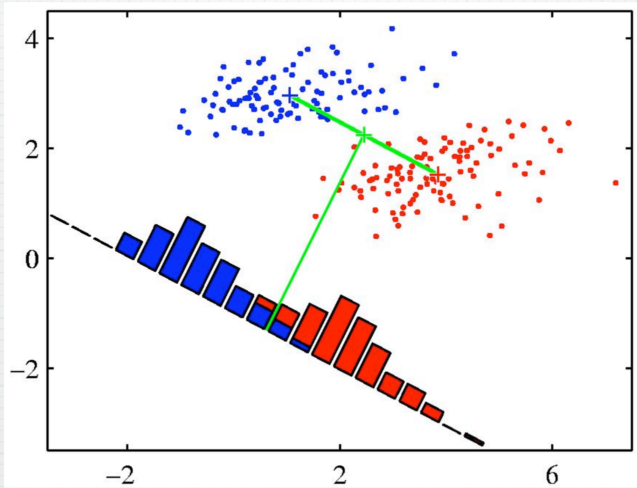
We observe that :

$$S_B \boldsymbol{w} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)\underbrace{(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}}_{\text{scalar}}$$

Always in the direction of $(\boldsymbol{m}_1 - \boldsymbol{m}_2)$

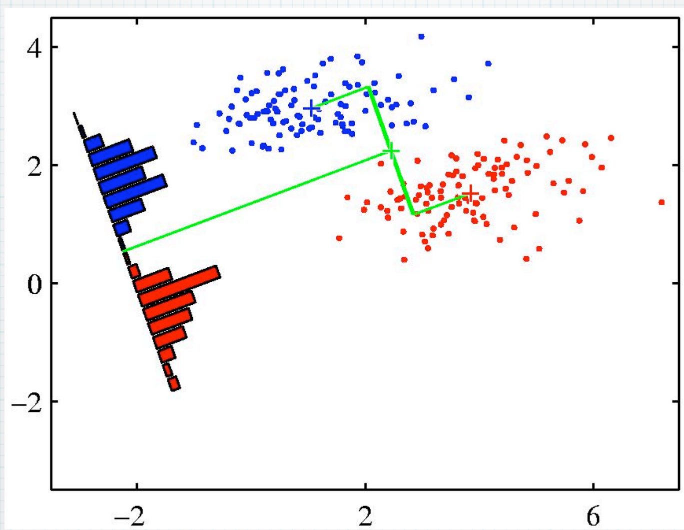
$$\Rightarrow \boxed{\boldsymbol{w} = S_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)}$$

LDA



Projection onto the line joining the class means

LDA



Solution of LDA

LDA

$$w = S_W^{-1}(m_1 - m_2)$$

• Gives the linear function with the maximum ratio of between-class scatter to within-class scatter.

• The problem, e.g. classification, has been reduced to a *q*-dimensional problem to a more manageable one-dimensional problem.

• Optimal for multivariate normal class conditional densities.

LDA

• The analysis can be extended to multiple classes.

• LDA is a ***linear*** technique for dimensionality reduction: it projects the data along directions that can be expressed as ***linear combination*** of the input features.

• The "appropriate" transformation depends on the data and on the **task** we want to perform on the data. Note that LDA uses class labels.

• Non-linear extensions of LDA exist (e.g., generalized LDA).