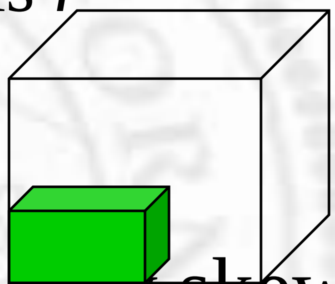
Dimension Reduction

Dimension Reduction

- Curse of dimensionality
  - with 50 features (dimensions), each quantized to 20 levels, create 20^{50} possible feature combinations, imagine how many samples you need to estimate $p(\mathbf{x}|w)$?
  - how do you visualize the structure in a 50 dimensional space?

Other problems

- Size of the local regions needed for density estimation getting larger and larger
  - To capture *r%* of the data, edge length is $r^{1/n}$
    - *n=10, r=0.01, x =0.63,*
    - *n=10, r=0.1, x=0.8*



- Data tend to boundary, creating boundary skew
  - Consider uniform distribution, *p*% interior
  - Exterior probability is $1-p^n$
    - *n=10, p=0.8, 0.89 exterior*
    - *N=100, p=0.8, 0.999.. exterior*

Solutions - Reduction

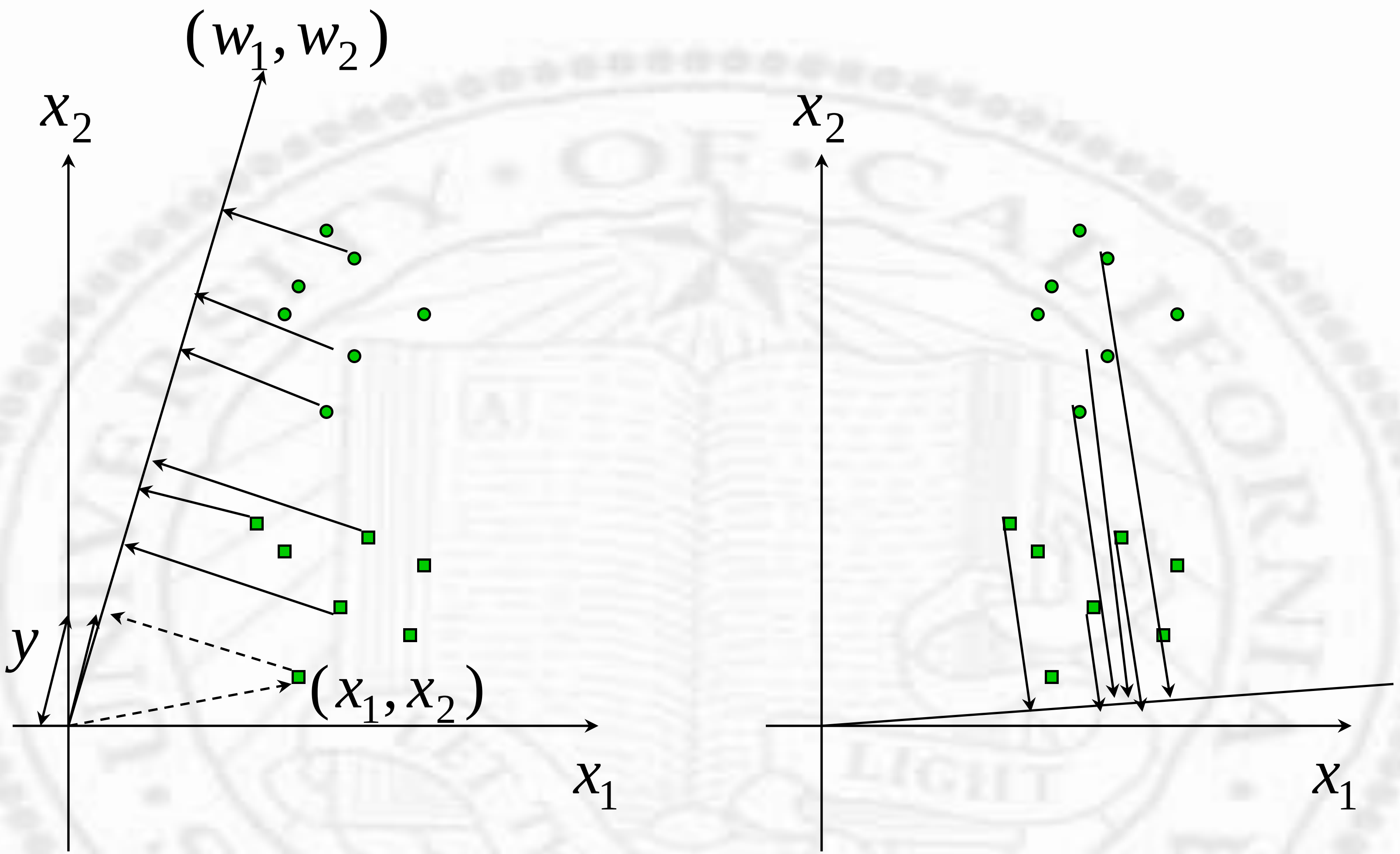
- Fisher’s linear discriminant
  - Preserve class separation (special case of principle component analysis)
- Multi-dimensional scaling
  - Preserve distance measures
- Principal component analysis
  - Best data *representation* (not necessarily best class *separation*)

Fisher's linear discriminant (2-class)

- Given n d-dimensional samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

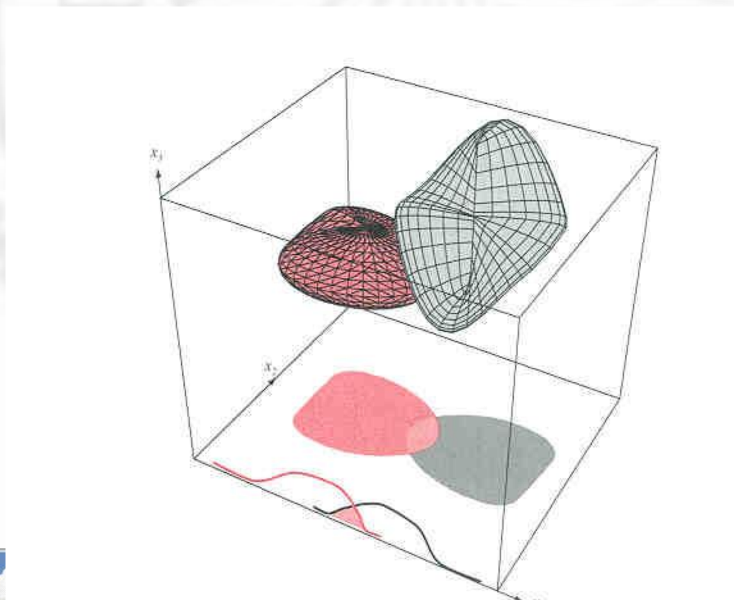
  $\mathbf{X}_1 \in \varpi_1, |\mathbf{X}_1| = n_1 \;\; \mathbf{X}_2 \in \varpi_2, |\mathbf{X}_2| = n_2 \;\; n_1 + n_2 = n$

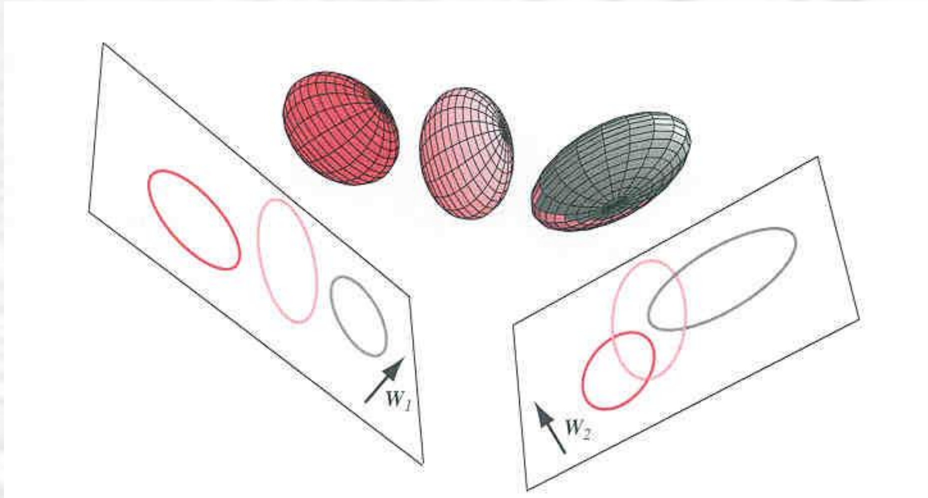
- a linear transform $y = \mathbf{w}^t \mathbf{x}$ which
  - maps d-D samples onto a line
  - best preserves class separation
- Intuitively, *good features are those with large separation of means relative to variances*


(w_1, w_2)
x_2
y
(x_1, x_2)
x_1
x_2
x_1

Caveats

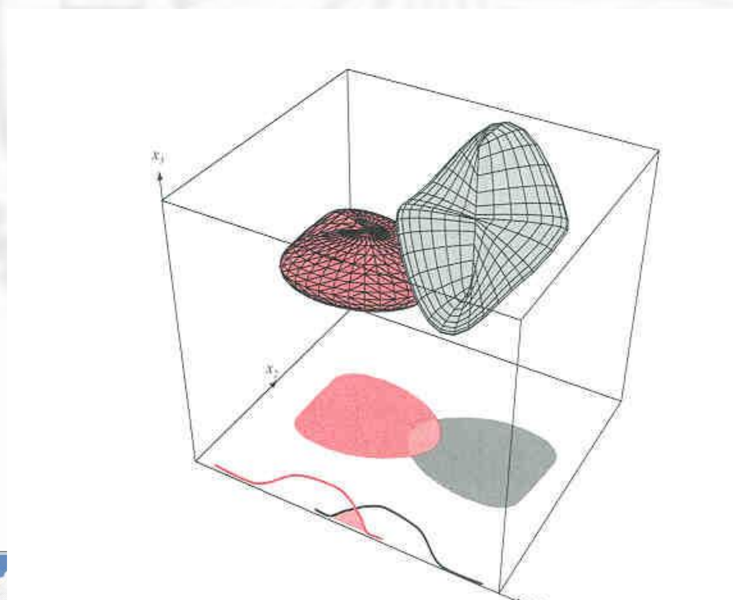
- The nature of the problem is that ambiguity might arise when you reduce problem dimension (a good reduction algorithm may minimize the problem, but may not completely eliminate the problem)

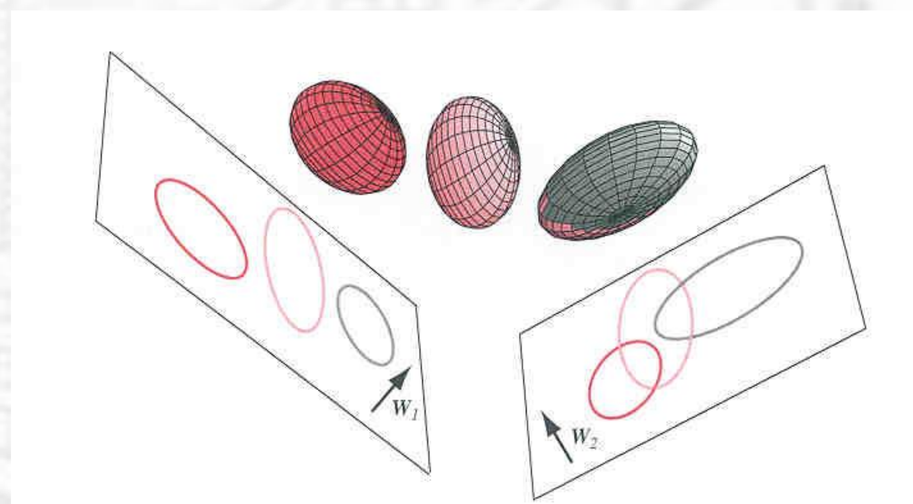




Caveats (cont)

- The figures also suggest that, sometimes, to get better performance, it is necessary to *increase* the dimension (more features), not to *decrease* it





In the original d-dimensional space

- Between class scatter

$$|\mathbf{m}_1 - \mathbf{m}_2|^2 \quad \mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathbf{X}_i} \mathbf{x}$$

- Within class scatter

$$s_1^{\ 2} + s_2^{\ 2} \quad s_i^{\ 2} = \sum_{\mathbf{x} \in \mathbf{X}_i} (\mathbf{x} - \mathbf{m}_i)^t (\mathbf{x} - \mathbf{m}_i)$$

- Ideally, function should be large

$$\frac{|\mathbf{m}_1 - \mathbf{m}_2|^2}{s_1^{\ 2} + s_2^{\ 2}}$$

In the transformed 1-dimensional space

- Between class scatter

$$|\hat{m}_1 - \hat{m}_2|^2 \quad \hat{m}_i = \frac{1}{n_i}\sum y = \frac{1}{n_i}\sum_{x \in \aleph_i} \mathbf{w^t x} = \mathbf{w^t m}_i$$

- Within class scatter

$$\hat{s}_1^{\,2} + \hat{s}_2^{\,2} \quad \hat{s}_i^{\,2} = \sum (y - \hat{m}_i)^2$$

- Ideally, function should be large

$$F(\mathbf{w}) = \frac{|\hat{m}_1 - \hat{m}_2|^2}{\hat{s}_1^{\,2} + \hat{s}_2^{\,2}}$$

- Or

$$|\hat{m}_1 - \hat{m}_2|^2 = (\mathbf{w}^{\mathbf{t}}\mathbf{m}_1 - \mathbf{w}^{\mathbf{t}}\mathbf{m}_2)^2$$

$$= \mathbf{w}^{\mathbf{t}}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^{\mathbf{t}}\mathbf{w} = \mathbf{w}^{\mathbf{t}}\mathbf{S}_b\mathbf{w}$$

$$\hat{s}_i^{\,2} = \sum (y - \hat{m}_i)^2 = \sum_{\mathbf{x}\in\mathbf{X}_i} (\mathbf{w}^{\mathbf{t}}\mathbf{x} - \mathbf{w}^{\mathbf{t}}\mathbf{m}_i)^2$$

$$= \sum_{\mathbf{x}\in\mathbf{X}_i} \mathbf{w}^{\mathbf{t}}(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathbf{t}}\mathbf{w} = \mathbf{w}^{\mathbf{t}}\mathbf{S}_i\mathbf{w}$$

$$\hat{s}_1^{\,2} + \hat{s}_2^{\,2} = \mathbf{w}^{\mathbf{t}}(\mathbf{S}_1 + \mathbf{S}_2)\mathbf{w} = \mathbf{w}^{\mathbf{t}}\mathbf{S}_w\mathbf{w}$$

$$F(w) = \frac{|\hat{m}_1 - \hat{m}_2|^2}{\hat{s}_1^{\,2} + \hat{s}_2^{\,2}} = \frac{\mathbf{w}^{\mathbf{t}}\mathbf{S}_{\mathbf{B}}\mathbf{w}}{\mathbf{w}^{\mathbf{t}}\mathbf{S}_{\mathbf{w}}\mathbf{w}}$$

The Analysis

- F(w): generalized Rayleigh quotient $\frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{w}}$
- To maximize F(w), w is the generalized eigenvector associated with the largest generalized eigenvalue

$$\mathbf{S_B w} = \lambda \mathbf{S}_w \mathbf{w} \quad or$$

$$\mathbf{S_w}^{-1}\mathbf{S_B w} = \lambda \mathbf{w}$$

$$\mathbf{w} = \mathbf{S_w}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

- Proof:

$$F(\mathbf{w}) = \frac{|\hat{m}_1 - \hat{m}_2|^2}{\hat{s}_1^{\,2} + \hat{s}_2^{\,2}} = \frac{\mathbf{w}^{\mathbf{t}}\mathbf{S}_B\mathbf{w}}{\mathbf{w}^{\mathbf{t}}\mathbf{S}_w\mathbf{w}}$$

$$\frac{dF(\mathbf{w})}{d\mathbf{w}} = \frac{2\mathbf{S}_B\mathbf{w}}{\mathbf{w}^{\mathbf{t}}\mathbf{S}_w\mathbf{w}} - \frac{2\mathbf{S}_w\mathbf{w}}{\mathbf{w}^{\mathbf{t}}\mathbf{S}_w\mathbf{w}} \boxed{\frac{\mathbf{w}^{\mathbf{t}}\mathbf{S}_B\mathbf{w}}{\mathbf{w}^{\mathbf{t}}\mathbf{S}_w\mathbf{w}}} = 0$$

$$2\mathbf{S}_B\mathbf{w}^* - \lambda 2\mathbf{S}_{\mathbf{w}}\mathbf{w}^* = 0 \qquad \boxed{\lambda = \frac{\mathbf{w}^{*\mathbf{t}}\mathbf{S}_B\mathbf{w}^*}{\mathbf{w}^{*\mathbf{t}}\mathbf{S}_w\mathbf{w}^*}}$$

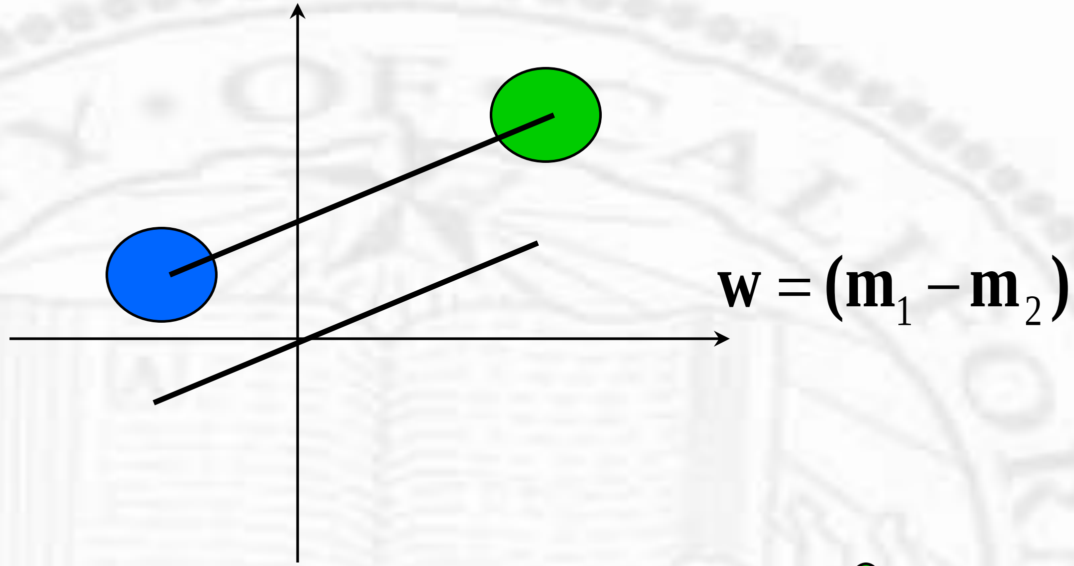
$$\mathbf{S}_B\mathbf{w}^* = \lambda\mathbf{S}_{\mathbf{w}}\mathbf{w}^*$$

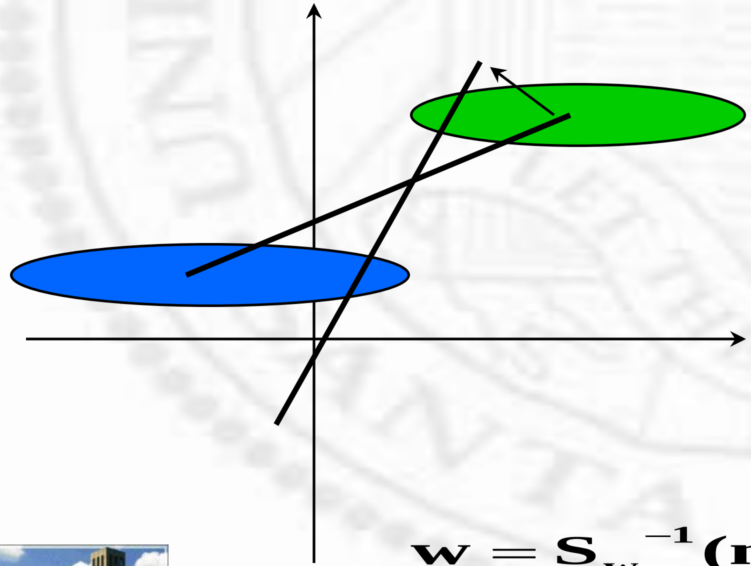
$$\mathbf{S}_{\mathbf{w}}^{-1}\mathbf{S}_B\mathbf{w}^* = \lambda\mathbf{w}^*$$

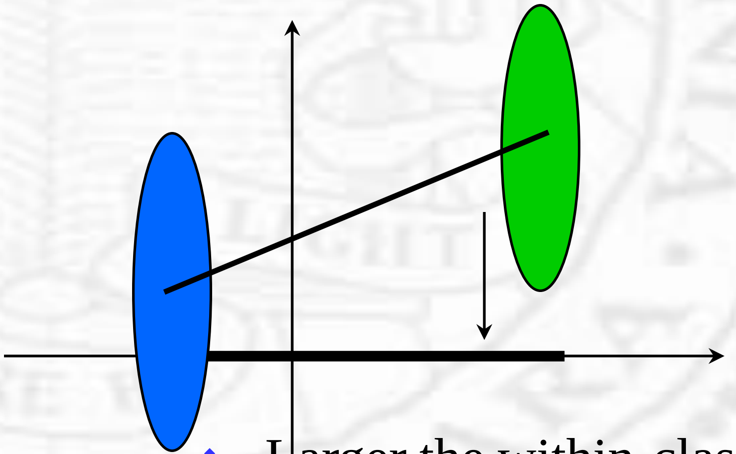
$$\mathbf{w}^* = \mathbf{S}_{\mathbf{w}}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

$$\because \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \rightarrow \mathbf{S}_B\mathbf{x} = c(\mathbf{m}_1 - \mathbf{m}_2)$$

Example







- Larger the within-class spread direction gets de-emphasized

$\mathbf{w} = \mathbf{S_w}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$

Fisher's linear discriminant (c-class)

- With c-1 discriminant functions
- Project from d-space to (c-1)-space
- Again, try to maximize between-class scatter to within-class scatter ratio for best separability

In the original feature space

- Within class scatter
  - Easy generalization into c classes

$$\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i$$

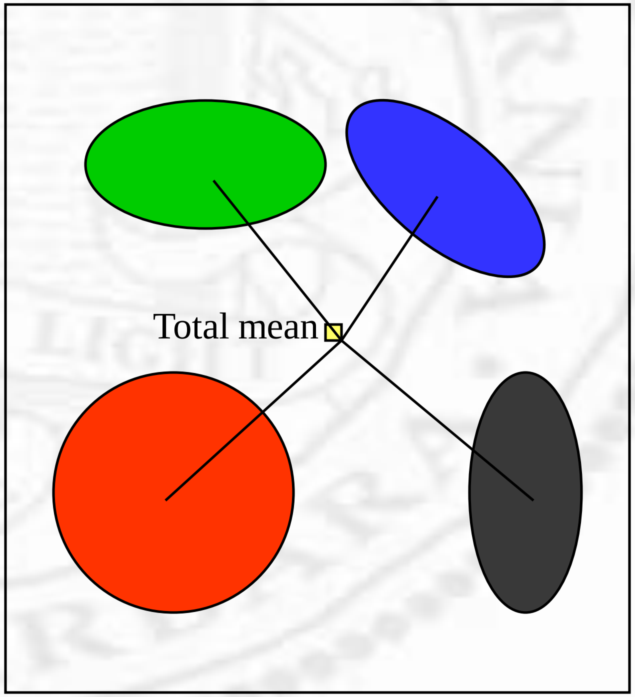
$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathbf{X}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathbf{X}_i} \mathbf{x}$$

Between Class Scattering

- More tricky
  - Total mean & total scatter
    $$\mathbf{m} = \frac{1}{n}\sum \mathbf{x} = \frac{1}{n}\sum_{i=1}^{c} n_i \mathbf{m}_i$$
    $$\mathbf{S}_T = \sum (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^t$$
  - Total scatter is made of
    - Scatter within a class
    - Scatter between classes

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m_i}-\mathbf{m})(\mathbf{m_i}-\mathbf{m})^t$$



Total mean & total scatter matrix

$$\mathbf{m} = \frac{1}{n}\sum \mathbf{x} = \frac{1}{n}\sum_{i=1}^{c} n_i \mathbf{m}_i$$

$$\mathbf{S}_T = \sum (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^t$$

$$\because \sum_{\mathbf{x}\in \mathbf{X}_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{m}-\mathbf{m}_i)^t = 0$$

$$\begin{aligned}
\mathbf{S}_T &= \sum (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^t \\
&= \sum_{i=1}^{c} \sum_{\mathbf{x}\in \mathbf{X}_i} (\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})(\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})^t \\
&= \sum_{i=1}^{c} \sum_{x\in \aleph_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t + \sum_{i=1}^{c} \sum_{\mathbf{x}\in \mathbf{X}_i} (\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t \\
&= \mathbf{S}_w + \sum_{i=1}^{c} n_i (\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t
\end{aligned}$$

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t$$

$$\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i$$

$$\mathbf{S}_i = \sum_{\mathbf{x}\in \mathbf{X}_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t$$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x}\in \mathbf{X}_i} \mathbf{x}$$

Meaning

- Total scatter = between class scatter + within class scatter
- In hypothesis testing
  - Between class scatter is significant
  - Within class scatter is insignificant (error)
- E.g., three different treatment option (surgery, drug, placebo)
  - Large between class scatter means one treatment is more effective than the others
  - Large within class scatter means that samples means variation among subjects of the same treatment

In the transformed (c-1)-dimensional space

$$y_i = w_i^t x \qquad i = 1,\ldots, c-1$$
$$y = \boxed{W}^t x$$
$$W_{d\times(c-1)}$$
$$\tilde{m}_i = \frac{1}{n_i}\sum_{y\in\aleph_i} y$$
$$\tilde{m} = \frac{1}{n}\sum_{i=1}^{c} n_i m_i$$
$$\tilde{S}_w = \sum_{i=1}^{c}\sum_{y\in\aleph_i}(y-\tilde{m}_i)(y-\tilde{m}_i)^t$$
$$\tilde{S}_B = \sum_{i=1}^{c}\sum_{y\in\aleph_i}(\tilde{m}_i-\tilde{m})(\tilde{m}_i-\tilde{m})^t$$

$$\tilde{S}_w = W^t S_w W$$
$$\tilde{S}_B = W^t S_B W$$
$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_w|} = \frac{|W^t S_B W|}{|W^t S_w W|}$$

- Transformed measure is a matrix
- Use determinant for spread volume

$$S_B w_i = \lambda_i S_w w_i$$

Multi-Dimensional Scaling

- Given n objects and a confusion (similarity or dis-similarity) matrix nxn
- Distance (similarity) can be numbers (Euclidean distance) or ranking
- Find an embedding in an m-dimensional space where the distance (similarity) is preserved

Algorithms

1. Set up the matrix of squared proximities $\mathbf{P^{(2)}} = [p^2]$.

2. Apply the double centering: $\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{P^{(2)}}\mathbf{J}$ using the matrix $\mathbf{J} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}'$, where n is the number of objects.

3. Extract the m largest positive eigenvalues $\lambda_1 \dots \lambda_m$ of $\mathbf{B}$ and the corresponding m eigenvectors $\mathbf{e_1} \dots \mathbf{e_m}$.

4. A m-dimensional spatial configuration of the n objects is derived from the coordinate matrix $\mathbf{X} = \mathbf{E_m}\mathbf{\Lambda}_m^{1/2}$, where $\mathbf{E_m}$ is the matrix of m eigenvectors and $\mathbf{\Lambda}_m$ is the diagonal matrix of m eigenvalues of $\mathbf{B}$, respectively.

Algorithms

- B is similar to "convariance matrix" and can be reconstructed by eigen vectors and eigenvalues

$$\mathbf{B} = -\frac{1}{2}(\boldsymbol{I} - \frac{1}{n}\mathbf{11'})\,\boldsymbol{P}^2\,(\boldsymbol{I} - \frac{1}{n}\mathbf{11'})$$
$$= -\frac{1}{2}(\boldsymbol{I} - \frac{1}{n}\mathbf{11'})\,\mathbf{XX'}(\boldsymbol{I} - \frac{1}{n}\mathbf{11'})$$
$$= -\frac{1}{2}(\boldsymbol{I} - \frac{1}{n}\mathbf{11'})\,\mathbf{X}((\boldsymbol{I} - \frac{1}{n}\mathbf{11'})'\,\mathbf{X})'$$
$$= -\frac{1}{2}((\boldsymbol{I} - \frac{1}{n}\mathbf{11'})\,\mathbf{X})((\boldsymbol{I} - \frac{1}{n}\mathbf{11'})\,\mathbf{X})'$$
$$= -\frac{1}{2}(\mathbf{X} - \frac{1}{n}\mathbf{11'}\,\mathbf{X})(\mathbf{X} - \frac{1}{n}\mathbf{11'}\,\mathbf{X})'$$
$$-\frac{1}{2}(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})'$$

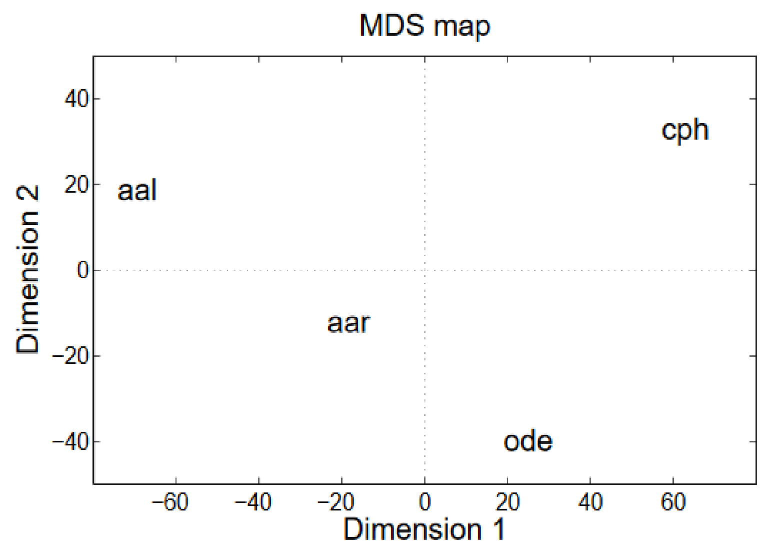
$$\begin{array}{rrrrr} & \text{cph} & \text{aar} & \text{ode} & \text{aal} \\ \text{cph} & 0 & 93 & 82 & 133 \\ \text{aar} & 93 & 0 & 52 & 60 \\ \text{ode} & 82 & 52 & 0 & 111 \\ \text{aal} & 133 & 60 & 111 & 0 \end{array}.$$

The matrix of squared proximities is

$$\mathbf{P}^{(2)} = \begin{bmatrix} 0 & 8649 & 6724 & 17689 \\ 8649 & 0 & 2704 & 3600 \\ 6724 & 2704 & 0 & 12321 \\ 17689 & 3600 & 12321 & 0 \end{bmatrix}.$$

Since there are $n = 4$ objects, the matrix $\mathbf{J}$ is calculated by

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - 0.25 \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix}.$$



$$\mathbf{B} = -\tfrac{1}{2}\mathbf{J}\mathbf{P}^{(2)}\mathbf{J} = \begin{bmatrix} 5035.0625 & -1553.0625 & 258.9375 & -3740.938 \\ -1553.0625 & 507.8125 & 5.3125 & 1039.938 \\ 258.9375 & 5.3125 & 2206.8125 & -2471.062 \\ -3740.9375 & 1039.9375 & -2471.0625 & 5172.062 \end{bmatrix}.$$

$$\lambda_1 = 9724.168, \ \lambda_2 = 3160.986, \quad \mathbf{e_1} = \begin{pmatrix} -0.637 \\ 0.187 \\ -0.253 \\ 0.704 \end{pmatrix}, \ \mathbf{e_2} = \begin{pmatrix} -0.586 \\ 0.214 \\ 0.706 \\ -0.334 \end{pmatrix}$$

$$\mathbf{X} = \begin{bmatrix} -0.637 & -0.586 \\ 0.187 & 0.214 \\ -0.253 & 0.706 \\ 0.704 & -0.334 \end{bmatrix} \begin{bmatrix} \sqrt{9724.168} & 0 \\ 0 & \sqrt{3160.986} \end{bmatrix} = \begin{bmatrix} -62.831 & -32.97448 \\ 18.403 & 12.02697 \\ -24.960 & 39.71091 \\ 69.388 & -18.76340 \end{bmatrix}.$$

Multi-Dimensional Scaling

- Original space
  - dimension d
- Reduced-dimensional space
  - dimension d'

$$\aleph = \{x_1, x_2, ..., x_n\}$$

$$\alpha_{ij} = |x_i - x_j|$$

$$\Im = \{y_1, y_2, ..., y_n\}$$

$$\beta_{ij} = |y_i - y_j|$$

Select $\Im$ in such a way to preserve the n(n-1)/2 *distance* measurements through dimension reduction

MDS Solution

- Find β_{ij} as close to original α_{ij} as possible
- Metric MDS

$$f \text{ is a monotonic, metric-preserving function}$$

$$f(\beta_{ij}) = \alpha_{ij}$$

$$f(\beta_{ij}) = a\alpha_{ij} + b$$

- NonMetric MDS
  - rank orders are the same in both
  - f can be any monotonic function

Possible Cost (Stress) Functions

$$c = \left(\frac{\sum_{i,j} (\alpha_{ij} - f(\beta_{ij}))}{\sum_{i,j} \alpha_{ij}^{2}} \right)^{\frac{1}{2}}$$

$$c = \frac{1}{\sum_{i<j} \alpha_{ij}^{2}} \sum_{i<j} (\alpha_{ij} - \beta_{ij})^2$$

$$c' = \sum_{i<j} \left(\frac{\alpha_{ij} - \beta_{ij}}{\alpha_{ij}} \right)^2$$

$$c'' = \frac{1}{\sum_{i<j} \alpha_{ij}} \sum_{i<j} \frac{(\alpha_{ij} - \beta_{ij})^2}{\alpha_{ij}}$$

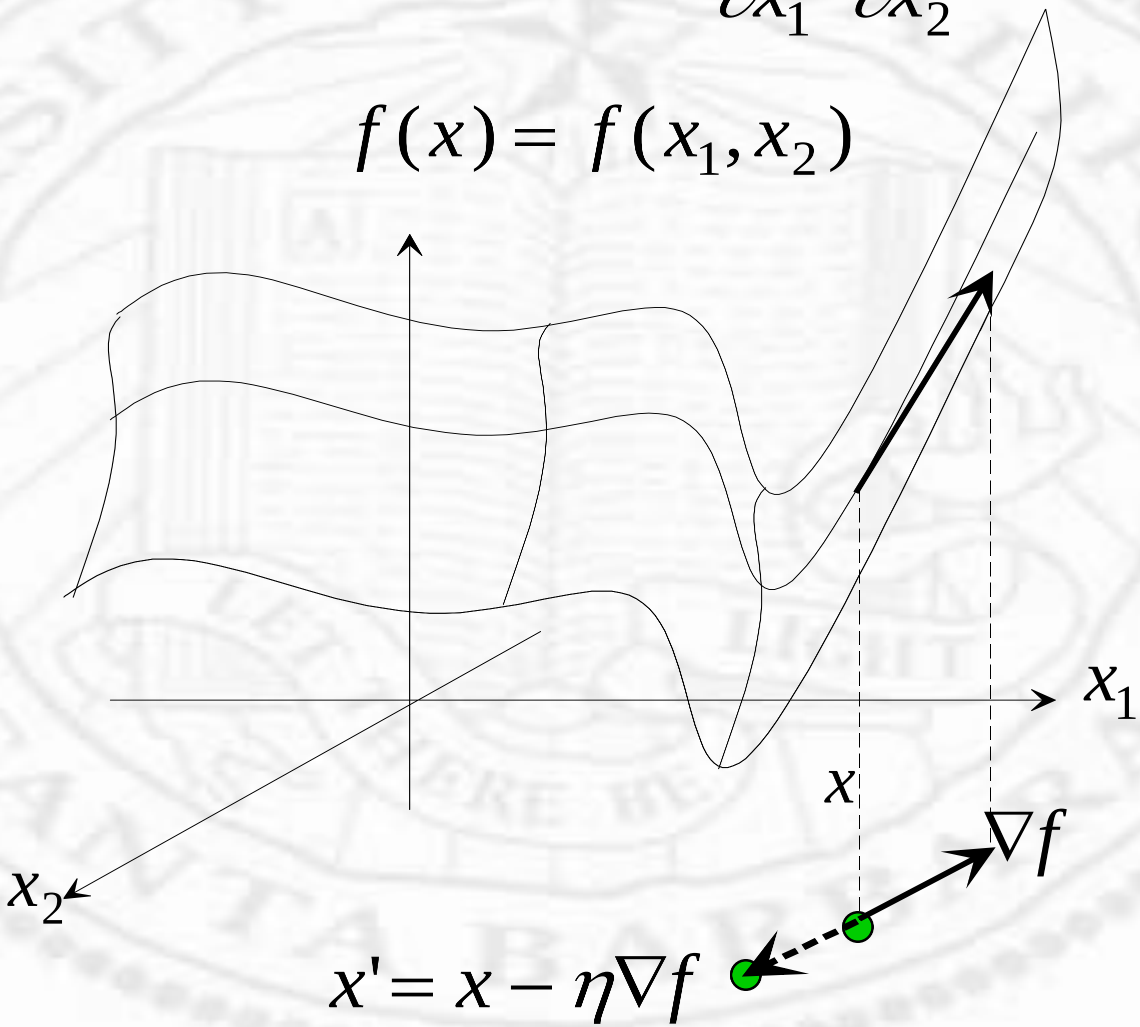
Gradient Descent

- A search mechanism
- Start at an arbitrarily chosen starting point
- Move in a direction (negative gradient) to minimize the cost function

An iterative algorithm

$$x' = x - \eta\nabla f = (x_1, x_2) - \eta\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)$$

$$f(x) = f(x_1, x_2)$$



Their gradient directions

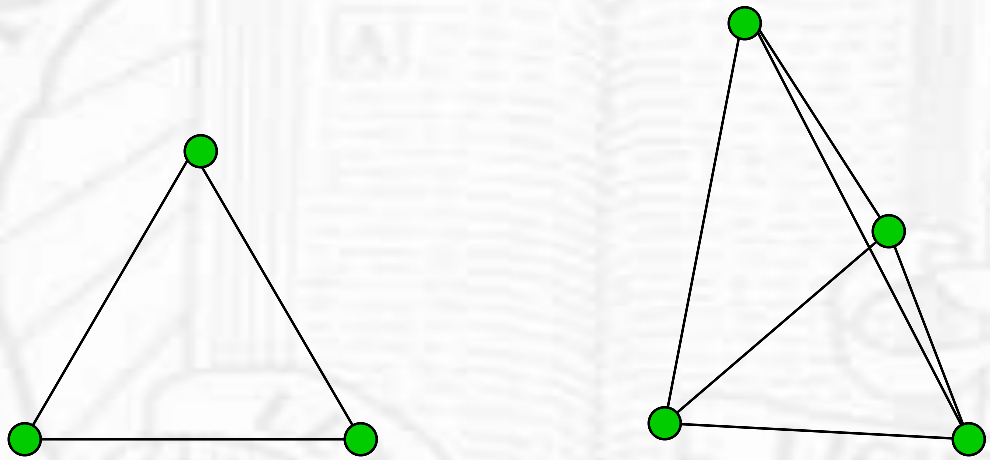
$$\nabla_{y_k} c = \frac{-2}{\sum_{i<j} \alpha_{ij}^{\,2}} \sum_{j \neq k} (\alpha_{kj} - \beta_{kj}) \frac{y_k - y_j}{\beta_{kj}}$$

$$\nabla_{y_k} c' = -2 \sum_{j \neq k} \frac{\alpha_{kj} - \beta_{kj}}{\alpha_{kj}^{\,2}} \frac{y_k - y_j}{\beta_{kj}}$$

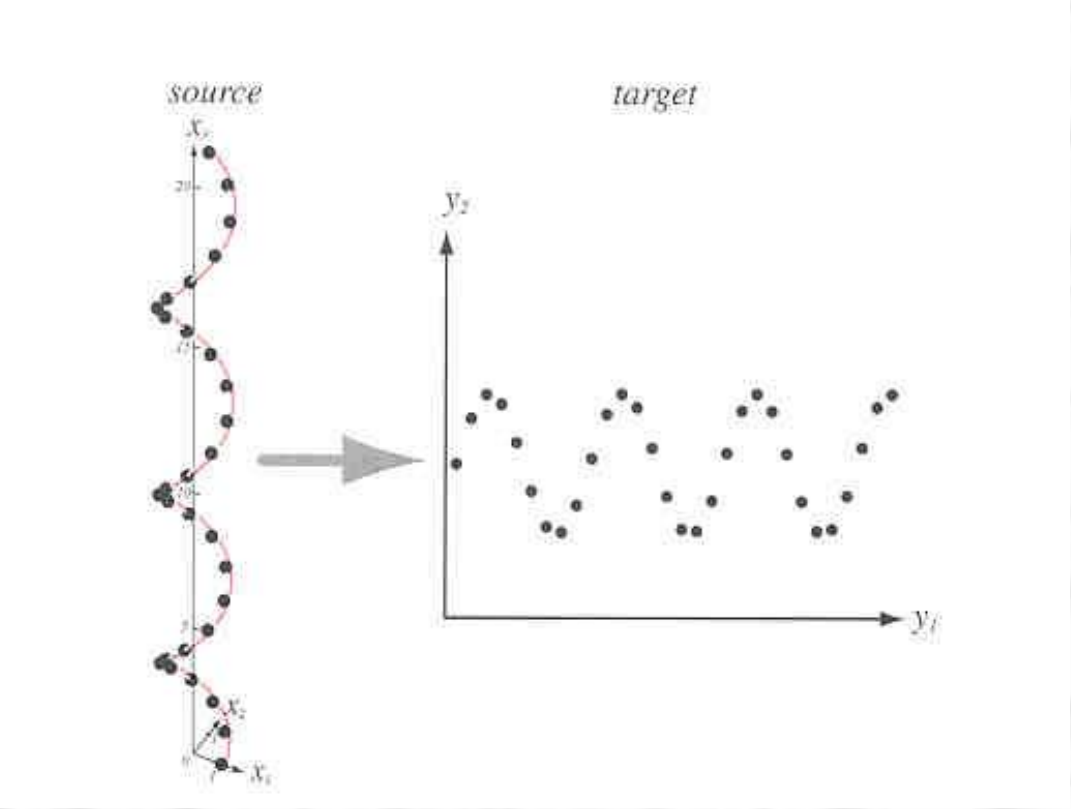
$$\nabla_{y_k} c'' = \frac{-2}{\sum_{i<j} \alpha_{ij}} \sum_{j \neq k} \frac{\alpha_{kj} - \beta_{kj}}{\alpha_{kj}} \frac{y_k - y_j}{\beta_{kj}}$$

How many dimensions?

- Again, for visualization purpose, it is usually 2 or 3



Example



An Example

- d movies, with rating from -1 (bad) to 0 (neutral) to 1 (good)
- R_i can be considered a random variable with the underlying universe being all n viewers
- $E(R_i)$ = expected (average) ratings from all viewers
- $var(R_i) = E(R_i-E(R_i))^2$ variance (spread) in ratings from all viewers
- $cov(i,j) = E[(R_i-E(R_i))\ (R_j-E(R_j))]$ covariance (correlation) of ratings of two movies

An Example (cont.)

- Covariance matrix: a dxd matrix with entry being cov(i,j)
- cov(i,j) is symmetrical
  - Has $Q\Lambda Q^T$ eigen decomposition
  - What are the physical meaning of Q and Λ?

PCA (Principal Component Analysis)

where

$$\mathbf{X}_{dxn}\mathbf{X}^{t}{}_{nxd} = \left(\begin{bmatrix} | & 0 & 0 \\ \mathbf{X}_1 & 0 & 0 \\ | & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & | & 0 \\ 0 & \mathbf{X}_2 & 0 \\ 0 & | & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & | \\ 0 & 0 & \mathbf{X}_3 \\ 0 & 0 & | \end{bmatrix} \right)$$

$$\left(\begin{bmatrix} - & \mathbf{X}_1^T & - \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ - & \mathbf{X}_2^T & - \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ - & \mathbf{X}_3^T & - \end{bmatrix} \right)$$

$$= \sum_i \mathbf{X}_i \mathbf{X}_i^T$$

PCA (Principal Component Analysis)

- N time the covariance matrix (assume the mean is zero for now)

$$\sum_i \mathbf{x}_i \mathbf{x}_i^T = \begin{bmatrix} \sum_{k=1}^{n} x_1^{(k)} x_1^{(k)} & \sum_{k=1}^{n} x_1^{(k)} x_2^{(k)} & \cdots & \sum_{k=1}^{n} x_1^{(k)} x_d^{(k)} \\ \sum_{k=1}^{n} x_2^{(k)} x_1^{(k)} & \sum_{k=1}^{n} x_2^{(k)} x_2^{(k)} & \cdots & \sum_{k=1}^{n} x_2^{(k)} x_d^{(k)} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{k=1}^{n} x_d^{(k)} x_1^{(k)} & \sum_{k=1}^{n} x_d^{(k)} x_2^{(k)} & & \sum_{k=1}^{n} x_d^{(k)} x_d^{(k)} \end{bmatrix}$$

Principal Component Analysis

- Extract a set of compact basis which best describe the data set

$$\aleph = \{x_1, x_2, \ldots, x_n\}$$

$$\begin{bmatrix} x_{11} & x_{12} & . & x_{1n} \\ x_{21} & x_{22} & . & x_{2n} \\ . & . & x_{ij} & . \\ x_{d1} & x_{d2} & . & x_{dn} \end{bmatrix}_{d\times n} =$$

$$\begin{bmatrix} u_{11} & u_{12} & . & u_{1d} \\ u_{21} & u_{22} & . & u_{2d} \\ . & . & u_{ij} & . \\ u_{d1} & u_{d2} & . & u_{dd} \end{bmatrix}_{d\times d} \begin{bmatrix} \sigma_{11} & 0 & 0 \\ 0 & . & 0 \\ 0 & 0 & \sigma_{nn} \\ 0 & 0 & 0 \end{bmatrix}_{d\times n} \begin{bmatrix} v_{11} & . & v_{n1} \\ . & . & . \\ v_{1n} & . & v_{nn} \end{bmatrix}_{n\times n}$$

$$\mathbf{X}_{d\times n} = \mathbf{U}_{d\times d} \mathbf{\Sigma}_{d\times n} \mathbf{V}^{t}{}_{n\times n}$$

Can be shown that

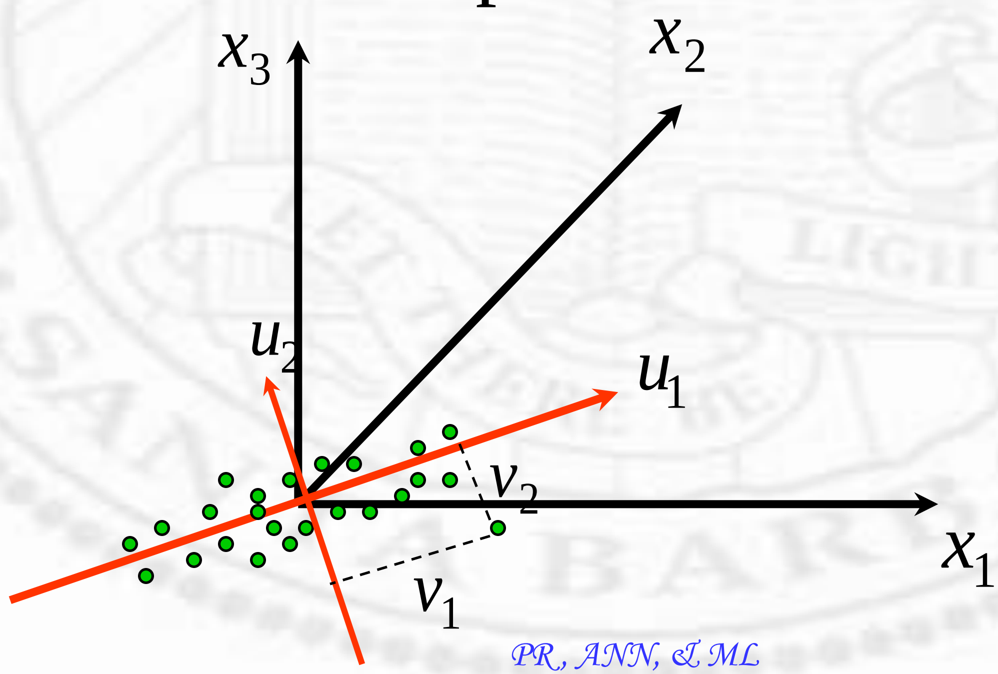
$$x_i = \sum_{j=1}^{n} v_{ij} \sigma_{jj} u_j$$

- x_i is an original vector
- u_j is a basis vector
- σ_{jj} is the significance of the basis vector
- v_{ij} is the weight of the particular basis vector

- If data set is highly correlated, usually only a few bases are significant
- Use v_i instead of x_i
- Reduce dimensionality from d to n or less

Important SVD properties

- Orthogonal bases
- Importance ranked axis direction
- Body-fitted coordinate system
- Uncorrelated components



Furthermore

$$\mathbf{X}_{d\times n} = \mathbf{U}_{d\times d}\boldsymbol{\Sigma}_{d\times n}\mathbf{V}^{t}{}_{n\times n}$$

$$\mathbf{X}_{d\times n}\mathbf{X}^{t}{}_{n\times d} = (\mathbf{U}_{d\times d}\boldsymbol{\Sigma}_{d\times n}\mathbf{V}^{t}{}_{n\times n})(\mathbf{U}_{n\times n}\boldsymbol{\Sigma}^{t}{}_{d\times n}\mathbf{U}^{t}{}_{d\times d})$$

$$= \mathbf{U}_{d\times d}\boldsymbol{\Sigma}^{2}{}_{d\times n}\mathbf{U}^{t}{}_{d\times d}$$

- SVD of the samples can be used to derive the PCA transform of the class
  - the same basis functions $u_i^{PCA} = u_i^{SVD}$
  - related eigenvalues $\sigma_{ii}^{PCA} = (\sigma_{ii}^{SVD})^2$

How to Use PCA

$$\mathbf{C}_{dxd} = \mathbf{X}_{d\times n}\mathbf{X}^{t}{}_{n\times d} = \mathbf{U}_{d\times d}\mathbf{\Sigma}^{2}{}_{d\times n}\mathbf{U}^{t}{}_{d\times d}$$

$$\mathbf{C}_{dxd}\mathbf{x} = \mathbf{U}_{d\times d}\mathbf{\Sigma}^{2}{}_{d\times n}\mathbf{U}^{t}{}_{d\times d}\mathbf{x}$$

- Scotty-beam-me-up:
  - Red: projection (decomposition) into important data dimensions
  - Green: “massage” according to importance
  - Blue: reconstruction onto important basis
- Represent in “body-fitted” coordinate system, e.g., for similarity search

Math Detail

$$\mathbf{M} = \sum_i \mathbf{x}_i \mathbf{x}_i^T$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{u}_1 & \vdots & \mathbf{u}_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{u}_1^T & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \mathbf{u}_n^T & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{u}_1 & \vdots & \mathbf{u}_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sqrt{\sigma_1} & & \\ & \ddots & \\ & & \sqrt{\sigma_n} \end{bmatrix} \begin{bmatrix} \sqrt{\sigma_1} & & \\ & \ddots & \\ & & \sqrt{\sigma_n} \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{u}_1^T & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \mathbf{u}_n^T & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots \\ \sqrt{\sigma_1}\mathbf{u}_1 & \vdots & \sqrt{\sigma_n}\mathbf{u}_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \cdots & \sqrt{\sigma_1}\mathbf{u}_1^T & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \sqrt{\sigma_n}\mathbf{u}_n^T & \cdots \end{bmatrix}$$

$$= \sum_{\boxed{i=1}}^{n} \left(\sqrt{\sigma_i}\mathbf{u}_i\right)\left(\sqrt{\sigma_i}\mathbf{u}_i^T\right) \longrightarrow \begin{bmatrix} \vdots & \vdots & \vdots \\ 0 & \mathbf{C}_i & 0 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \cdots & 0 & \cdots \\ \cdots & \mathbf{C}_j^T & \cdots \\ \cdots & 0 & \cdots \end{bmatrix} = \begin{cases} \mathbf{C}_i\mathbf{C}_j^T & i = j \\ 0 & i \neq j \end{cases}$$

$$\approx \sum_{\boxed{i=1}}^{k} \left(\underline{\sqrt{\sigma_i}\mathbf{u}_i}\right)\left(\sqrt{\sigma_i}\mathbf{u}_i^T\right)$$

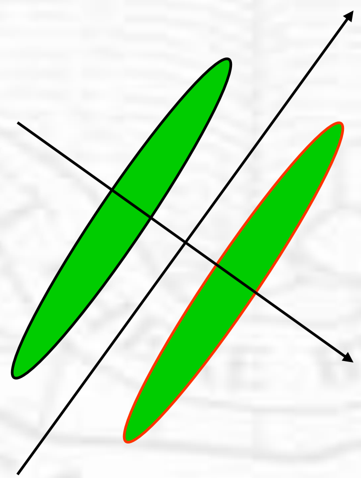
- k<<n, only a few dimensions are kept
- Embedding are the rows of $\sqrt{\sigma_i}\mathbf{u}_i$

Intuition

- **u**'s represent
  - Body-fitted
  - Uncorrelated
  - Importance-ranked dimensions
- Instead of using original vectors (**x**) projected on standard basis, use (**x**) projected on **u**
- Use as many or as few as you want (recall dimension reduction)

Caveat

- PCA gives the dimension for best *representation* of data, which does not necessarily implies best dimension for *discrimination* of data



Caveat

- PCA is sensitive to data preprocessing
  - Centering
  - Normalization
- Different normalization (weighting) gives different preference to features
  - NBA player salary = f (height, ppg)
- The number of important dimensions (e.g., height and ppg are correlated) should be preserved

Caveat

- XX^T is a very frequently seen math construct
  - Treat as a vector in PCA
  - Treat as a vector of random variables in KL
  - Treat as a vector of partial derivatives in Hessian
- XX^T is
  - Symmetric, positive semidefinite
  - Eigen values are real and >=0

Kernel PCA

- A generalization of PCA in the feature space
- Idea is this:
  - Linear structures might not exist in original feature space
  - But might exist after a nonlinear mapping into a higher-dimensional space
  - Linear algebra can be used for data analysis in higher dimensional space
  - With kernel tricks, mapping need not be actually done

Kernel CPA

- Requirements: only inner products are used in decomposing covariance matrix
- Dot(xi,xj) can be done
  - In the original space
  - In Kernel space without explicit mapping

Math Details

- Compute covariance matrix $\mathbf{R} = \frac{1}{N}\sum_i \varphi(\mathbf{x_i})\varphi(\mathbf{x}_i)^T$
- Find eigen vectors and values $\mathbf{Rq} = \lambda\mathbf{q}$
- Represent in kernel math $\mathbf{q} = \sum_{j=1}^{n} \alpha_j \varphi(\mathbf{x_j})$
  - "representable" components in $\varphi(\mathbf{x_j})$

Details

$$\mathbf{q}_k = \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j})$$

$$\mathbf{R}\mathbf{q}_k = \lambda \mathbf{q}_k$$

$$\Rightarrow \sum_{i=1}^{n} \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T \mathbf{q}_k = N\lambda \mathbf{q}_k \qquad \mathbf{R} = \frac{1}{N}\sum_i \varphi(\mathbf{x_i})\varphi(\mathbf{x}_i)^T$$

$$\Rightarrow \sum_{i=1}^{n} \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j}) = N\lambda \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j}) \qquad \mathbf{q}_k = \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j})$$

$$\Rightarrow \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x}_i)K(\mathbf{x}_i,\mathbf{x}_j) = N\lambda \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j}) \qquad \varphi(\mathbf{x})^T\mathbf{q}_k = \varphi(\mathbf{x})^T\sum_{j=1}^{n}\alpha_{kj}\varphi(\mathbf{x_j}) = \sum_{j=1}^{n}\alpha_{kj}\varphi(\mathbf{x})^T\varphi(\mathbf{x_j}) = \sum_{j=1}^{n}\alpha_{kj}K(\mathbf{x}_j,\mathbf{x})$$

$$\Rightarrow \varphi(\mathbf{x}_k)\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x}_i)K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_k)N\lambda \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j})$$

$$\Rightarrow \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{kj}K(\mathbf{x}_k,\mathbf{x}_i)K(\mathbf{x}_i,\mathbf{x}_j) = N\lambda \sum_{j=1}^{n} \alpha_{kj}K(\mathbf{x}_k,\mathbf{x}_j)$$

$$\mathbf{K}^2\boldsymbol{\alpha} = N\lambda\mathbf{K}\boldsymbol{\alpha} \Rightarrow \mathbf{K}\boldsymbol{\alpha} = N\lambda\boldsymbol{\alpha}$$

How to Use?

- Solve $\mathbf{K}\boldsymbol{\alpha} = N\lambda\boldsymbol{\alpha}$
  - **K**: kernel matrix, $\boldsymbol{\alpha}_k$: eigen vectors
- Representation $\varphi(\mathbf{x})^T \mathbf{q}_k = \varphi(\mathbf{x})^T \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x_j}) = \sum_{j=1}^{n} \alpha_{kj}\varphi(\mathbf{x})^T \varphi(\mathbf{x_j}) = \sum_{j=1}^{n} \alpha_{kj} K(\mathbf{x}_j, \mathbf{x})$
  - only k(x, xj) are needed
  - $\boldsymbol{\alpha}_k$: solved in the previous step
- Possible to find representation basis and map unknown vectors using Kernel function without explicit mapping

PCA and MDS

- PCA provides a linear solution to a version of the metric MDS
- Distance measurements are real and symmetrical
- Use a particular definition of distance: inner product
  - Caveat: inner product requires a coordinate system (origin) while pair-wise distance does not
  - Inner product defines pair-wise distance but not vice versa
- Put all the pair-wise distances into a matrix, m_{ij}=distance between features i and j (this is the Gram matrix)
- Recall that **M** is made of n rank-one matrices
- Only a small number (2-3 for visualization purpose) of those are kept if singular values drop off quickly – mapping into a lower dimension space

Final Notes

- Other techniques, such as Self-Organization Map (SOM) are available
- SOM is discussed later in non-supervised techniques