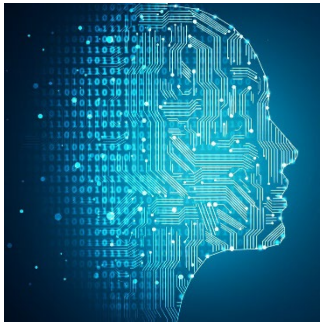
Machine Learning

Principal Components Analysis

Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Outline

- Dimensionality Reduction
- Projections
- PCA
  - Projection Perspective
  - Maximum Variance Perspective
  - Connection to SVD
- MDS
  - Connection to PCA

Unsupervised Learning

- On to unsupervised learning!
- We’ll come back to supervised learning later.
- There are three main unsupervised learning topics that we will cover in this course:
    1. Dimensionality reduction
    2. Clustering
    3. Density estimation

Dimensionality Reduction

- In dimensionality reduction problems, we observe $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$
  - Notice we do not have labels (unsupervised learning)
- Goal: transform these variables to new ones

$$\boldsymbol{x}_i \to \boldsymbol{\theta}_i \in \mathbb{R}^k$$

where $k < d$, such that information loss is minimized

Dimensionality Reduction

Reasons for doing dimensionality reduction:

- Computational efficiency
- Visualization ($k = 2, 3$)
- Compression
- Interpreting data (which dimensions are important?)
- Eliminate rank deficiency
- Eliminate useless/noisy features
- Avoid overfitting

Dimensionality Reduction Types

- Methods for dimensionality reduction can be classified according to:
  1. How is information loss quantified?
  2. Supervised or unsupervised?
  3. Feature selection (select existing features) or feature extraction (form new features from old ones)?
  4. Parametric or nonparametric?
  5. Linear or nonlinear?
  6. Generative or discriminative?

PCA

- The first method for dimensionality reduction that we will discuss is principal components analysis (PCA)
    1. How is information loss quantified? Least squares
    2. Supervised or unsupervised?
    3. Feature selection or feature extraction?
    4. Parametric or nonparametric?
    5. Linear or nonlinear?
    6. Generative or discriminative?

Either, although our initial perspective is discriminative

Importance of PCA

PCA is perhaps the most important unsupervised method

- For high-dimensional supervised learning problems, performing PCA first can be very helpful
- Many other dimensionality reduction methods can be framed as learning a different representation of the data and then applying PCA
  - E.g. diffusion maps and kernel PCA (kernel trick applied to PCA)
- Other dimensionality reduction methods use PCA as a preprocessing step
- You should become very familiar with it

Projections

- Let $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k \in \mathbb{R}^d$ be linearly independent column vectors. Denote

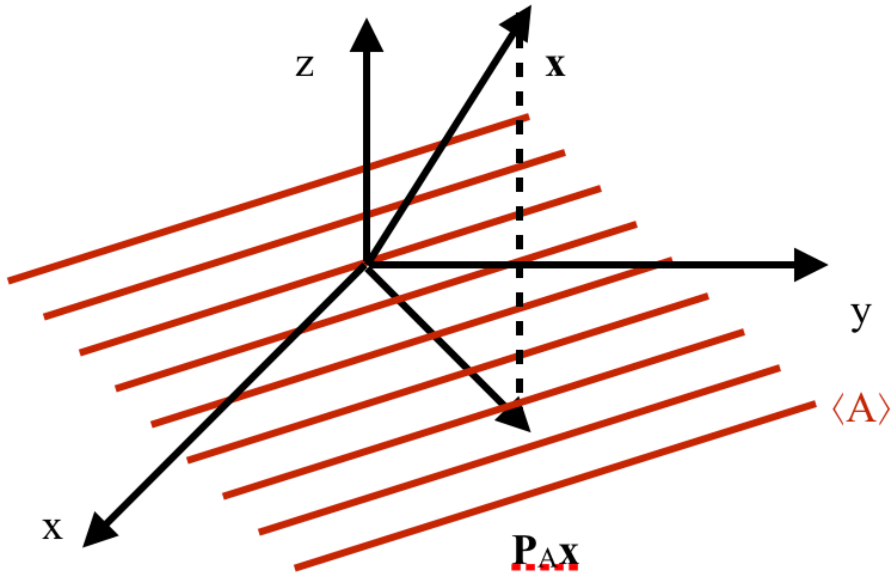
$$\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_k] \qquad (d \times k)$$

- The linear span of $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k$ is the column span of $\boldsymbol{A}$, written

$$\begin{aligned}\langle \boldsymbol{A} \rangle &= \text{colspan}(\boldsymbol{A}) \\ &= \left\{ \boldsymbol{A}\boldsymbol{\theta} \mid \boldsymbol{\theta} \in \mathbb{R}^k \right\}.\end{aligned}$$

- The *projection* onto $\langle \boldsymbol{A} \rangle$ is the mapping $\boldsymbol{P}_A : \mathbb{R}^d \to \langle \boldsymbol{A} \rangle \subseteq \mathbb{R}^d$

$\boldsymbol{P}_A \boldsymbol{x} =$ closest point in $\langle \boldsymbol{A} \rangle$ to $\boldsymbol{x}$



Projections

- Every point in $\langle \boldsymbol{A} \rangle$ equals $\boldsymbol{A\theta}$ for some $\boldsymbol{\theta} \in \mathbb{R}^k$
- Therefore, $\boldsymbol{P}_A \boldsymbol{x} = \boldsymbol{A}\widehat{\boldsymbol{\theta}}$, where

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{x} - \boldsymbol{A\theta}\|$$

- We have previously seen that the solution is

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{x}$$

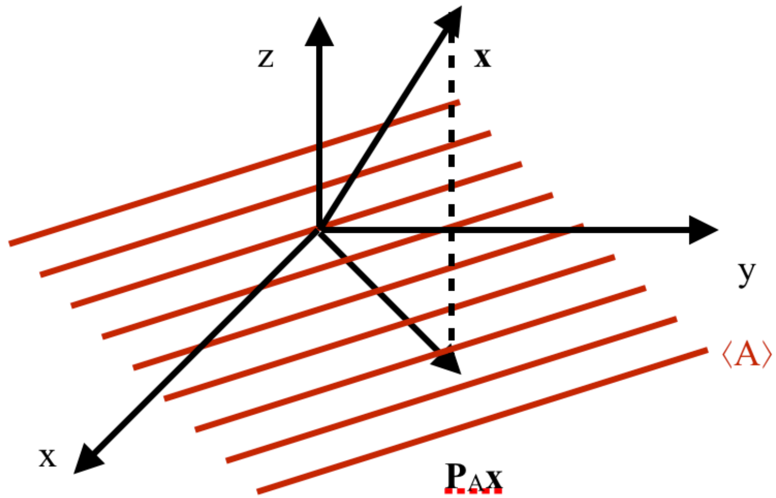
- Therefore,

$$\begin{aligned} \boldsymbol{P}_A\boldsymbol{x} &= \boldsymbol{A}\hat{\theta} \\ &= \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{x} \\ \Rightarrow \boldsymbol{P}_A &= \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T \end{aligned}$$

Properties of Projections

- If $\boldsymbol{a}_1, \dots, \boldsymbol{a}_k$ are orthonormal, then $\boldsymbol{P}_{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{A}^T$ $(\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I})$
- The orthogonality principle states that

$$\forall \boldsymbol{x}, \boldsymbol{x} - \boldsymbol{P}_{\boldsymbol{A}}\boldsymbol{x} \in \langle \boldsymbol{A} \rangle^{\perp}$$



- Proof: Let $\boldsymbol{u} \in \langle \boldsymbol{A} \rangle$. Then we can write $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{\theta}$ for some $\boldsymbol{\theta} \in \mathbb{R}^k$. Then

$$\begin{aligned}
\langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{P}_{\boldsymbol{A}}\boldsymbol{x} \rangle &= \left\langle \boldsymbol{u}, \left(\boldsymbol{I} - \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\right)\boldsymbol{x} \right\rangle \\
&= \boldsymbol{\theta}^T\boldsymbol{A}^T\left(\boldsymbol{I} - \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\right)\boldsymbol{x} \\
&= \boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{x} \\
&= 0.
\end{aligned}$$

Group Exercise

1. Show that projection matrices are idempotent, that is,

$$\boldsymbol{P}_A^2 = \boldsymbol{P}_A.$$

   Give an intuitive explanation of this property.

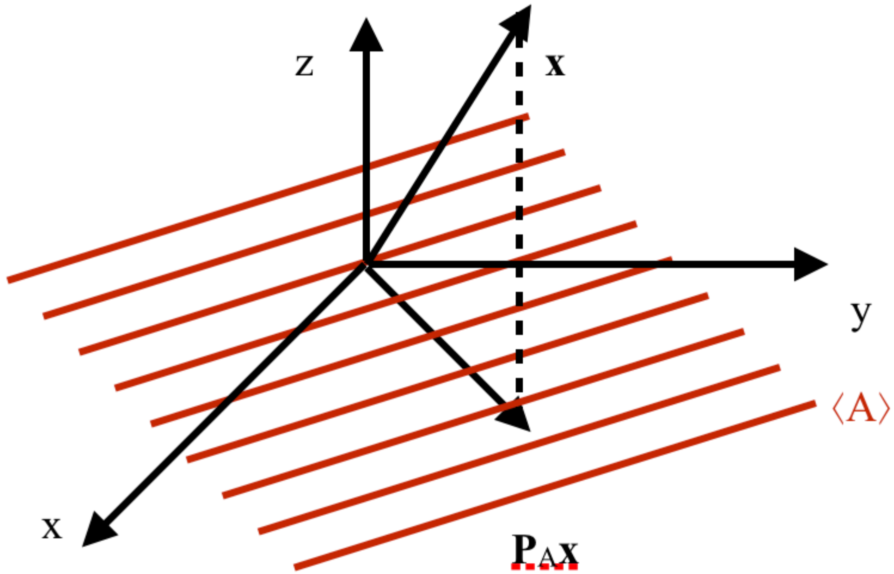
2. Let $\boldsymbol{B}$ be a $d \times (d-k)$ full rank matrix such that

$$\langle \boldsymbol{B} \rangle = \langle \boldsymbol{A} \rangle^{\perp}.$$

   Determine formulas for

   (a) $\boldsymbol{P}_A \boldsymbol{P}_B$

   (b) $\boldsymbol{P}_A + \boldsymbol{P}_B$



PCA

- The idea behind PCA is to approximate

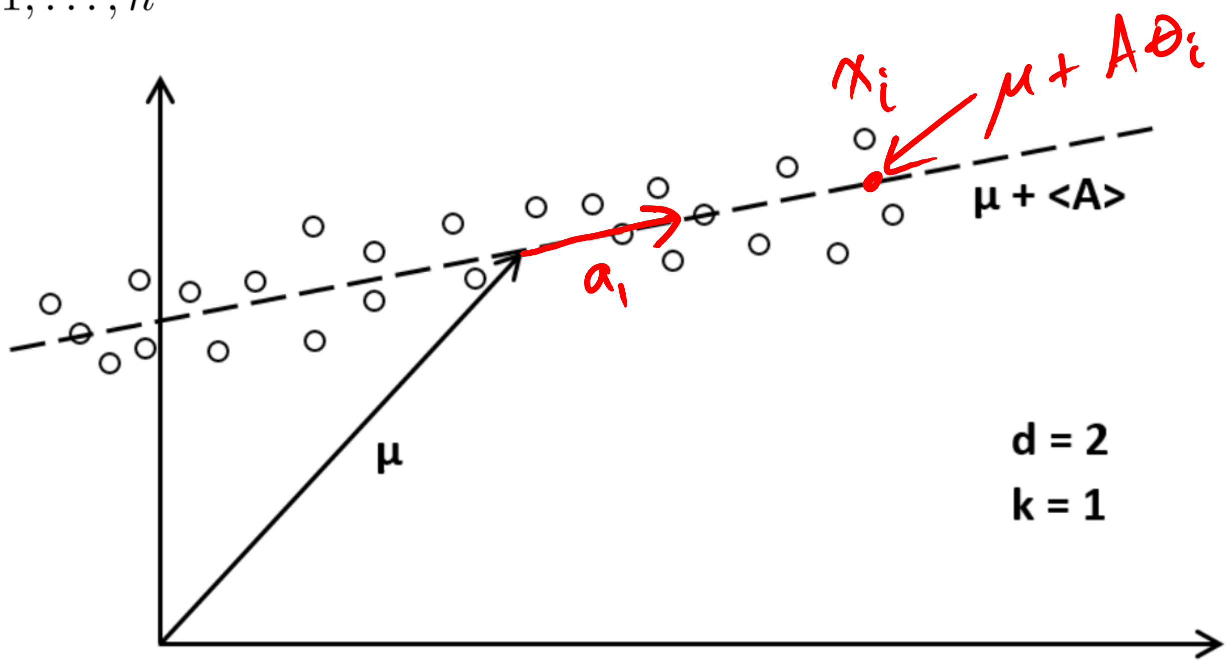
$$\boldsymbol{x}_i \approx \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{\theta}_i$$

where

$$A = [a_1] \in \mathbb{R}^d$$

$$\boldsymbol{\mu} \in \mathbb{R}^d$$
$$\boldsymbol{A} \in \mathcal{A}_k := \{\boldsymbol{A} \in \mathbb{R}^{d\times k} \mid \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_{k\times k}\}$$
$$\boldsymbol{\theta}_i \in \mathbb{R}^k, i = 1, \ldots, n$$

- **Example:** $d = 2, k = 1.$

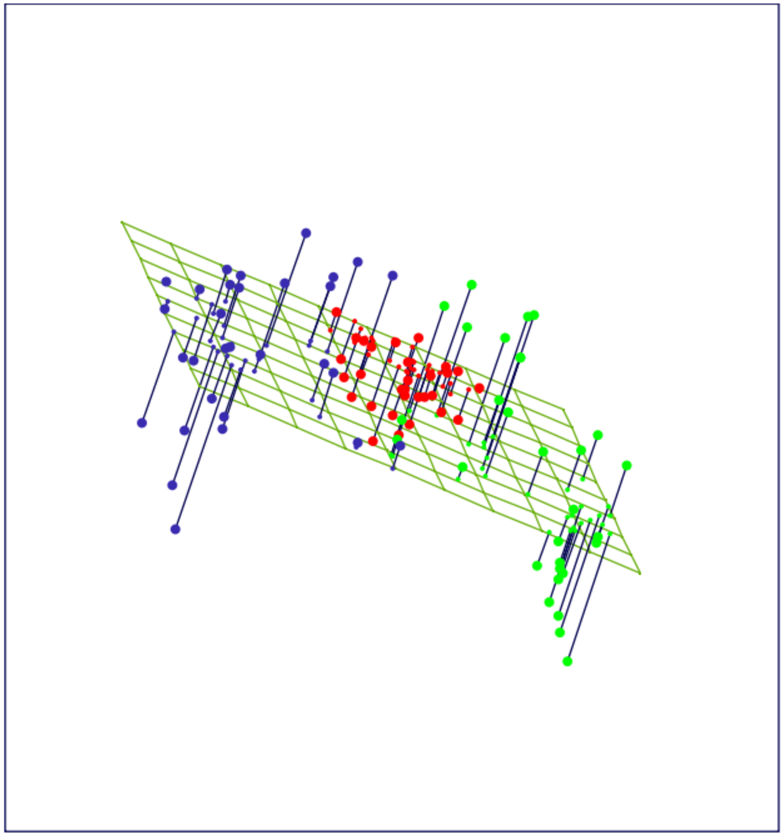


PCA

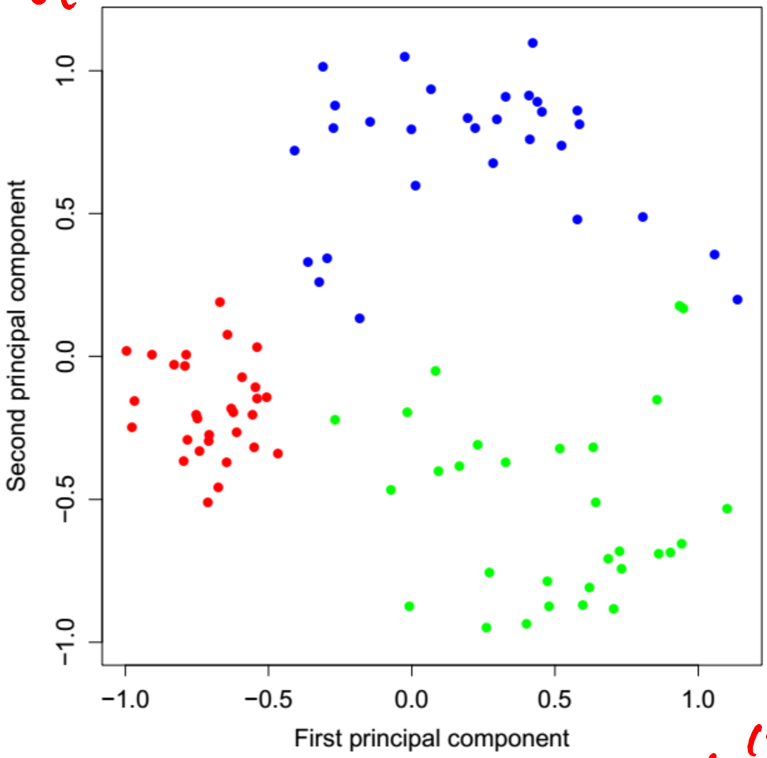
- **Example:** $d = 3, k = 2$

$\hat{x}_i = \mu + A\theta_i$

$\theta_i = \begin{bmatrix} \theta_i^{(1)} \\ \theta_i^{(2)} \end{bmatrix}$



$\theta_i^{(2)}$



$\theta_i^{(1)}$

PCA

- Mathematically, we define $\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n$ to be the solution of

$$\min_{\substack{\boldsymbol{\mu},\boldsymbol{A},\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n \\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{\theta}_i\|^2$$

- PCA gives the least squares rank-k linear approximation to the data set.
- The solution is given in terms of the spectral (or eigenvalue decomposition of the sample covariance matrix:

$$S = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T = U\Lambda U^T$$

PCA Solution

- Note that S is PSD so $\lambda_i \geq 0$:

$$z^T S z = \frac{1}{n}\sum_{i=1}^{n} z^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \boldsymbol{z}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{z}^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})\right)^2 \geq 0$$

- A solution to PCA is:

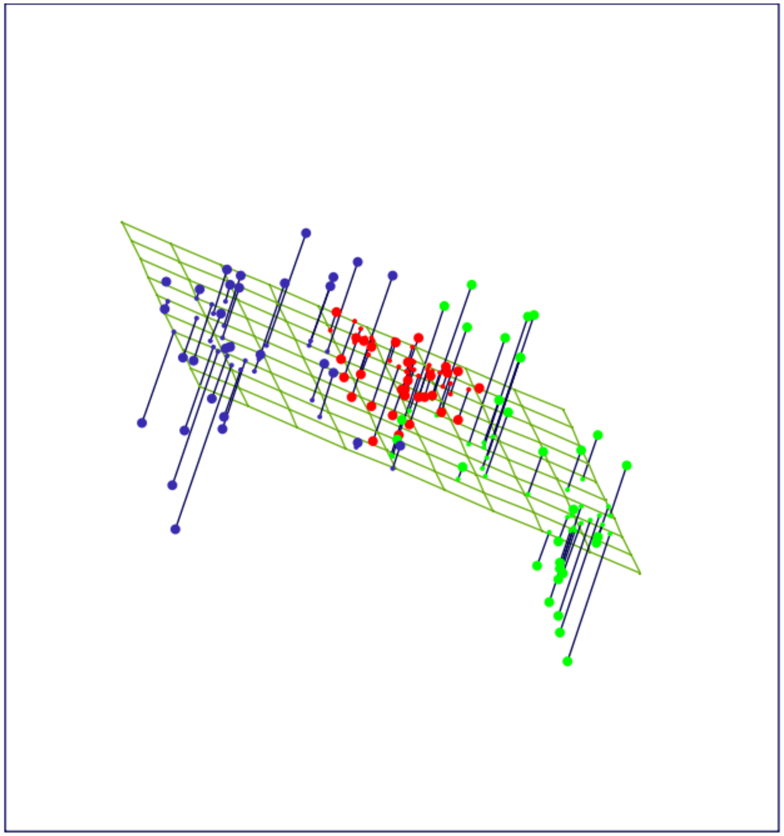
$$\boldsymbol{\mu} = \overline{\boldsymbol{x}}, \qquad \boldsymbol{A} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$$
$$\boldsymbol{\theta}_i = \boldsymbol{A}^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})$$

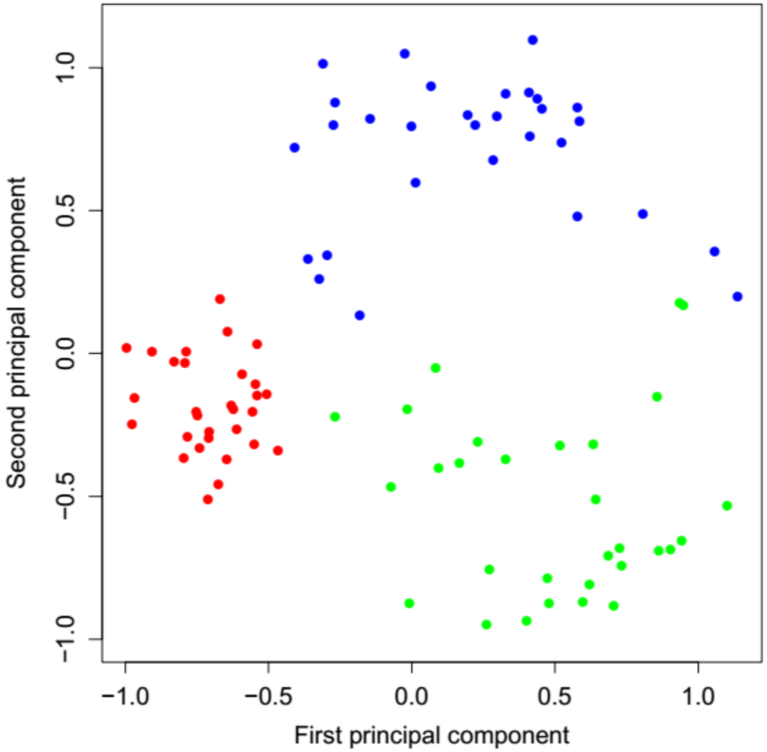
Terminology and Concepts

- Principal components
  - $\boldsymbol{\theta}^{(j)} = j$th principal component
- Principal eigenvectors/directions
  - $\boldsymbol{u}_j = j$th principal eigenvector/direction
- Reconstruction of $\boldsymbol{x}_i$
  - $\widehat{\boldsymbol{x}}_i = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{\theta}_i$

PCA

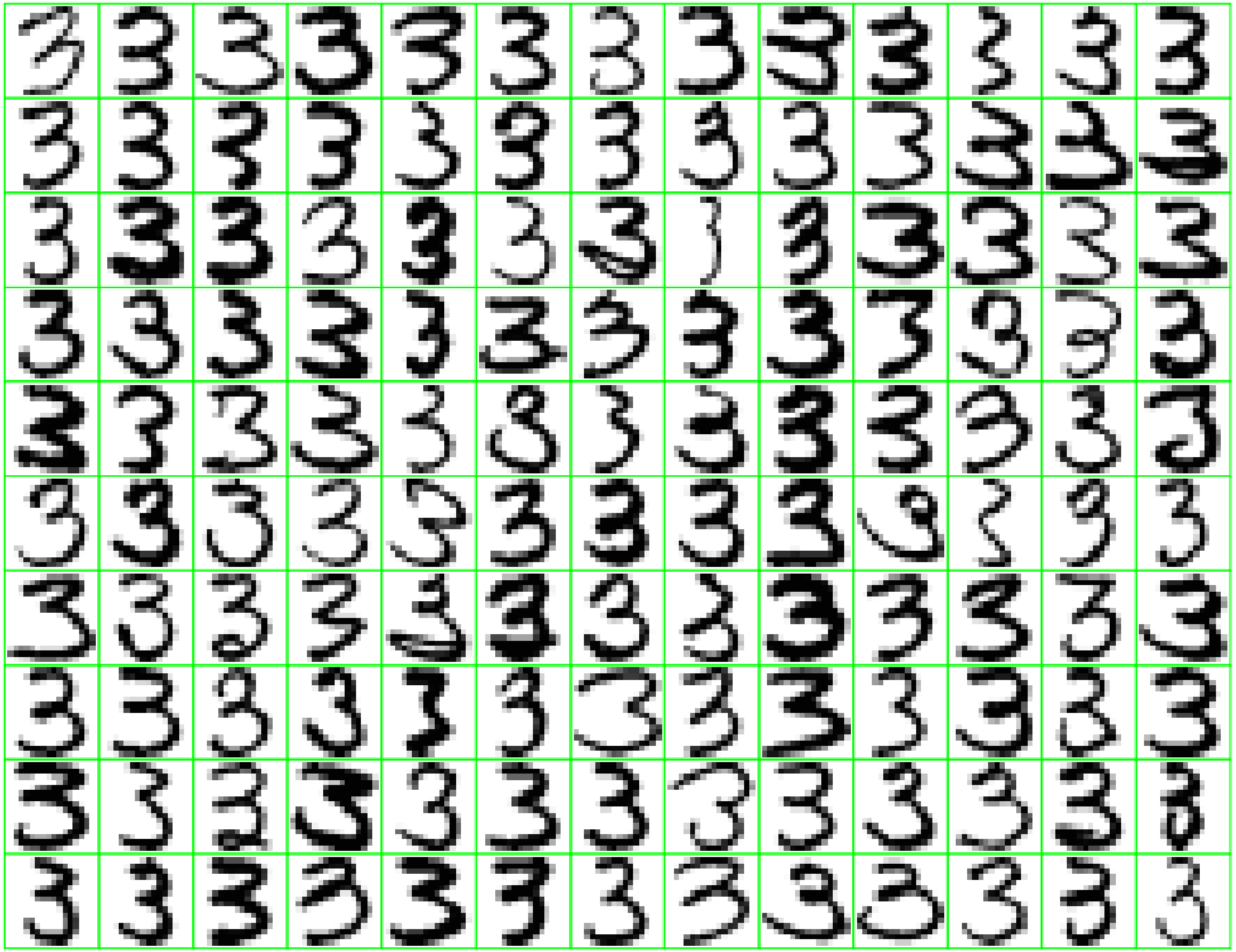
- **Example:** $d = 3, k = 2$





Example: Handwritten Digits

- Training data



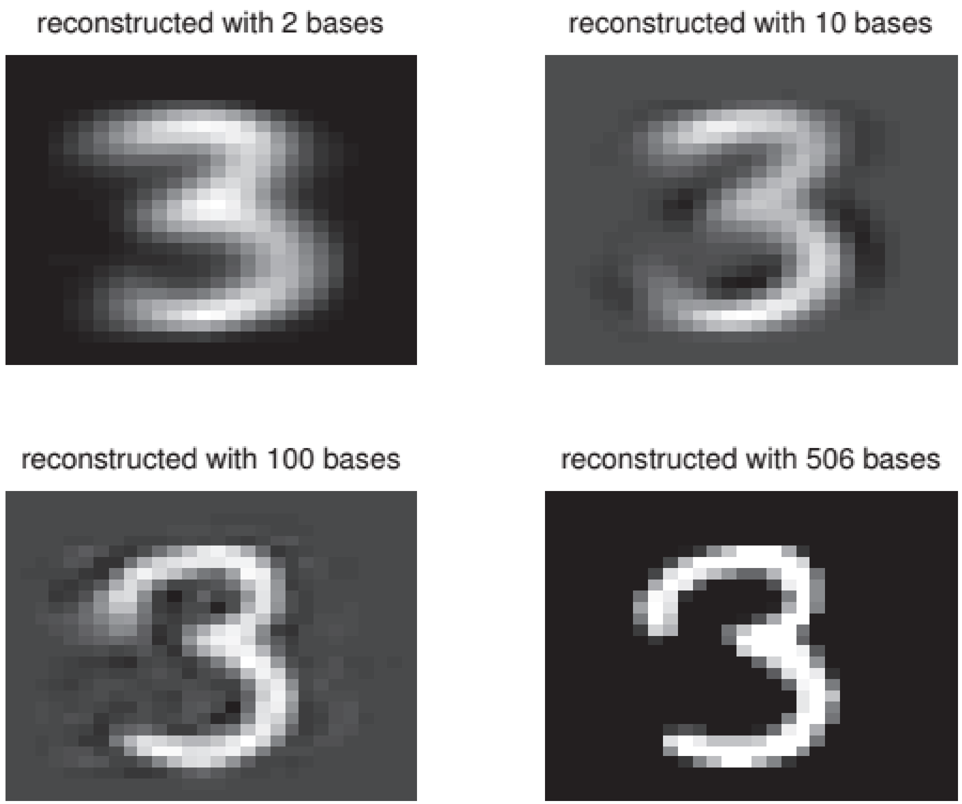
Example: Handwritten Digits

- Reconstruction

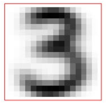
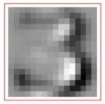
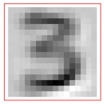
k=2

k=10

k=100



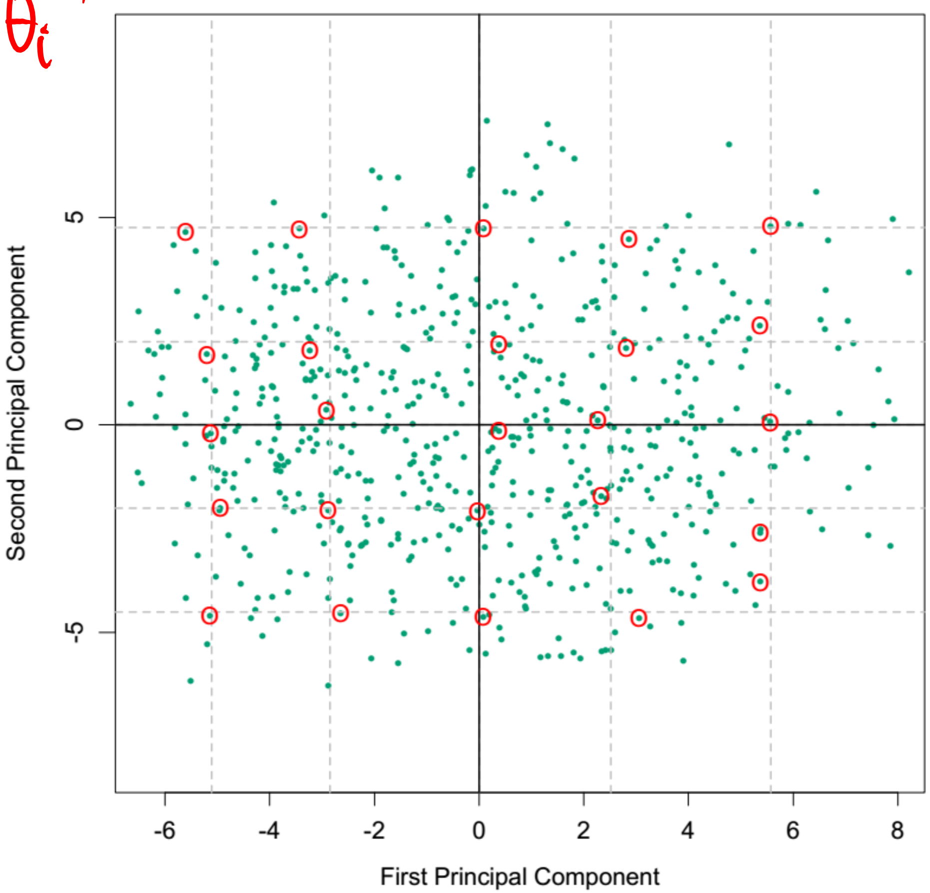
$$\widehat{\boldsymbol{x}}_i = \boldsymbol{\mu} + \overbrace{\theta_i^{(1)}\boldsymbol{u}_1 + \theta_i^{(2)}\boldsymbol{u}_2}^{A\theta} =$$

 $+\,\theta_i^{(1)}$  $+\,\theta_i^{(2)}$ 

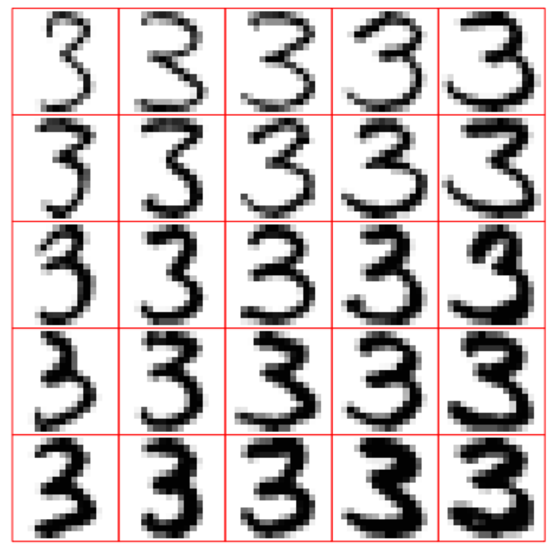
Example: Handwritten Digits

- First two PCs

$\theta_i^{(2)}$



$\theta_i^{(1)}$



Connection to Projections

- Suppose $\overline{\boldsymbol{x}} = \mathbf{0}$
- Recall that due to orthonormality, the projection matrix is $\boldsymbol{A}\boldsymbol{A}^T$
- The rank-k approximation to $\boldsymbol{x}_i$ is

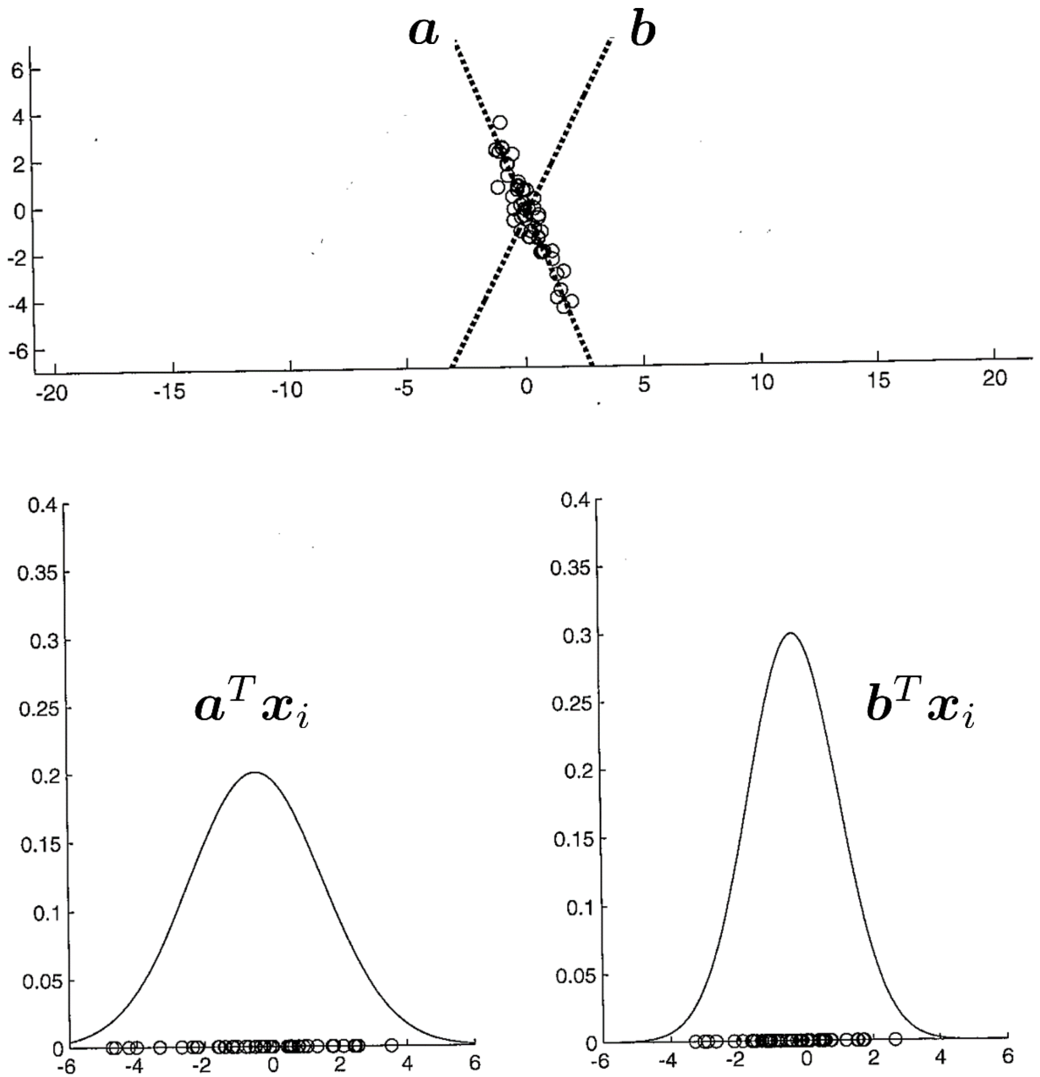
$$\widehat{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{\theta}_i = \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{x}_i$$

$$= \sum_{j=1}^{k} \boldsymbol{u}_j \theta_i^{(j)}$$

$$\theta_i^{(j)} = \boldsymbol{u}_j^T \boldsymbol{x}_i$$

- Intuition:
  - Columns of $\boldsymbol{A}$ define a k dimensional coordinate system for $\langle \boldsymbol{A} \rangle$
  - $\boldsymbol{\theta}_i = \boldsymbol{A}^T \boldsymbol{x}_i$ are the coordinates of $\widehat{\boldsymbol{x}}_i$ in the subspace

Connection to Projections



Maximum Variance Perspective

- Suppose $\bar{\boldsymbol{x}} = \mathbf{0}$
- Let $\boldsymbol{X}$ be a random vector of which $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are realizations
- Goal: find the unit vector $\boldsymbol{a}_1 \in \mathbb{R}^d$ ($\|\boldsymbol{a}_1\| = 1$) which maximizes the sample variance of

$$\theta^{(1)} = \boldsymbol{a}_1^T \boldsymbol{X}$$

- Sample mean of $\theta^{(1)}$:

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{a}_1^T \boldsymbol{x}_i = \boldsymbol{a}_1^T \left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \right) = \boldsymbol{a}_1^T \bar{\boldsymbol{x}}$$

Maximum Variance Perspective

- Sample variance of $\theta^{(1)}$:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{a}_1^T\boldsymbol{x}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{a}_1^T\boldsymbol{x}_i\right)\left(\boldsymbol{x}_i^T\boldsymbol{a}_1\right)$$

$$= \boldsymbol{a}_1^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{a}_1$$

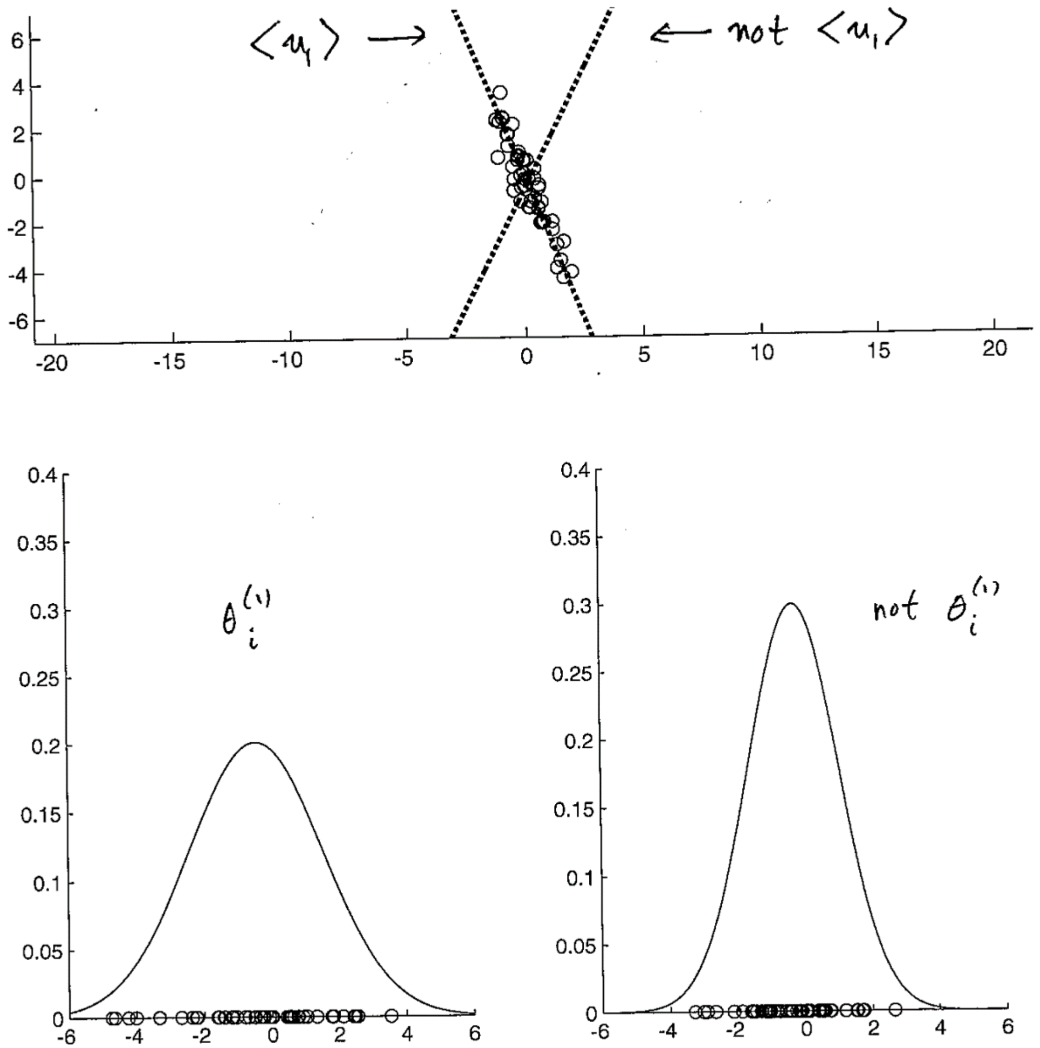
$$= \boldsymbol{a}_1^T S \boldsymbol{a}_1$$

- The solution of

$$\max_{\boldsymbol{a}_1:\|\boldsymbol{a}_1\|=1} \boldsymbol{a}_1^T S \boldsymbol{a}_1$$

is $\boldsymbol{u}_1$ (the first eigenvector of S). Thus $\theta^{(1)} = \boldsymbol{u}_1^T\boldsymbol{X}$.

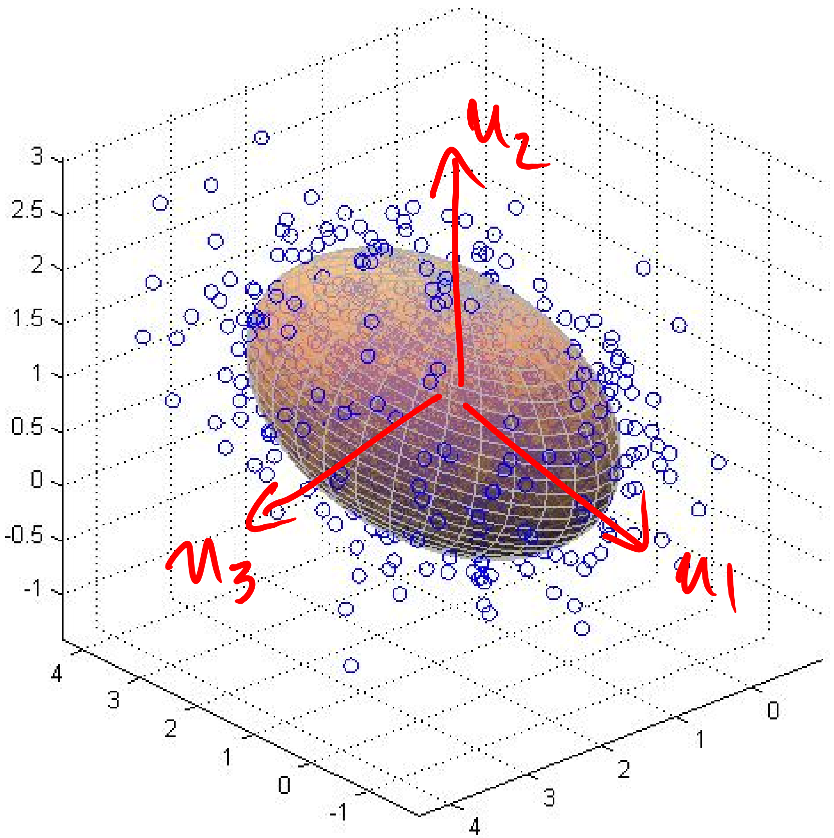
Maximum Variance Perspective



Maximum Variance Perspective

- More generally, we have the following result:
- **Theorem:** Let $\theta^{(k)} = \boldsymbol{a}_k^T \boldsymbol{X}$ and $\mathrm{var}(\theta^{(k)}) = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{a}_k^T \boldsymbol{x}_i)^2$. A vector $\boldsymbol{a}_k$ that maximizes $\mathrm{var}(\theta^{(k)})$ subject to
    - $\|\boldsymbol{a}_k\| = 1$
    - $\boldsymbol{a}_k \perp \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k-1}$

  is $\boldsymbol{a}_k = \boldsymbol{u}_k$.
- What is the optimized variance of $\theta^{(k)}$?



Optimized Variance

$$var_{samp}\left(\boldsymbol{u}_j^T \boldsymbol{X}\right) = \boldsymbol{u}_j^T S \boldsymbol{u}_j$$

$$= \boldsymbol{u}_j^T U \Lambda U^T \boldsymbol{u}_j$$

$$= [0, \dots, 0,1,0, \dots, 0]\Lambda \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \lambda_j$$

Selecting k

- It can be shown that the optimal objective function for PCA is

$$\min_{\substack{\boldsymbol{\mu},\boldsymbol{A},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n \\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{\theta}_i\|^2 = n(\lambda_{k+1} + \cdots + \lambda_d)$$

- When $k = 0$, this specializes to

$$\min_{\boldsymbol{\mu}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu}\|^2 = n(\lambda_1 + \cdots + \lambda_d)$$

  which we call the total variation of the data.

- Common heuristic: choose smallest k s.t.

  $\frac{\lambda_1+\cdots+\lambda_k}{\lambda_1+\cdots+\lambda_d} =$ % of variance explained by the top k PCs

  - Common threshold is 95%

Connection to SVD

- Assume $\overline{\boldsymbol{x}} = \mathbf{0}$
- Data matrix $(d \times n)$ $X = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$
  - Recall that $S = \frac{1}{n} X X^T$
- The singular value decomposition (SVD) of X is

$$X = U \Sigma V^T$$

  where U $(d \times d)$ and V $(n \times n)$ are orthogonal matrices, and $\Sigma = diag\left(\sigma_1, \dots, \sigma_{\min\{d,n\}}\right)$ is $d \times n$
- Then $u_j = j$th left singular vector $= j$th principal component
  - Also, $\lambda_j = \frac{1}{n} \sigma_j^2$

Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS)

- Another dimensionality reduction method, typically focused on visualization
- **Example**: Suppose we have the following distance matrix between major cities in the US (3D distances)

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0

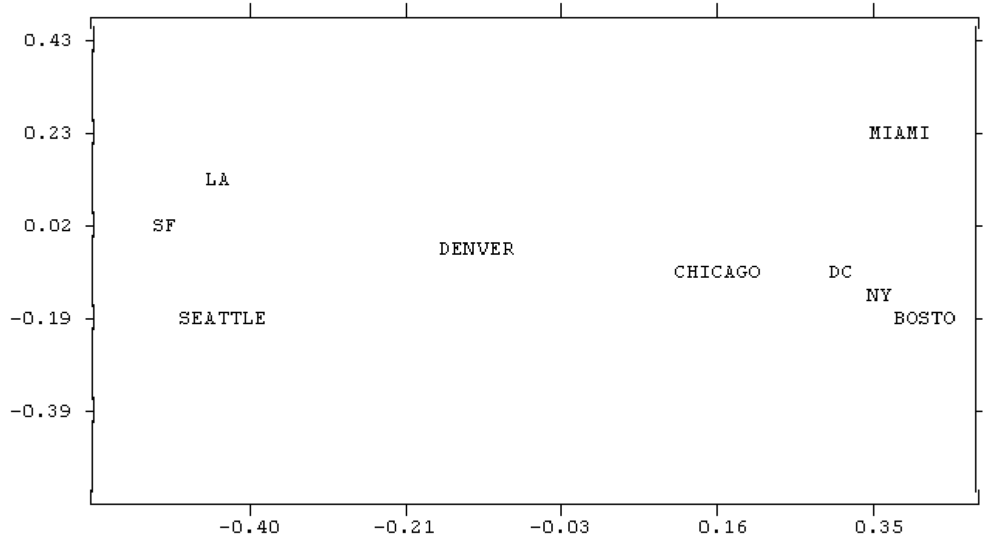
- Can we create a 2D representation of this dataset?
- Use MDS

MDS Example

- Distance matrix

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0

- MDS-generated map



MDS

- MDS works by minimizing a loss function between the distances in the high-dimensional space and the distances in the low-dimensional space
- Setup: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ are the measurements in the original, high –dimensional space
- Find representations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ $(k < d)$ such that the loss function is minimized
- Different loss functions give you different variations on MDS

Metric MDS

- $d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ (pairwise Euclidean distance)
- Metric MDS minimizes the "Stress" function:

$$Stress_M(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij} - \|\boldsymbol{y}_i - \boldsymbol{y}_j\|\right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2} \right)^{1/2}$$

  - The $\boldsymbol{y}_i$s are the variables that are chosen to minimize the stress.
  - Other kinds of distances can be chosen to give different variations

Choosing the dimension k

- If the stress is zero, then there is no distortion of the distances
  - Thus smaller stress ⇒ a better representation
  - Stress decreases as the dimension k increases (less distortion required with higher dimensions)
- However, if the data are noisy, some "distortion" may be ok
- **Example**: distances from buildings in NYC measured from center of the roof
  - Clearly a 3D dataset
  - But a 3D MDS representation may have nonzero stress/loss

Choosing the dimension k

- How do we know what k should be?
- No surefire answer…
- Common approach is to look at a "scree plot"
- "Elbows" aren't always obvious and so other approaches may be needed
  - E.g., Shepard diagrams (plot of $\boldsymbol{x}_i$ distances vs $\boldsymbol{y}_i$ distances) may be useful

Nonmetric MDS

- Recall the stress function for metric MDS:

$$Stress_M(\boldsymbol{y}_1, \dots, \boldsymbol{y}_n) = \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij} - \left\| \boldsymbol{y}_i - \boldsymbol{y}_j \right\| \right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2} \right)^{1/2}$$

- Now let d_{ij} be a measure of dissimilarity (could be Euclidean distance) between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
- Let f be some monotonic function
- Nonmetric MDS minimizes the following:

$$Stress_{NM}(f, \boldsymbol{y}_1, \dots, \boldsymbol{y}_n) = \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(f\left(d_{ij}\right) - \left\| \boldsymbol{y}_i - \boldsymbol{y}_j \right\| \right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \boldsymbol{y}_i - \boldsymbol{y}_j \right\|^2} \right)^{1/2}$$

Classical MDs

- A special case of metric MDS
- Let $d_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|$
- Let $D^{(2)} = \left[d_{ij}^2\right]$ be the matrix of squared distances
- Double center the matrix: $B = -\frac{1}{2} J D^{(2)} J$
  - $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ is an n-dimensional vector of ones)
- Define b_{ij} to be the i, jth entry of B
- Classical MDS minimizes the “Strain” function:

$$Strain(y_1, \dots, y_n) = \left(\frac{\sum_{i,j}\left(b_{ij} - \langle y_i, y_j\rangle\right)^2}{\sum_{i,j} b_{ij}^2}\right)^{1/2}$$

Classical MDS

- Classical MDS minimizes the "Strain" function:

$$Strain(y_1, \dots, y_n) = \left(\frac{\sum_{i,j} \left(b_{ij} - \langle y_i, y_j \rangle \right)^2}{\sum_{i,j} b_{ij}^2} \right)^{1/2}$$

- It turns out, this can be solved using eigenvalue decomposition
  - Efficient computation!
  - Strictly assumes that the distances are Euclidean (may be too restrictive sometimes)
- Can be shown that classical MDS and PCA are equivalent (recover one from the other)

Further Reading

- ISL Section 10.2
- ESL Sections 3.4.1, 14.5, and 14.8
- MDS figures: http://www.analytictech.com/borgatti/mds.htm