CSCI 5521: Introduction to Machine Learning

Dimension Reduction (Chpt 6)

Rui Kuang

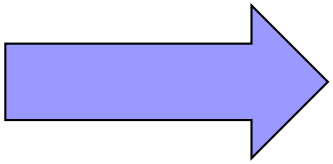
Department of Computer Science and Engineering

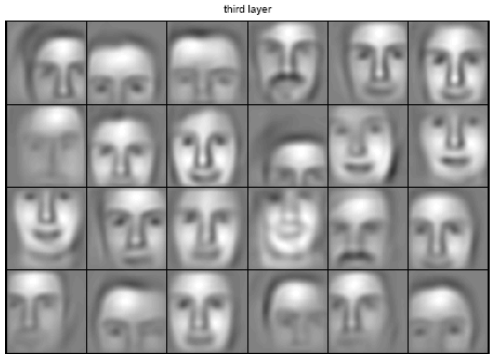
University of Minnesota

Dimensionality Reduction: Face Recognition

- Learning a compact representation of images with millions of pixels.

Millions of pixels



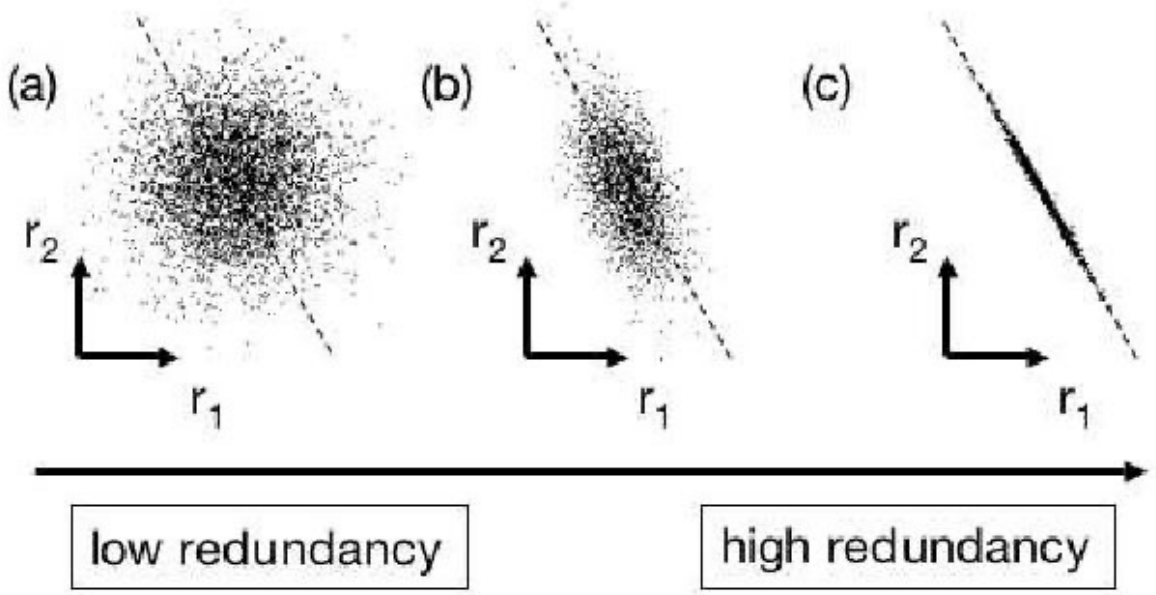


Tens of eigen-faces

Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Less parameters
- Saves the cost of observing the features
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions

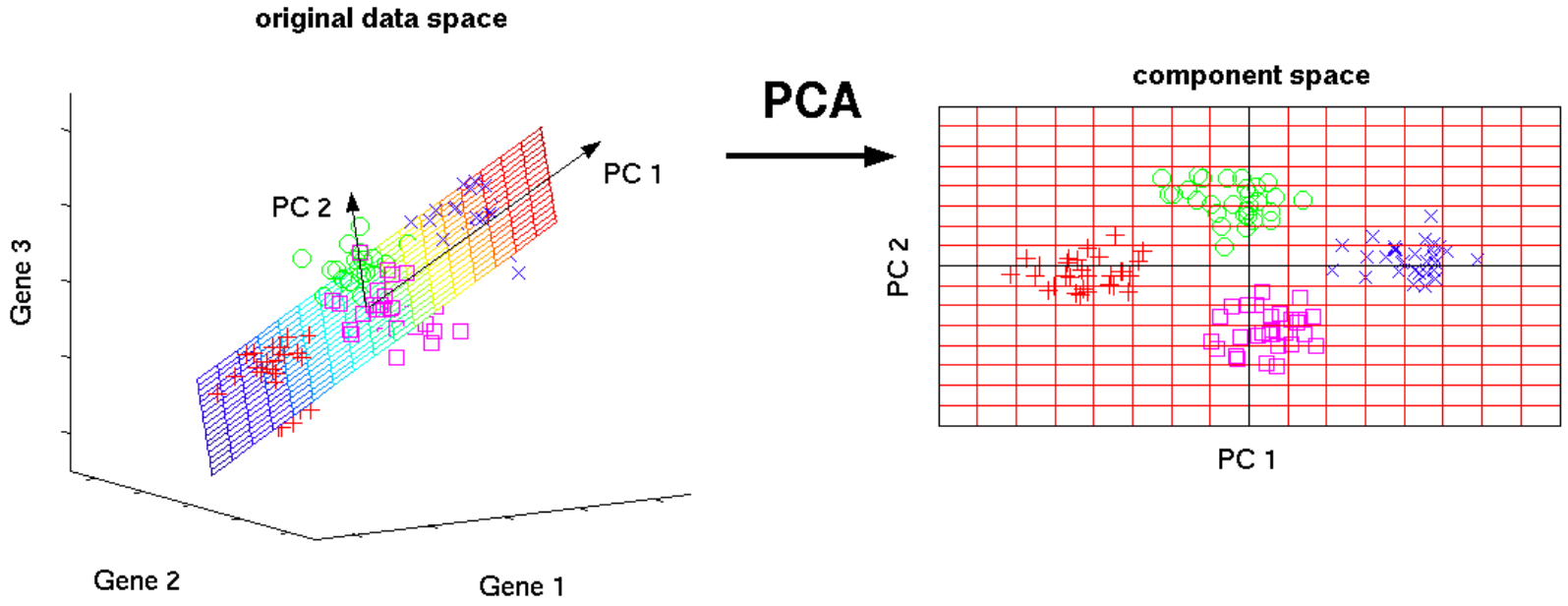
Principal Components Analysis (PCA)



- High dimensional data are often noisy and redundant.
- We want to identify the key directions to have a compressed representation that are noise free and can be visualized
- One way is to find the principle components.

Principal Components Analysis (PCA)

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected there, information loss is minimized.
- The projection of $\boldsymbol{x}$ on the direction of $\boldsymbol{w}$ is: $z = \boldsymbol{w}^T\boldsymbol{x}$

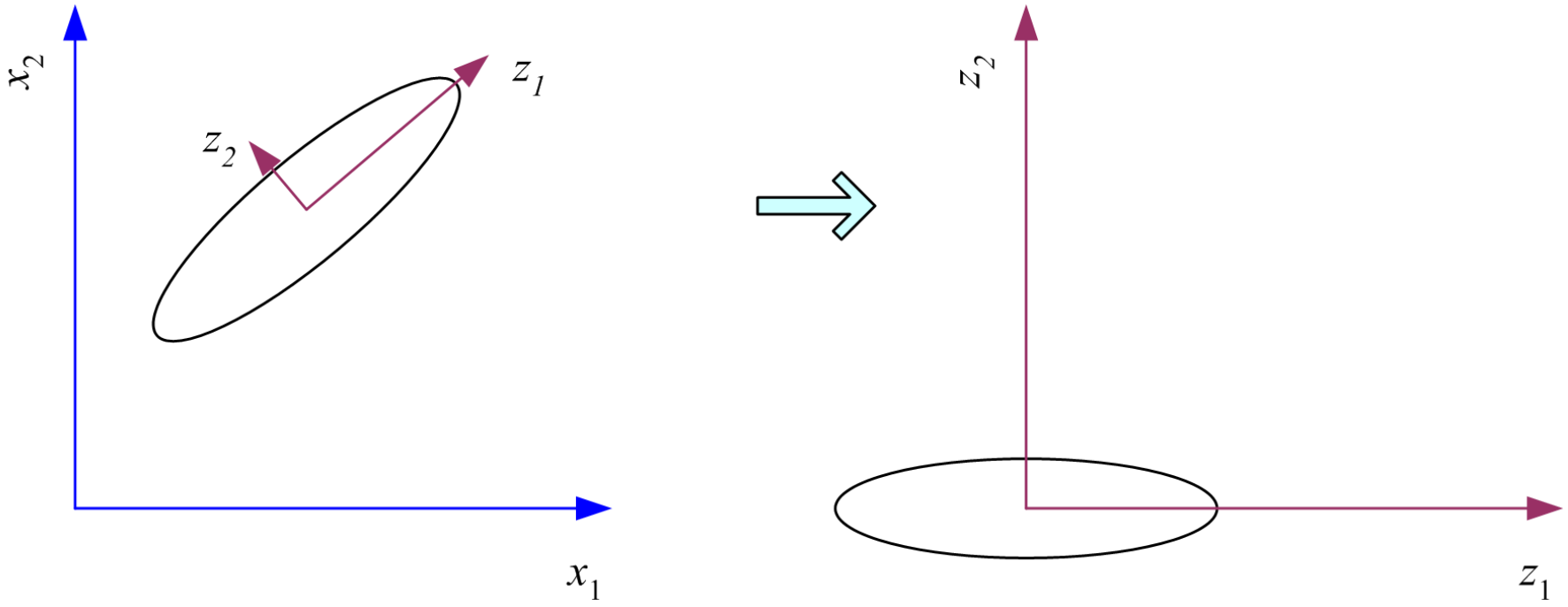


What PCA does

$$\boldsymbol{z} = \mathbf{W}^T(\boldsymbol{x} - \boldsymbol{m})$$

where the columns of $\mathbf{W}$ are the eigenvectors of $\sum$, and $\boldsymbol{m}$ is sample mean

Centers the data at the origin and rotates the axes



Principal Components Analysis (PCA)

- Find $\boldsymbol{w}$ such that Var(z) is maximized

$$\begin{aligned}\text{Var}(z) = \text{Var}(\boldsymbol{w}^T\boldsymbol{x}) &= E[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})^2] \\ &= E[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})] \\ &= E[\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{w}] \\ &= \boldsymbol{w}^T E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{w} = \boldsymbol{w}^T \sum \boldsymbol{w}\end{aligned}$$

where $\text{Var}(\boldsymbol{x}) = E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T] = \sum$

- Maximize Var(z) subject to $||\boldsymbol{w}||=1$

$$\max_{\mathbf{w}_1} \mathbf{w}_1^T \Sigma \mathbf{w}_1 - \alpha\left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right)$$

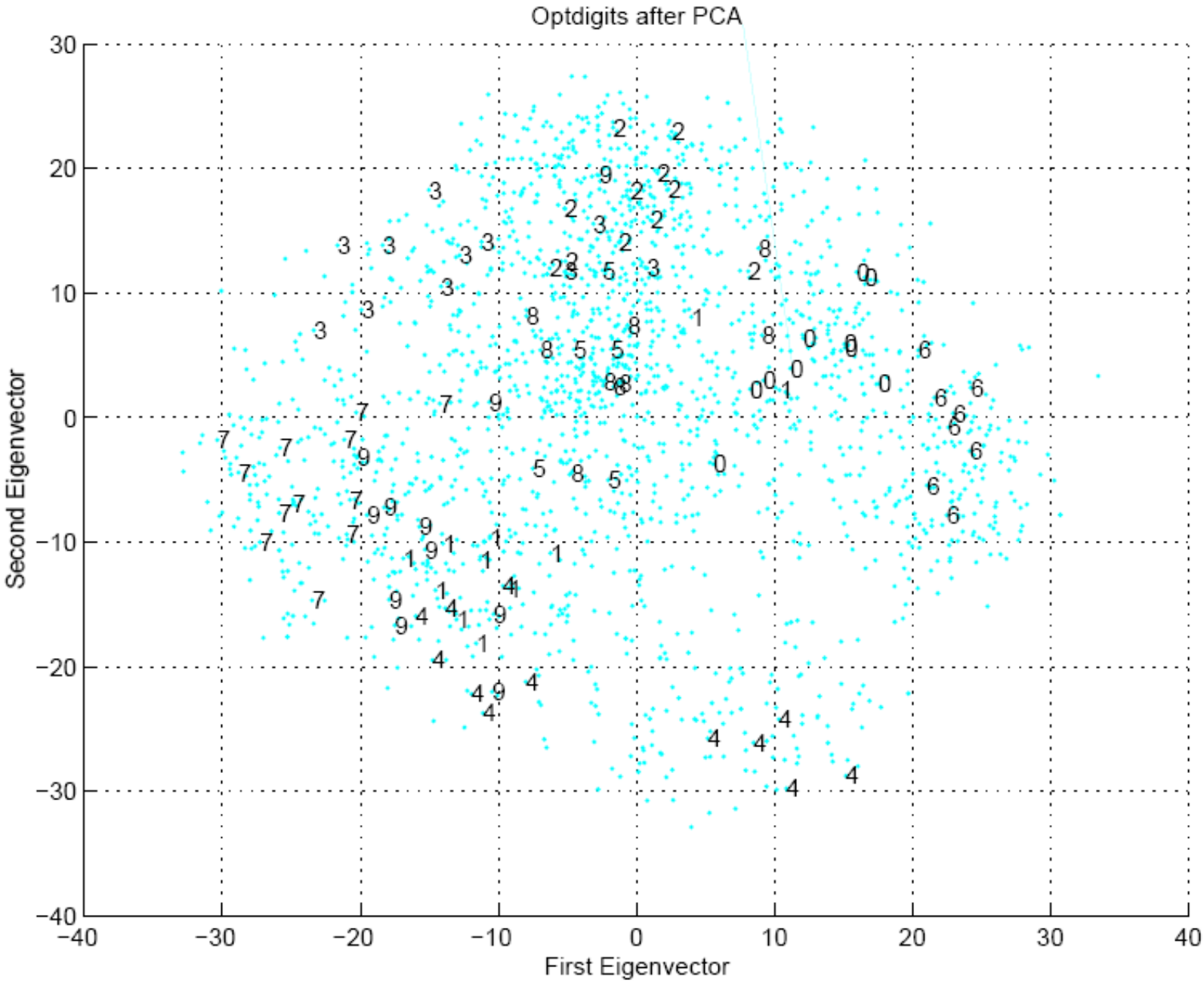
$\sum \boldsymbol{w}_1 = \alpha \boldsymbol{w}_1$ that is, $\boldsymbol{w}_1$ is an eigenvector of $\sum$
Choose the one with the largest eigenvalue for Var(z) to be max

- Second principal component: Max Var(z_2), s.t., $||\boldsymbol{w}_2||=1$ and orthogonal to $\boldsymbol{w}_1$

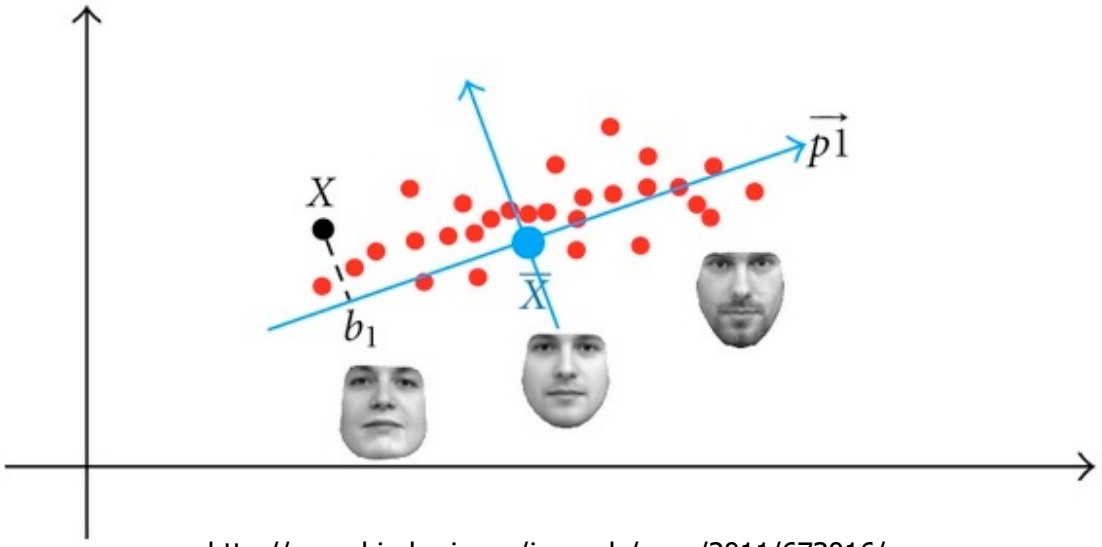
$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \Sigma \mathbf{w}_2 - \alpha\left(\mathbf{w}_2^T \mathbf{w}_2 - 1\right) - \beta\left(\mathbf{w}_2^T \mathbf{w}_1 - 0\right)$$

$\sum \boldsymbol{w}_2 = \alpha \boldsymbol{w}_2$ that is, $\boldsymbol{w}_2$ is another eigenvector of $\sum$

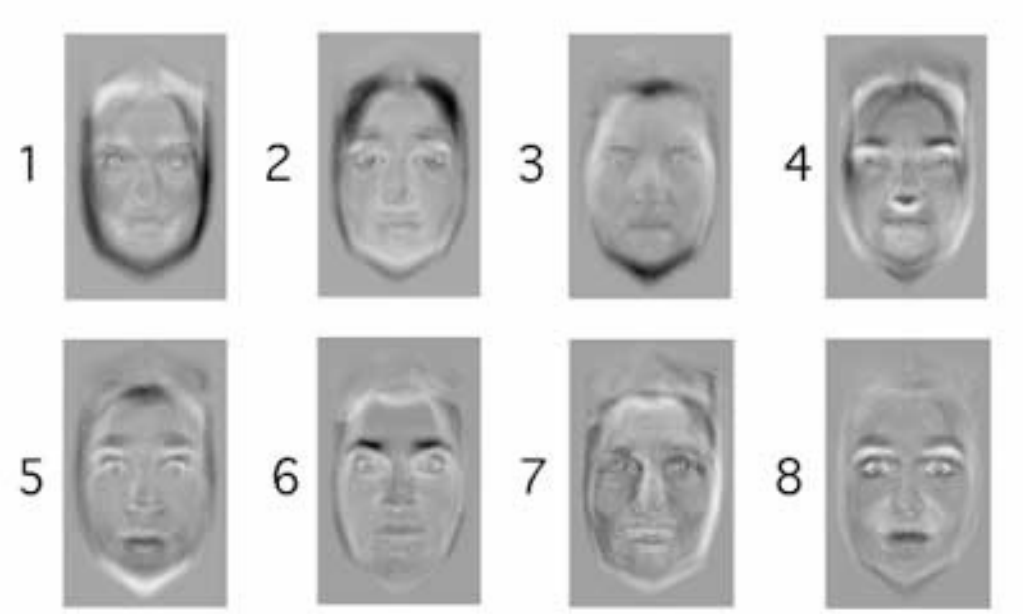
and so on.


Optdigits after PCA
Second Eigenvector
First Eigenvector
30
20
10
0
−10
−20
−30
−40
−40
−30
−20
−10
0
10
20
30
40

Face Recognition



http://www.hindawi.com/journals/aans/2011/673016/



http://mathdesc.fr/documents/facerecog/PerceptionFacialExpression.htm

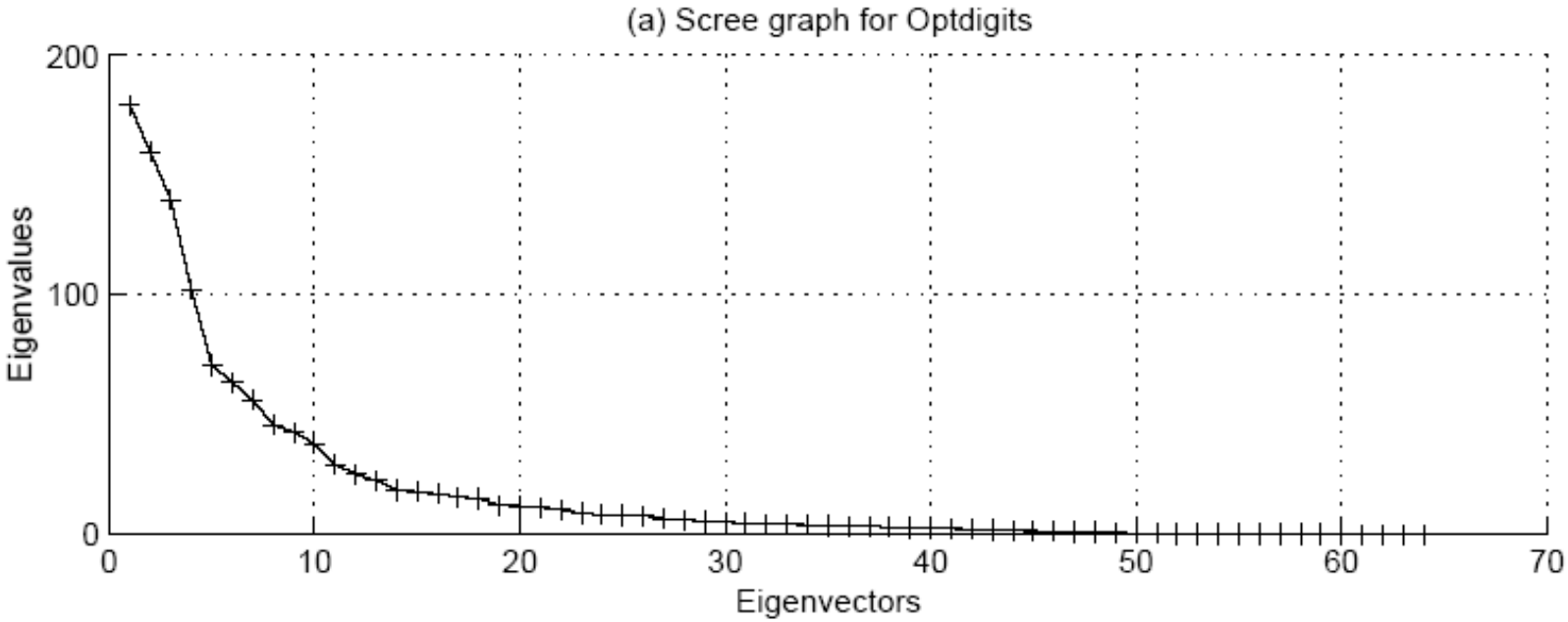
How to choose k ?

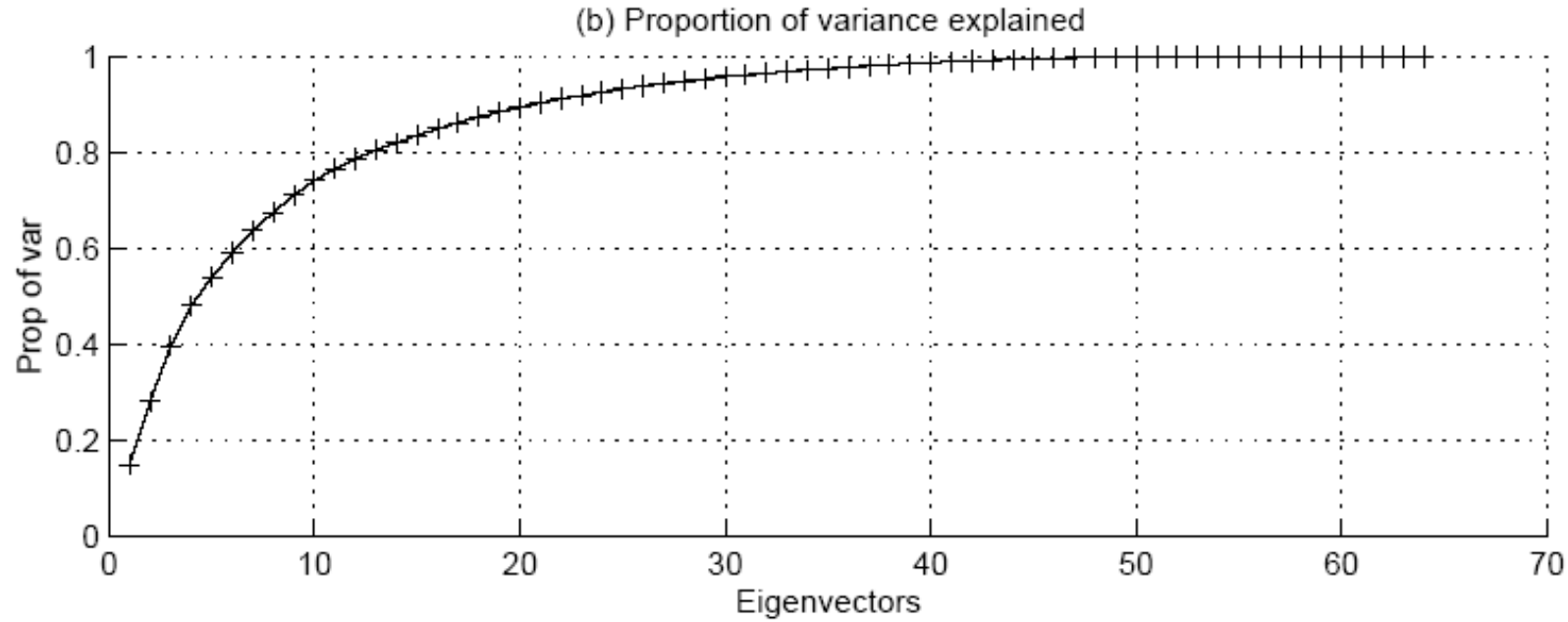
- Proportion of Variance (PoV) explained

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when λ_i are sorted in descending order

- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs *k*, stop at "elbow"


(a) Scree graph for Optdigits
Eigenvalues
200
100
0
0
10
20
30
40
50
60
70
Eigenvectors

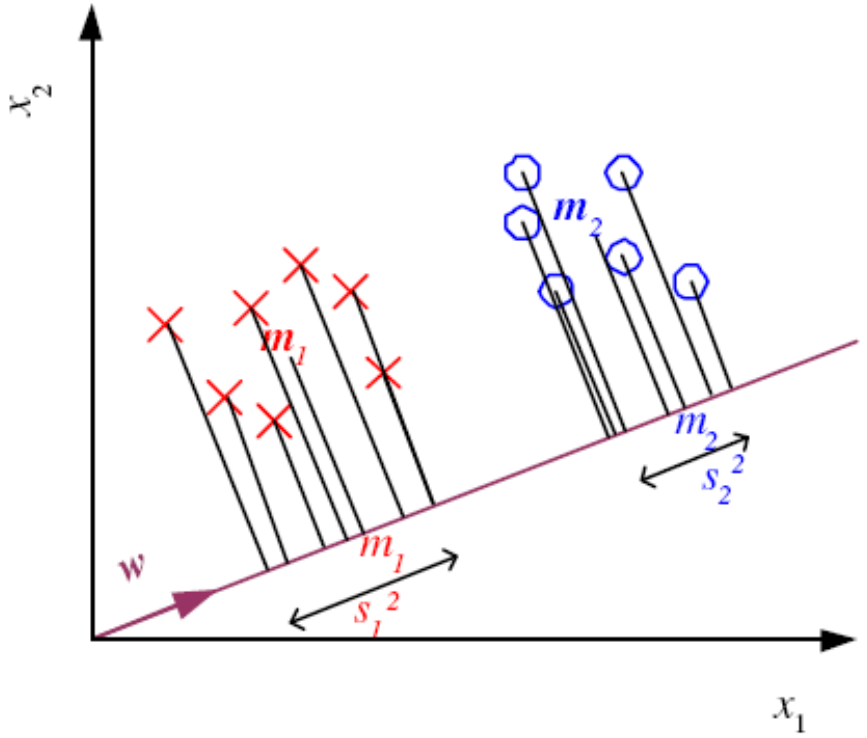

(b) Proportion of variance explained
Prop of var
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
70
Eigenvectors

PCA Discussions

- Linearity: Linear rotation of the original space
- Gaussian assumption: mean and variance are enough to characterize the noise and redundancy.
- We choose the principal components to be orthogonal, but they don't have to be.
- Often used with clustering algorithms to cluster high-dimensional data

Linear Discriminant Analysis

- Find a low-dimensional space such that when ***x*** is projected, classes are well-separated.
- Find ***w*** that maximizes



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

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} \qquad s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t$$

- Between-class scatter:

$$\left(m_1 - m_2\right)^2 = \left(\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2\right)^2$$
$$= \mathbf{w}^T\left(\mathbf{m}_1 - \mathbf{m}_2\right)\left(\mathbf{m}_1 - \mathbf{m}_2\right)^T\mathbf{w}$$
$$= \mathbf{w}^T\mathbf{S}_B\mathbf{w} \text{ where } \mathbf{S}_B = \left(\mathbf{m}_1 - \mathbf{m}_2\right)\left(\mathbf{m}_1 - \mathbf{m}_2\right)^T$$

- Within-class scatter:

$$s_1^2 = \sum_t \left(\mathbf{w}^T\mathbf{x}^t - m_1\right)^2 r^t$$
$$= \sum_t \mathbf{w}^T\left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T\mathbf{w}r^t = \mathbf{w}^T\mathbf{S}_1\mathbf{w}$$
$$\text{where } \mathbf{S}_1 = \sum_t \left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T r^t$$
$$s_1^2 + s_2^2 = \mathbf{w}^T\mathbf{S}_W\mathbf{w} \text{ where } \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Find ***w*** that max

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\left|\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$$

$$\frac{\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} (2(\mathbf{m}_1 - \mathbf{m}_2) - \frac{\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} 2S_W w) = 0$$

- LDA soln:

$$\mathbf{w} = c \cdot \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

K>2 Classes

- Within-class scatter:

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t \left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T$$

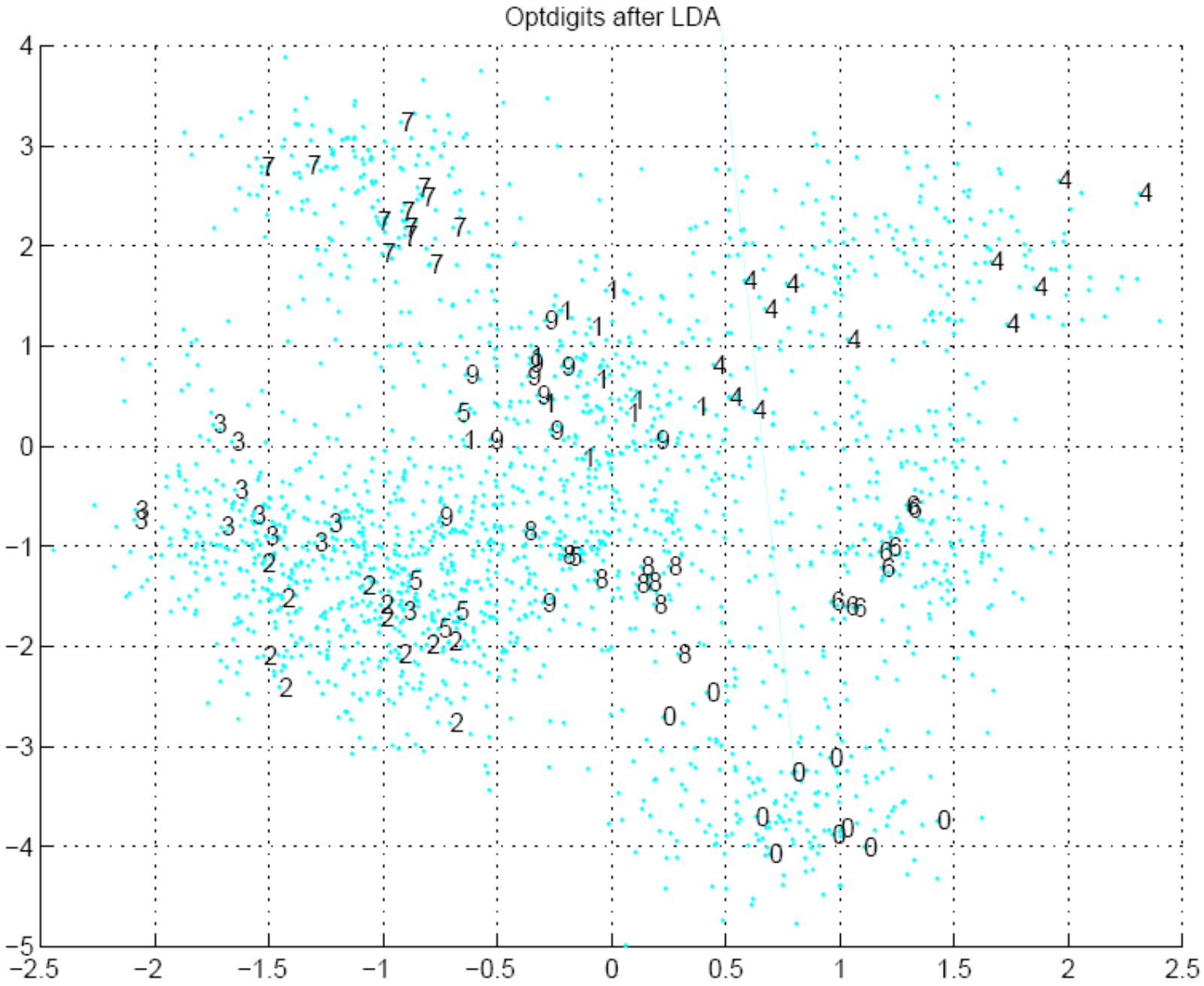
- Between-class scatter (among means):

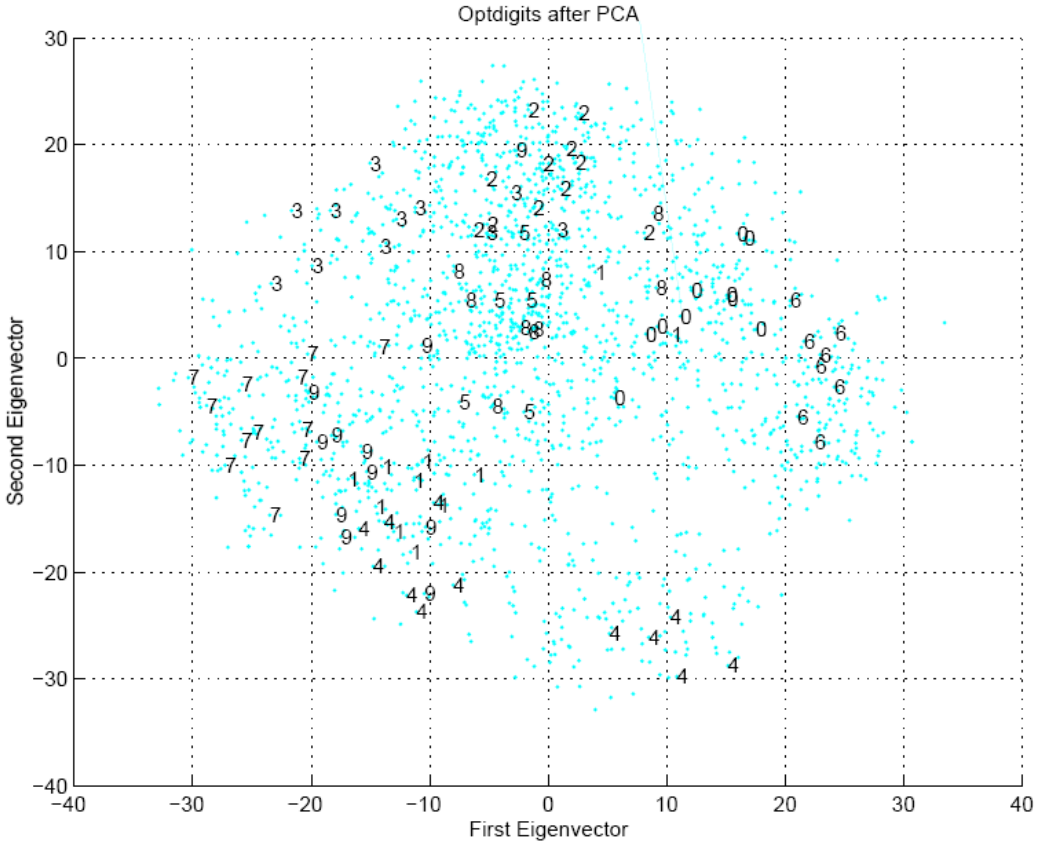
$$\mathbf{S}_B = \sum_{i=1}^{K} N_i \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{m}_i$$

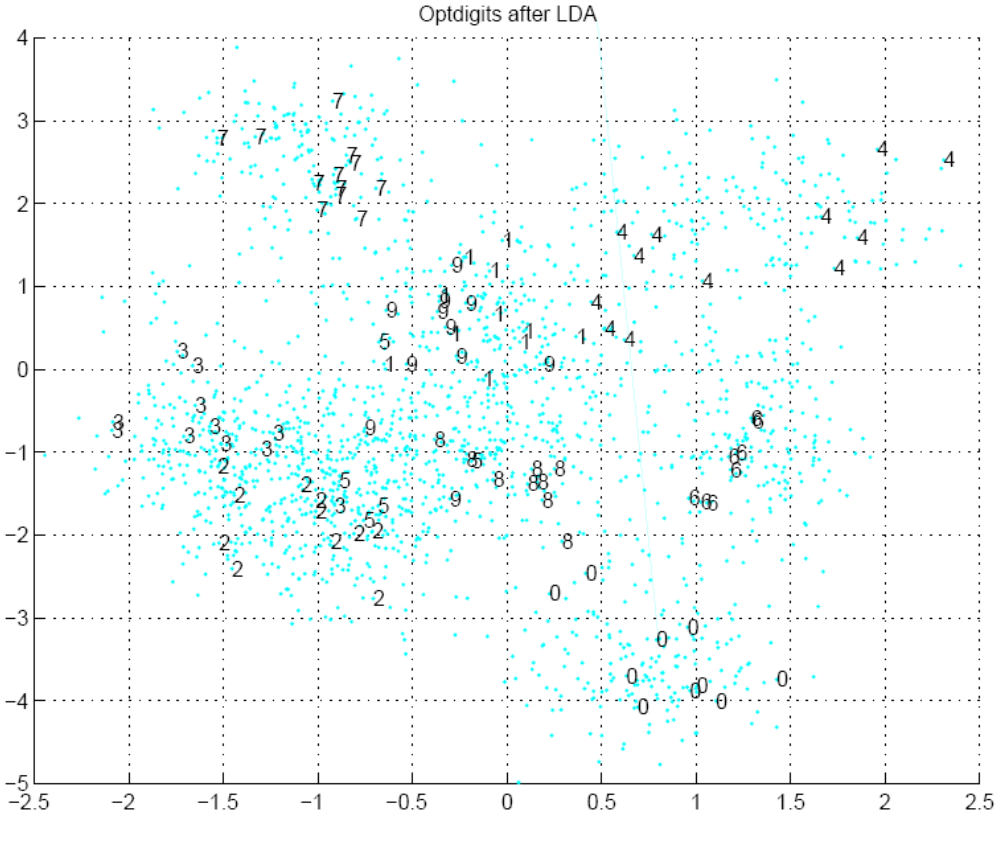
- Find **W** that max

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

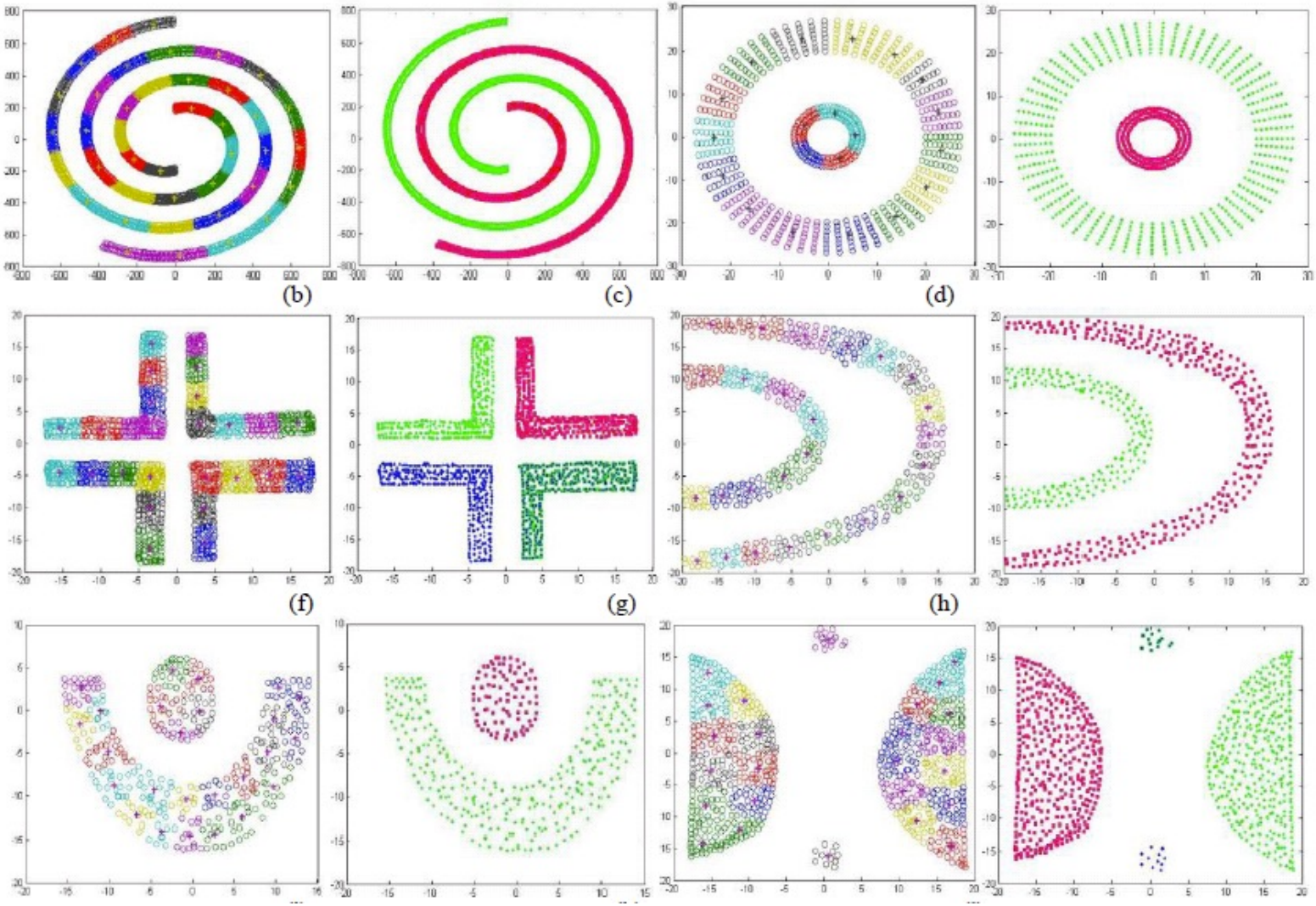
The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$
Maximum rank of K-1


Optdigits after LDA
4
3
2
1
0
−1
−2
−3
−4
−5
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5


Optdigits after PCA
Second Eigenvector
First Eigenvector


Optdigits after LDA

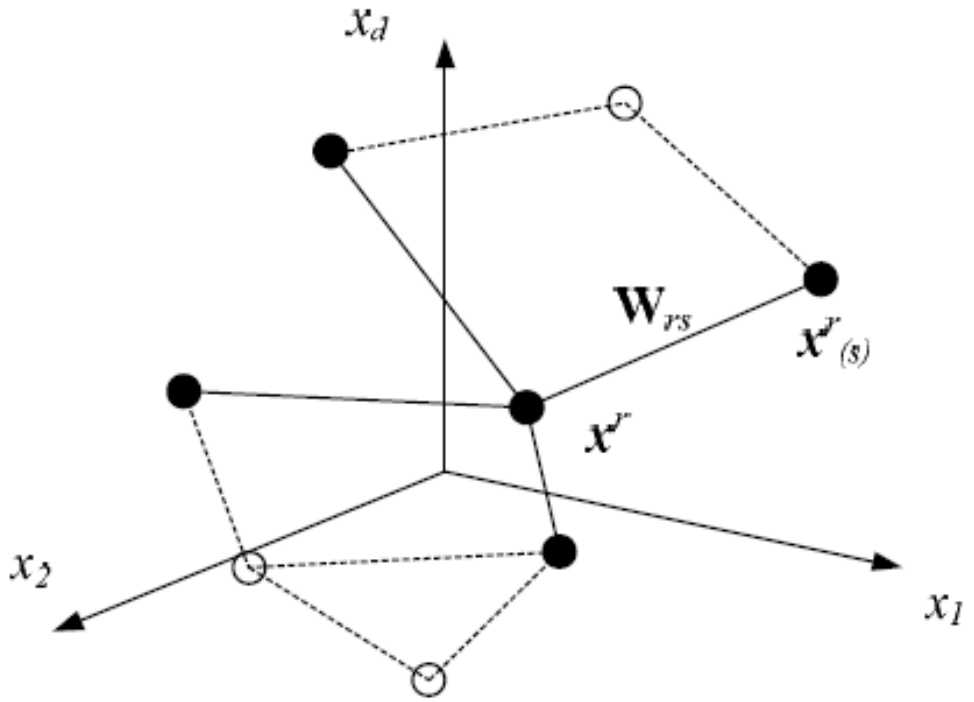
Projection of Non-linear Data



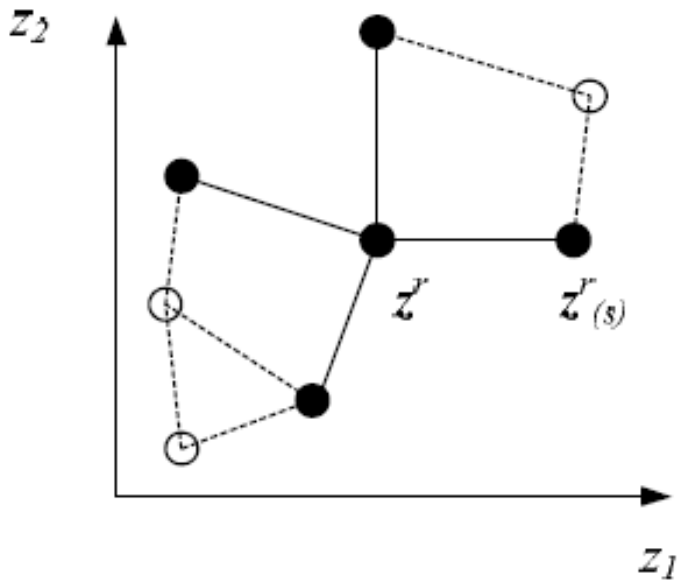
http://stackoverflow.com/

Locally Linear Embedding

- Mapping to a new space allowing linear embedding.
- Use overlapping hyperplanes to approximate the non-linear surface.

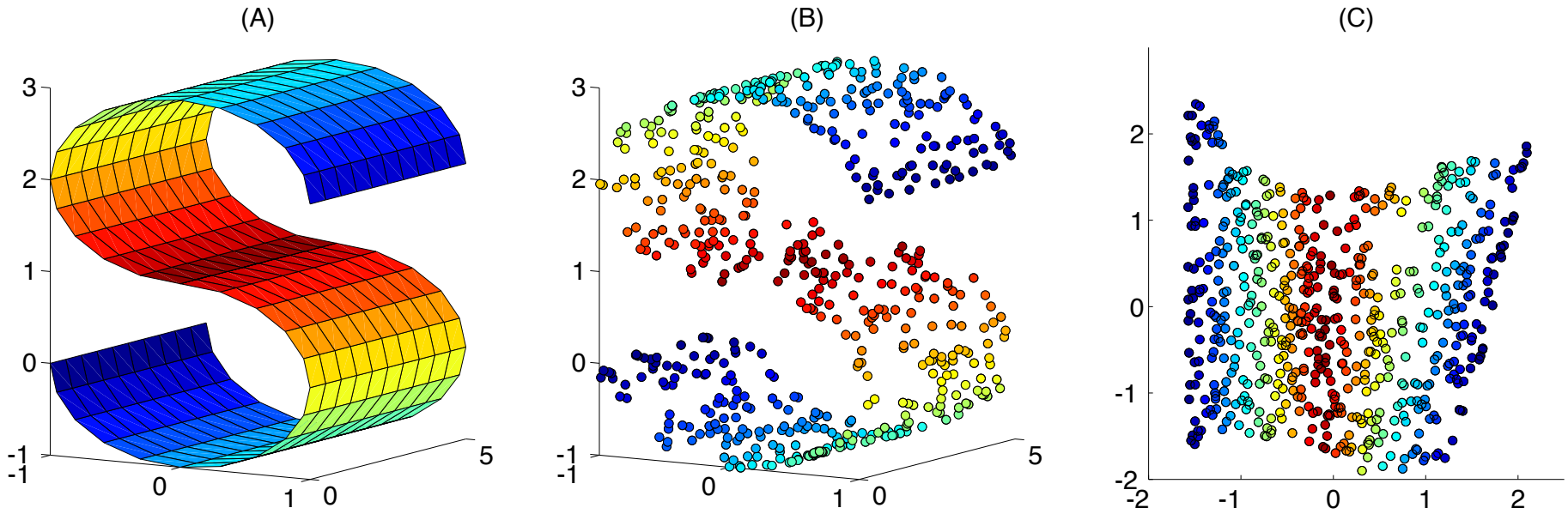


x space



z space

LLE on S Manifold



Saul et al., An Introduction to LLE

Locally Linear Embedding

1. Given $\boldsymbol{x}^r$ find its neighbors $\boldsymbol{x}^s_{(r)}$
2. Find $\mathbf{W}_{rs}$ that minimize

$$E(\mathbf{W} \mid X) = \sum_r \left\| \mathbf{x}^r - \sum_s \mathbf{W}_{rs} \mathbf{x}^s_{(r)} \right\|^2, \text{subj: } \sum_s W_{rs} = 1$$

3. Find the new coordinates $\boldsymbol{z}^r$ that minimize

$$E(\mathbf{z} \mid \mathbf{W}) = \sum_r \left\| z^r - \sum_s \mathbf{W}_{rs} z^s_{(r)} \right\|^2, \text{subj: } \sum_r z^r = 0 \text{ and } \sum_r (z^r)^T z^r = 1$$

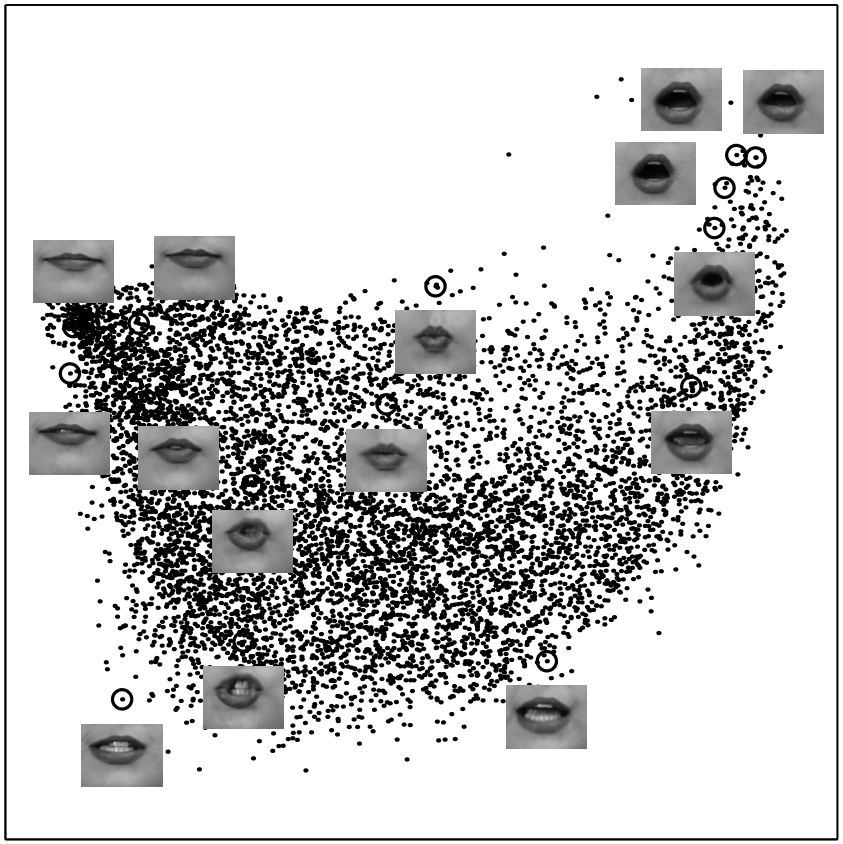
Locally Linear Embedding

$$E(\mathbf{z}\,|\,\mathbf{W}) = \sum_r \left\| z^r - \sum_s \mathbf{W}_{rs} z^s_{(r)} \right\|^2 ,, \text{subj: } \sum_r z^r = 0 \text{ and } \sum_r (z^r)^T z^r = 1$$

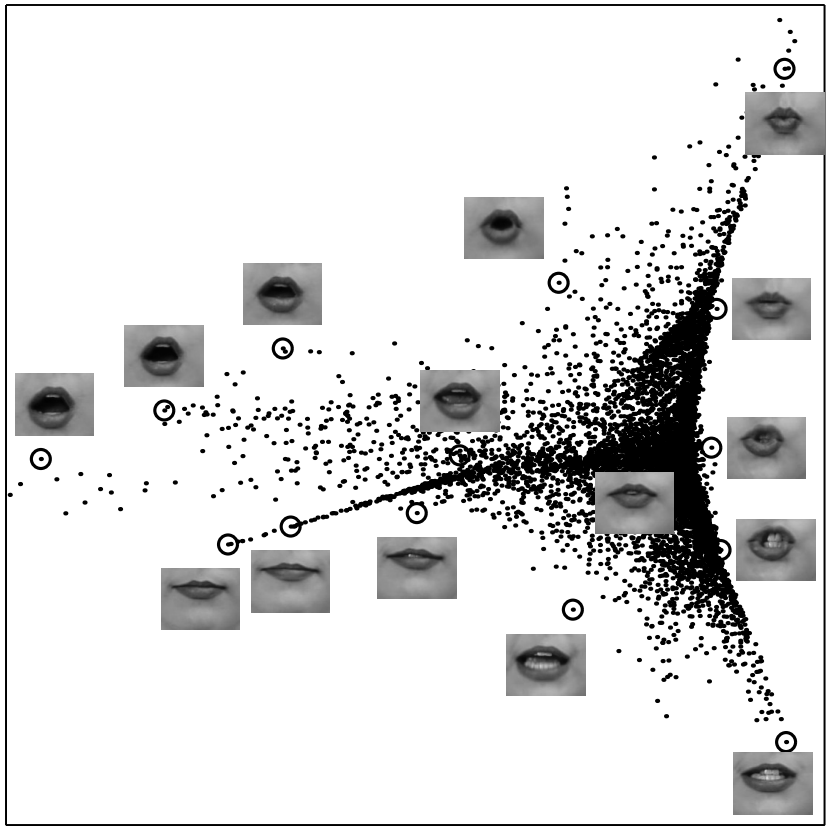
$$= \sum_{r,s} (\delta_{rs} - W_{rs} - W_{sr} + \sum_i W_{is} W_{ir})(z^r)^T z^s$$

- With n neighbors and d dimensions, d≤n-1. d is often a bit smaller.
- Take k+1 lowest eigenvectors and discard the first one.

LLE on Lip Images



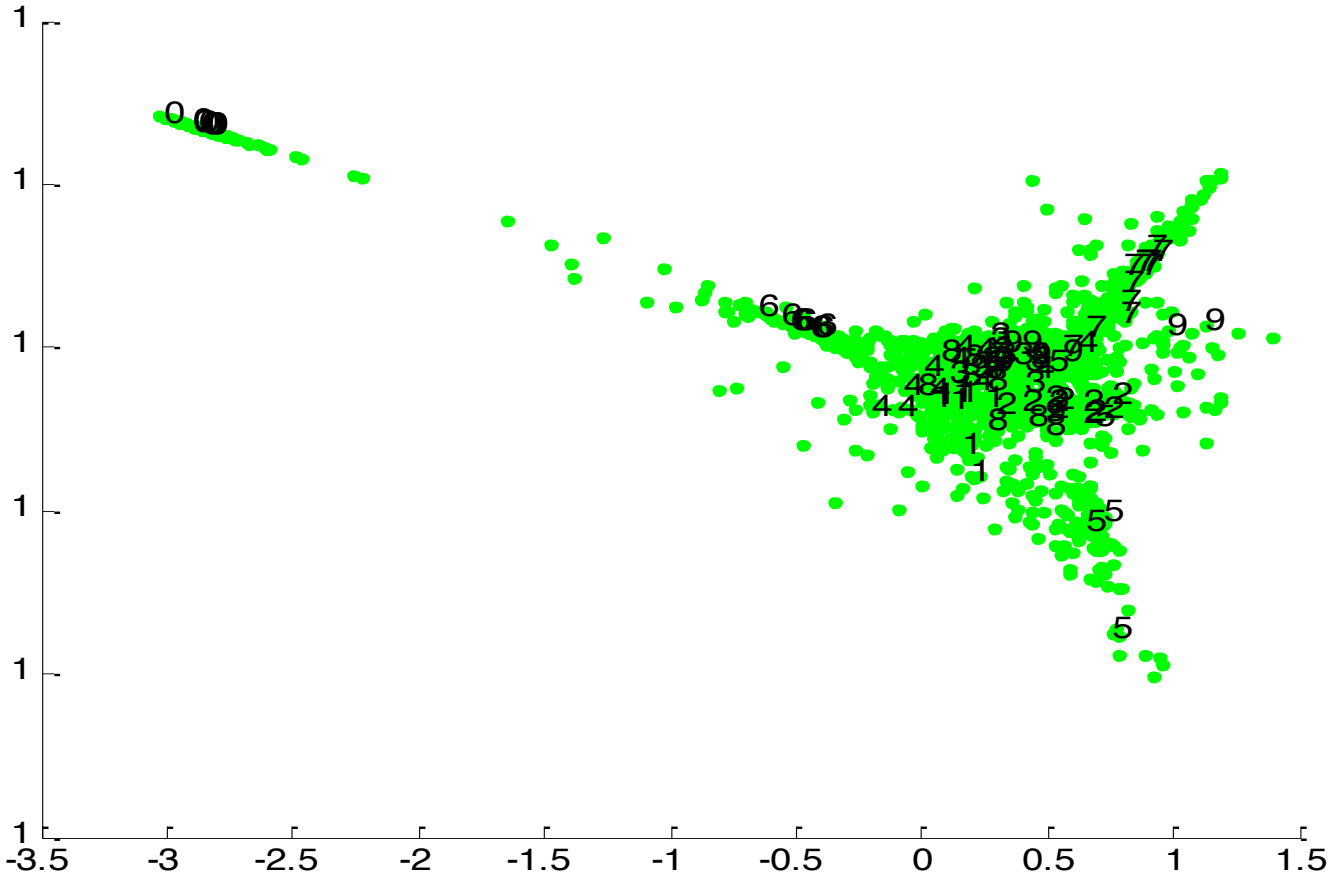
PCA



LLE

Saul et al., An Introduction to LLE

LLE on Optdigits



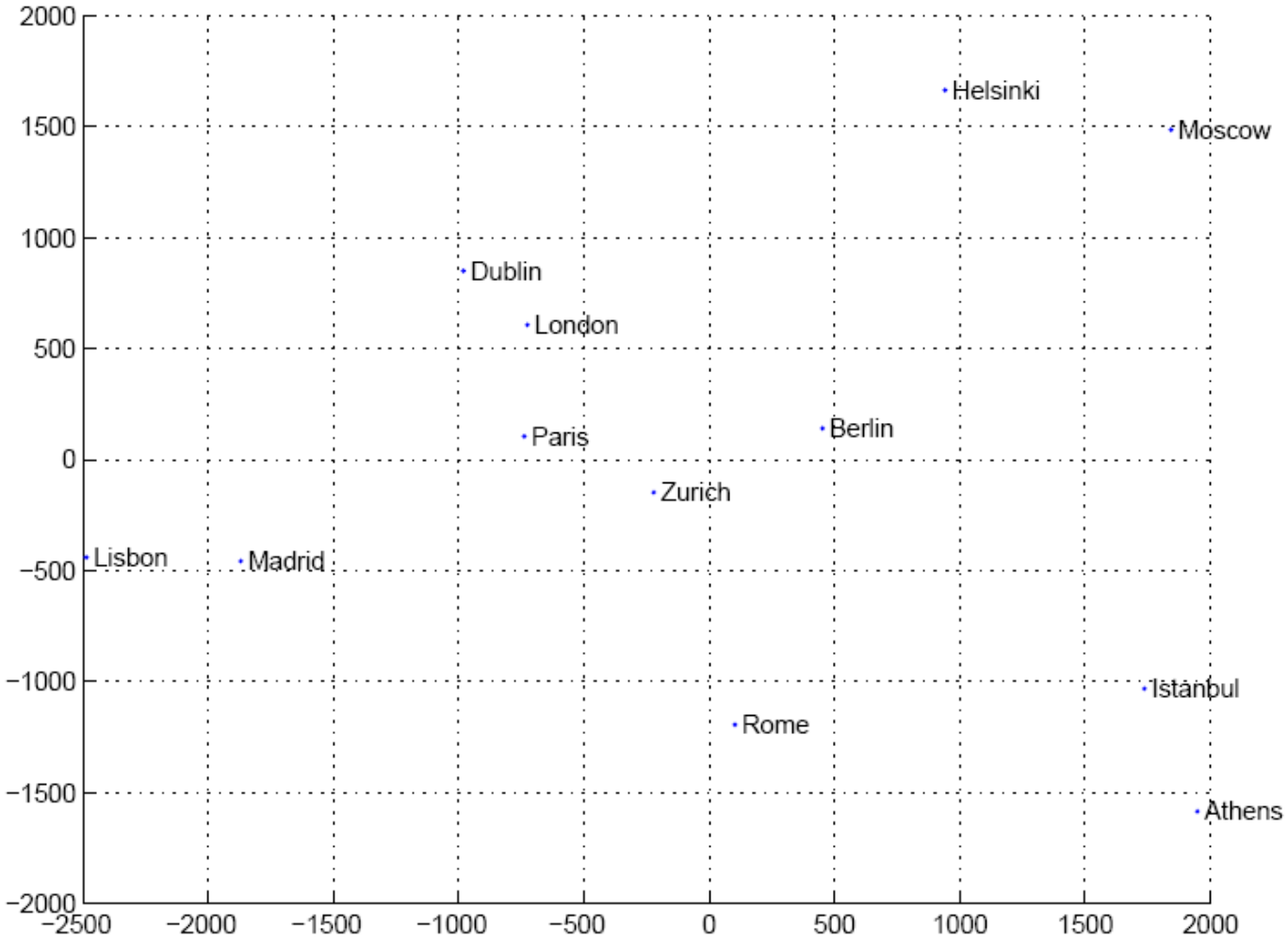
Matlab source from http://www.cs.toronto.edu/~roweis/lle/code.html

Construct Map By Distances

	Vi	Br	An	Ge	Ba	Zu	St	Ma	Mu	Ilu	Be	Br	Ha	Fr	Ha	Du	Es	Co	Bi	Do	Le	Co	Ma	Ba	Pa	St	Bo	To	Ly	Mo	Ma	Tu	Mi	Am	Ro	Os	St	Ma	Bi	Lo	Br
Vienna																																									
Brussels	8																																							
Antwerp	7	13																																						
Geneva	8	12	10																																					
Basle	8	10	8	11																																				
Zurich	8	12	10	13	11																																			
Stuttgart	7	10	9	11	9	11																																		
Mannheim	5	9	7	9	8	9	8																																	
Munich	10	11	10	12	10	12	12	8																																
Nurnberg	6	9	8	10	8	10	10	6	11																															
Berlin	9	11	10	12	10	12	12	8	14	11																														
Bremen	9	11	10	11	10	11	11	8	13	10	13																													
Hamburg	9	13	12	12	10	12	12	8	14	11	14	15																												
Frankfurt	8	13	11	12	10	12	11	9	13	11	13	12	14																											
Hannover	7	9	9	9	8	9	9	5	10	8	10	9	11	10																										
Dusseldorf	9	14	13	12	10	12	12	9	14	11	14	15	17	15	11																									
Essen	8	9	8	10	9	10	10	7	12	9	12	12	12	11	8	12																								
Cologne	5	8	8	9	7	9	10	7	10	8	10	9	10	9	8	10	8																							
Bielefeld	5	7	7	7	6	7	7	4	8	7	8	8	9	8	8	9	6	6																						
Dortmund	5	8	8	8	6	8	8	5	9	8	9	9	10	9	8	10	7	7	9																					
Leipzig	4	7	7	8	6	8	9	6	9	7	9	8	9	8	7	9	7	9	5	6																				
Copenhagen	5	7	7	7	6	7	6	4	7	5	7	7	12	7	8	9	5	5	6	6	5																			
Madrid	8	10	8	11	10	11	9	7	10	8	10	9	10	10	9	10	8	7	7	7	6	7																		
Barcelona	7	9	8	10	9	10	8	6	9	7	9	8	9	9	9	9	7	7	7	7	6	7	10																	
Paris	8	15	13	13	11	13	11	10	12	10	12	12	14	14	10	15	10	9	8	9	8	8	11	10																
Strasbourg	6	11	9	11	9	11	9	9	10	9	10	9	10	12	7	11	8	8	6	7	7	5	9	8	12														
Bordeaux	7	10	9	11	9	11	9	7	10	8	10	9	10	10	9	10	8	8	7	8	7	7	10	10	11	9														
Toulouse	6	8	7	9	8	9	7	6	8	7	8	8	8	9	7	8	7	6	6	7	5	5	8	8	9	8	9													
Lyon	8	12	10	13	11	13	11	9	12	10	12	11	13	12	9	12	10	9	7	8	8	8	11	10	13	11	11	9												
Montpellier	6	9	8	10	8	10	9	7	9	7	9	8	9	9	8	9	7	8	6	7	7	6	9	9	10	8	10	8	10											
Marseille	7	10	9	11	9	11	9	7	10	8	10	9	10	10	9	10	8	8	7	8	7	7	10	10	11	9	11	9	11	10										
Turin	5	8	6	9	8	9	8	7	8	7	8	7	8	8	6	8	6	7	4	4	6	4	8	7	9	8	7	5	9	7	7									
Milan	9	11	9	12	11	12	10	8	12	9	11	10	11	11	9	11	9	8	7	7	7	7	11	10	12	10	10	8	12	9	10	9								
Amsterdam	8	14	13	12	10	12	11	9	12	10	12	13	16	13	10	16	10	9	8	9	8	9	10	9	15	11	10	8	12	9	10	8	11							
Rotterdam	6	8	7	8	7	8	7	6	8	7	8	7	8	8	7	9	6	6	5	5	5	5	8	7	9	8	7	5	8	6	7	7	8	9						
Oslo	5	8	8	8	7	8	7	5	8	6	8	8	12	8	8	10	6	6	6	6	6	10	7	7	9	6	7	5	9	6	7	5	8	11	5					
Stockholm	7	10	9	10	9	10	9	7	10	8	10	10	14	10	9	12	8	7	7	7	6	10	9	8	11	8	8	6	11	7	8	7	10	13	7	12				
Malmo	4	6	6	6	5	6	5	3	6	5	6	6	12	7	7	8	4	4	6	6	4	11	6	6	7	5	6	5	7	5	6	3	6	9	4	10	10			
Birmingham	6	8	7	8	7	8	7	6	8	6	8	7	8	8	7	9	7	5	6	6	4	6	8	7	9	7	7	5	8	6	7	5	8	9	7	6	8	5		
London	7	14	12	12	10	12	10	9	11	9	11	12	16	13	9	15	10	8	7	8	7	9	10	9	16	11	10	8	13	9	10	8	11	16	8	11	13	9	9	
Bristol	7	10	9	8	7	8	7	6	9	6	9	9	11	10	8	12	8	5	7	7	4	7	8	7	11	7	7	5	8	6	7	5	8	11	7	7	9	6	9	11

Map from CIA – The World Factbook: http://www.cia.gov/

Map of Europe by MDS



Multidimensional Scaling

- Given pairwise distances between N points,

  d_{ij}, $i,j = 1,\ldots,N$

  place on a low-dim map s.t. distances are preserved.

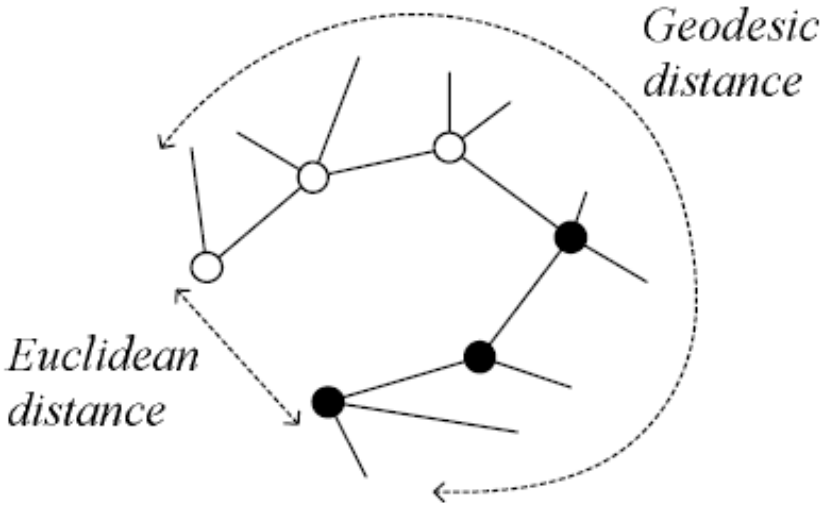
- $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x} \mid \theta)$ Find θ that min Sammon stress

$$E(\theta \mid \mathcal{X}) = \sum_{r,s} \frac{\left(\left\|\mathbf{z}^r - \mathbf{z}^s\right\| - \left\|\mathbf{x}^r - \mathbf{x}^s\right\|\right)^2}{\left\|\mathbf{x}^r - \mathbf{x}^s\right\|^2}$$

$$= \sum_{r,s} \frac{\left(\left\|\mathbf{g}\left(\mathbf{x}^r \mid \theta\right) - \mathbf{g}\left(\mathbf{x}^s \mid \theta\right)\right\| - \left\|\mathbf{x}^r - \mathbf{x}^s\right\|\right)^2}{\left\|\mathbf{x}^r - \mathbf{x}^s\right\|^2}$$

Isomap

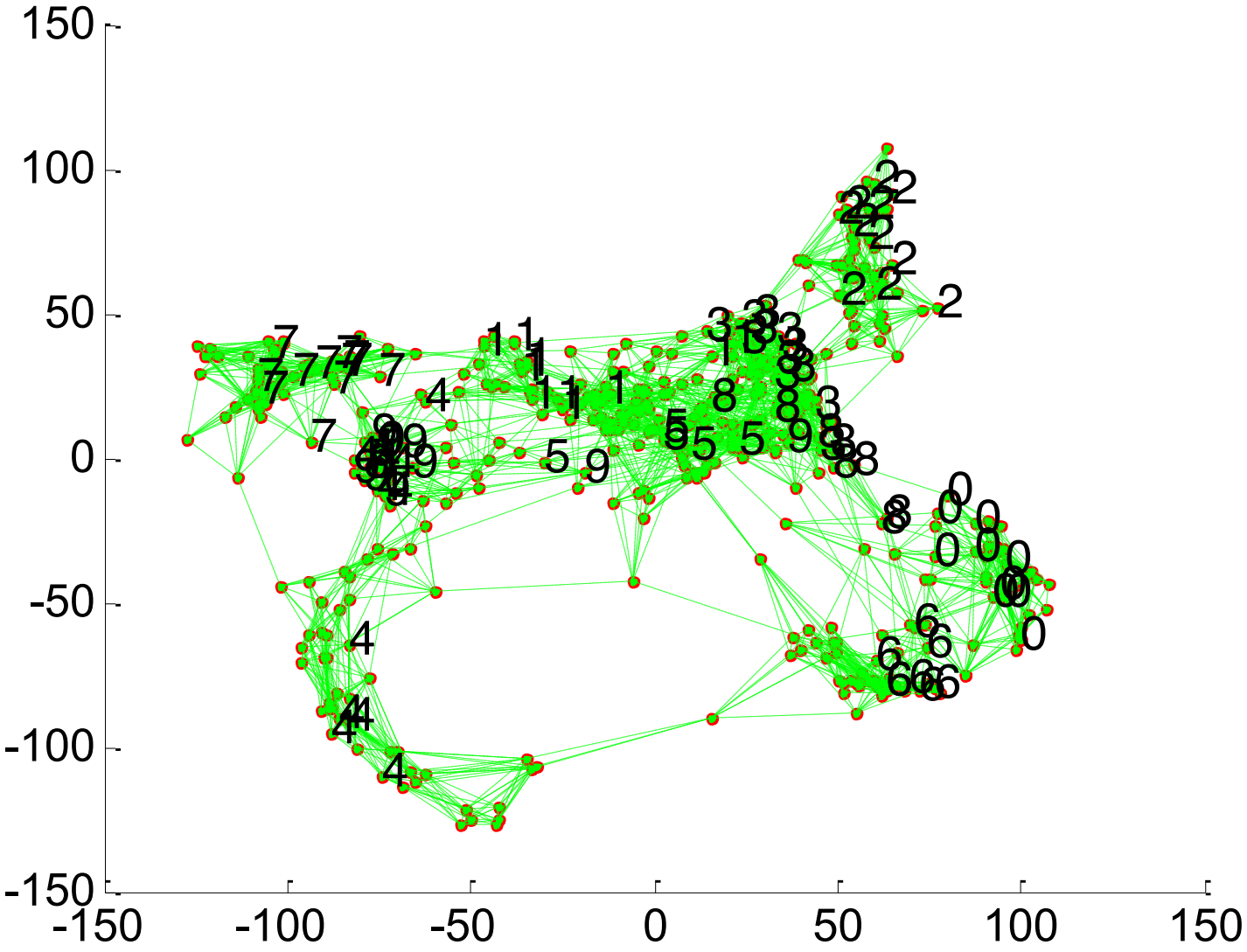
- Geodesic distance is the distance along the manifold that the data lies in, as opposed to the Euclidean distance in the input space



Isomap

- Instances r and s are connected in the graph if $||\boldsymbol{x}^r-\boldsymbol{x}^s||<\varepsilon$ or if $\boldsymbol{x}^s$ is one of the k neighbors of $\boldsymbol{x}^r$
  The edge length is $||\boldsymbol{x}^r-\boldsymbol{x}^s||$
- For two nodes r and s not connected, the distance is equal to the shortest path between them (force faraway data points to go through their neighbors - unfolding).
- Once the NxN distance matrix is thus formed, use MDS to find a lower-dimensional mapping
- No general mapping function; rerun for each new point

Optdigits after Isomap (with neighborhood graph).



Matlab source from http://web.mit.edu/cocosci/isomap/isomap.html

Summary

- **Linear projection:**
  - **PCA:** Project ***x*** to ***z*** to maximize the variance
  - **LDA:** Projection maximize within-class scatter and minimize between class scatter
- **Non-linear embedding**
  - **Isomap:** Use geodesic distance along the manifold instead of Euclidean distance
  - **LLE:** Linear patch assumption (linear combinations of neighbors)

Subset Selection vs Extraction

- Feature selection (subset selection): Choosing $k<d$ important features, ignoring the remaining $d - k$

  Subset selection algorithms

- Feature extraction: Project the

  original x_i , i =1,...,d dimensions to

  new $k<d$ dimensions, z_j , j =1,...,k

Principal components analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

Subset Selection

- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially $\emptyset$.
  - At each iteration, find the best new feature

    $j = \text{argmin}_i E (F \cup x_i)$
  - Add x_j to F if $E (F \cup x_j) < E (F)$

- Backward search: Start with all features and remove one at a time, if possible.
- Hill-climbing $O(d^2)$ algorithm
- Floating search (Add k, remove l)

Forward Selection

1. Add the highest ranked feature
2. Check classification performance

Forward Selection

1. Add the highest ranked feature
2. Check classification performance
3. Add the next highest ranked feature

Classification Accuracy: 75%→95%

Vegetarian	No
Plays video games	Yes
Family history	No
Athletic	No
Smoker	Yes
Sex	Male
Lung capacity	5.8L
Hair color	Red
Car	Audi
…	
Weight	185 lbs

Rank features →

Family history	No
Smoker	Yes
Weight	185 lbs
Sex	Male
Plays video games	No
Athletic	No
Lung capacity	5.8L
Vegetarian	No
Hair color	Red
…	
Car	Audi

Forward Selection →

Family history	No
Smoker	Yes

Forward Selection

1. Add the highest ranked feature
2. Check classification performance
3. Add the next highest ranked feature

Classification Accuracy: 95%→80%

Vegetarian	No
Plays video games	Yes
Family history	No
Athletic	No
Smoker	Yes
Sex	Male
Lung capacity	5.8L
Hair color	Red
Car	Audi
…	
Weight	185 lbs

Rank features →

Family history	No
Smoker	Yes
Weight	185 lbs
Sex	Male
Plays video games	No
Athletic	No
Lung capacity	5.8L
Vegetarian	No
Hair color	Red
…	
Car	Audi

Forward Selection →

Family history	No
Smoker	Yes
Weight	185 lbs

Backward Elimination

1. Remove the lowest ranked feature
2. Check classification performance

Classification Accuracy: 60%→75%

Vegetarian	No
Plays video games	Yes
Family history	No
Athletic	No
Smoker	Yes
Sex	Male
Lung capacity	5.8L
Hair color	Red
Car	Audi
…	
Weight	185 lbs

Rank features →

Family history	No
Smoker	Yes
Weight	185 lbs
Sex	Male
Plays video games	No
Athletic	No
Lung capacity	5.8L
Vegetarian	No
Hair color	Red
…	
Car	Audi

Forward Selection →

Family history	No
Smoker	Yes
Weight	185 lbs
Sex	Male
Plays video games	No
Athletic	No
Lung capacity	5.8L
Vegetarian	No
Hair color	Red
…	
Car	Audi

Backward Elimination

1. Remove the lowest ranked feature
2. Check classification performance
3. Remove the next lowest ranked feature until performance worse

Classification Accuracy: 95%

Vegetarian	No
Plays video games	Yes
Family history	No
Athletic	No
Smoker	Yes
Sex	Male
Lung capacity	5.8L
Hair color	Red
Car	Audi
…	
Weight	185 lbs

Rank features →

Family history	No
Smoker	Yes
Weight	185 lbs
Sex	Male
Plays video games	No
Athletic	No
Lung capacity	5.8L
Vegetarian	No
Hair color	Red
…	
Car	Audi

Forward Selection →

Family history	No
Smoker	Yes

Feature Selection

- NP-hard to search through all the combinations
  - Need heuristic solutions
- The assumption is based on the maximum classification performance.
  - There might be more than one subset of features that can give the optimal classification performance.
- Easy to ignore the relation among the features
  - Combinations of several non-informative features might be meaningful