Dimensionality Reduction

Aarti Singh

Machine Learning 10-701/15-781
Apr 5, 2010

Slides Courtesy: Tom Mitchell, Eric Xing, Lawrence Saul

High-Dimensional data

- High-Dimensions = Lot of Features

Document classification

Features per document =

thousands of words/unigrams

millions of bigrams, contextual information

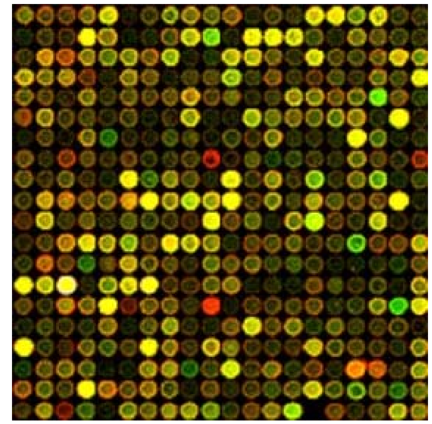
Surveys - Netflix

480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

High-Dimensional data

- High-Dimensions = Lot of Features



Discovering gene networks

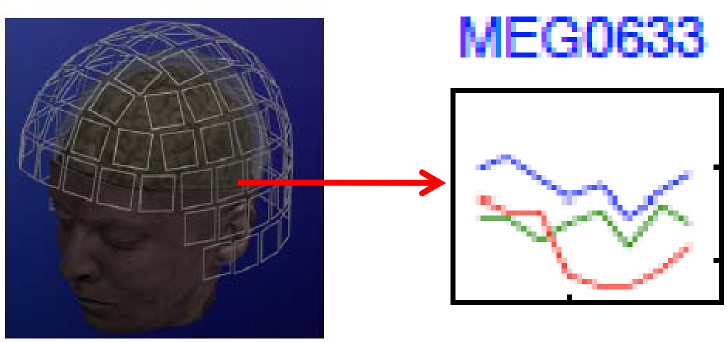
10,000 genes x 1000 drugs

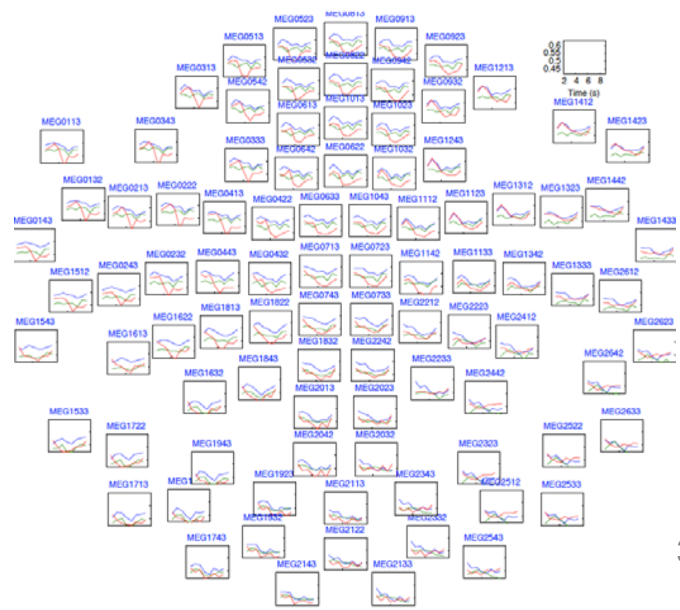
x several species

MEG Brain Imaging

120 locations x 500 time points

x 20 objects

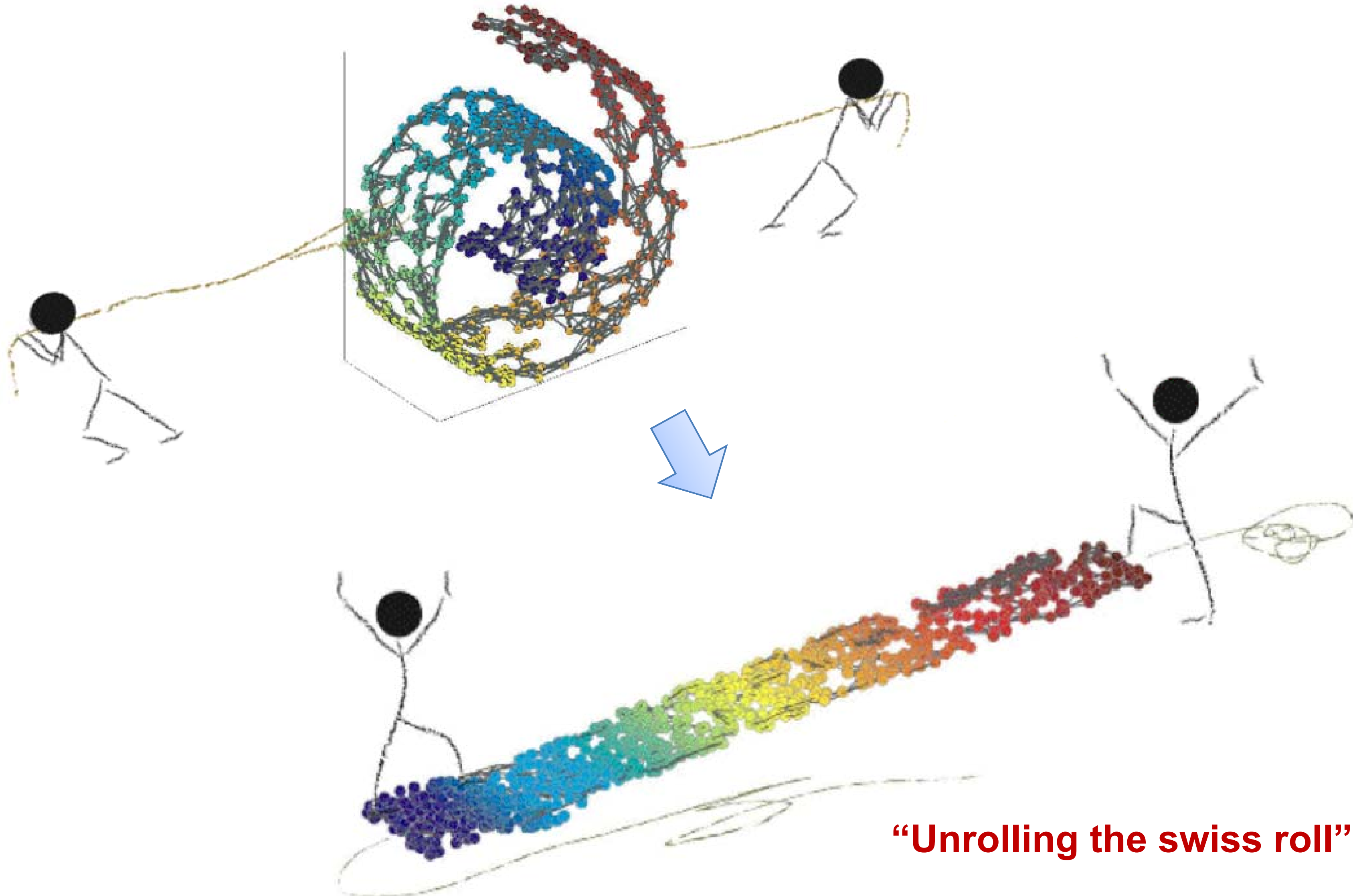




Curse of Dimensionality

- Why are more features bad?
  - Redundant features (not all words are useful to classify a document) more noise added than signal
  - Hard to interpret and visualize
  - Hard to store and process data (computationally challenging)
  - Complexity of decision rule tends to grow with # features. Hard to learn complex rules as VC dimension increases (statistically challenging)

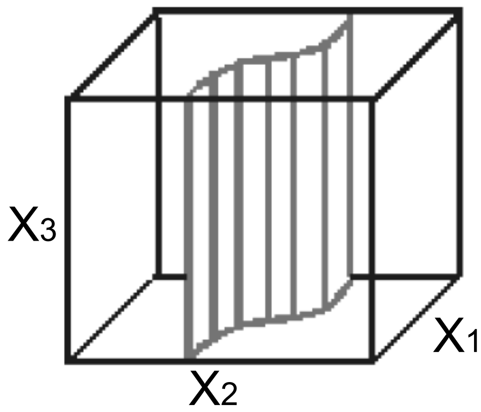
Dimensionality Reduction



"Unrolling the swiss roll"

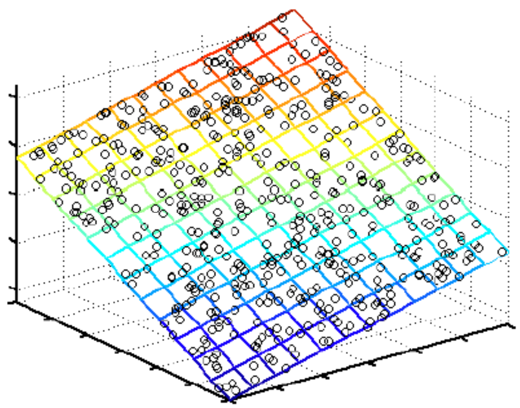
Dimensionality Reduction

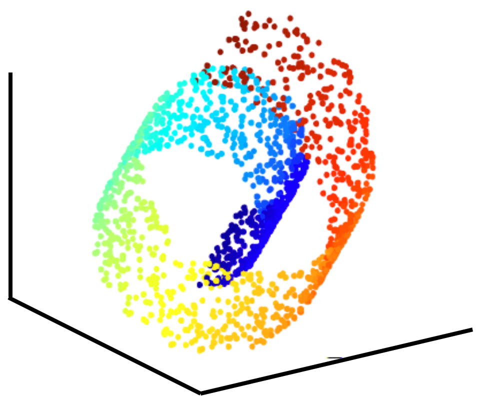
- Feature Selection – Only a few features are relevant to the learning task



X_3 - Irrelevant

- Latent features – Some linear/nonlinear combination of features provides a more efficient representation than observed features





Feature Selection

- Approach 1: **Score each feature and extract a subset**

Common scoring methods:

- Training or cross-validated accuracy of single-feature classifiers $f_i: X_i \rightarrow Y$
- Estimated mutual information between X_i and Y :

$$\hat{I}(X_i, Y) = \sum_k \sum_y \hat{P}(X_i = k, Y = y) \log \frac{\hat{P}(X_i = k, Y = y)}{\hat{P}(X_i = k)\hat{P}(Y = y)}$$

- χ^2 statistic to measure independence between X_i and Y
- Domain specific criteria
  - Text: Score "stop" words ("the", "of", …) as zero
  - fMRI: Score voxel by T-test for activation versus rest condition
  - …

Feature Selection

- Approach 1: **Score each feature and extract a subset**

Common subset selection methods:

- One step: Choose d highest scoring features
- Iterative:
  - Choose single highest scoring feature X_k
  - Rescore all features, conditioned on the set of already-selected features
    - E.g., $Score(X_i \mid X_k) = I(X_i, Y \mid X_k)$
    - E.g, $Score(X_i \mid X_k)$ = Accuracy(predicting Y from X_i and X_k)
  - Repeat, calculating new scores on each iteration, conditioning on set of selected features

Feature Selection: Text Classification

Approximately 10^5 words in English

[Rogati&Yang, 2002]

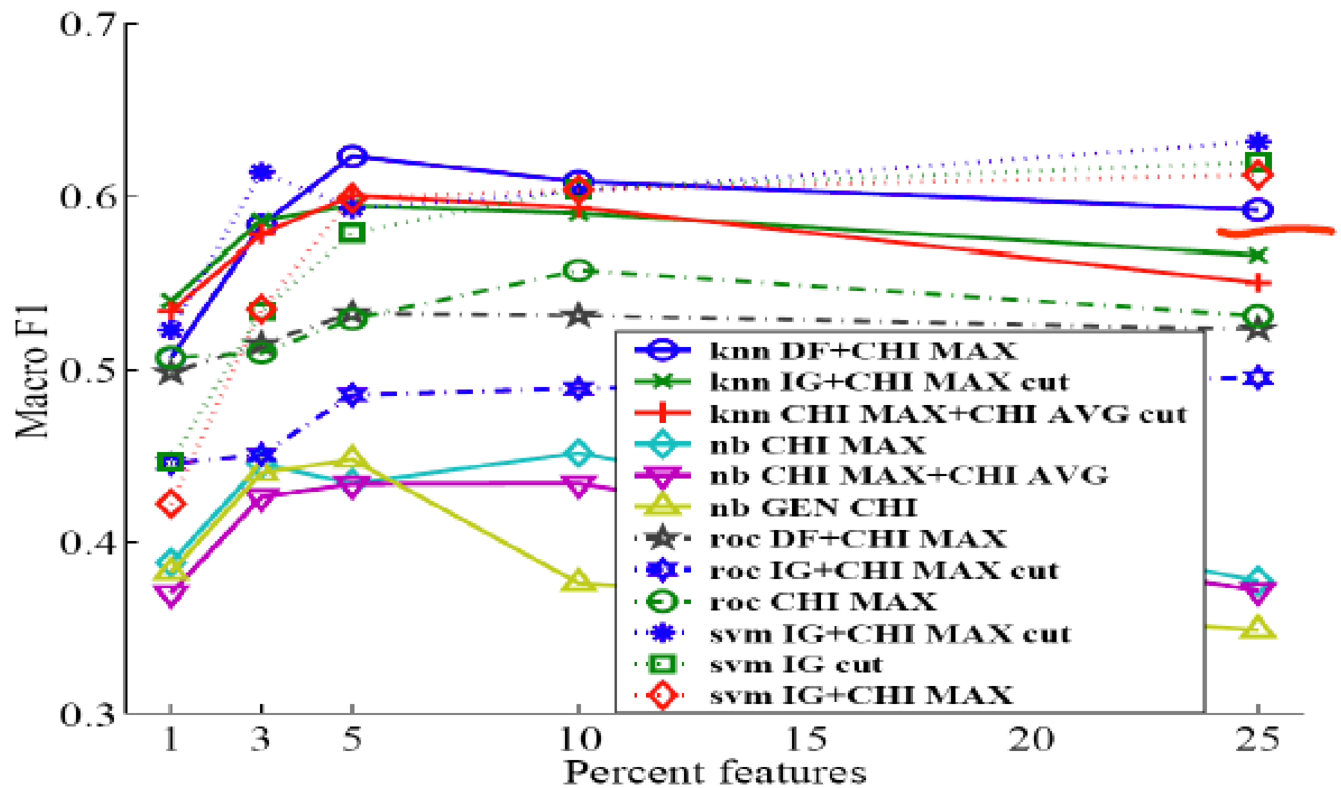


Figure 2: Top 3 feature selection methods for Reuters-21578 (Macro F1)

IG=information gain, chi= χ^2 , DF=doc frequency,

Impact of Feature Selection on Classification of fMRI Data

[Pereira et al., 2005]

Accuracy classifying category of word read by subject

#voxels	mean	subjects 233B	329B	332B	424B	474B	496B	77B	86B
50	0.735	0.783	0.817	0.55	0.783	0.75	0.8	0.65	0.75
100	0.742	0.767	0.8	0.533	0.817	0.85	0.783	0.6	0.783
200	0.737	0.783	0.783	0.517	0.817	0.883	0.75	0.583	0.783
300	**0.75**	**0.8**	**0.817**	**0.567**	**0.833**	**0.883**	**0.75**	**0.583**	**0.767**
400	0.742	0.8	0.783	0.583	0.85	0.833	0.75	0.583	0.75
800	0.735	0.833	0.817	0.567	0.833	0.833	0.7	0.55	0.75
1600	0.698	0.8	0.817	0.45	0.783	0.833	0.633	0.5	0.75
all (~2500)	0.638	0.767	0.767	0.25	0.75	0.833	0.567	0.433	0.733

Table 1: **Average accuracy across all pairs of categories, restricting the procedure to use a certain number of voxels for each subject.** The highlighted line corresponds to the best mean accuracy, obtained using 300 voxels.

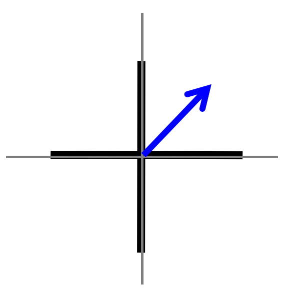
Each feature X_i is a voxel, scored by error in regression to predict X_i from Y

Feature Selection

- Approach 2: **Regularization (MAP)**
  Integrate feature selection into learning objective by penalizing number of features with non-zero weights

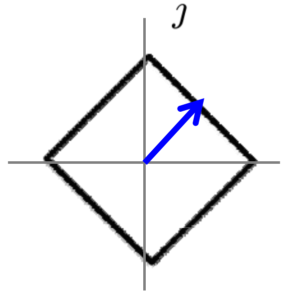
$$\widehat{W} = \arg\max_{W} \underbrace{\sum_{i=1}^{n} \log P(Y_i | X_i; W)}_{\text{log likelihood}} + \underbrace{\lambda \|W\|}_{\text{penalty}}$$

$$\|W\|_0 = \#\{W_j > 0\}$$



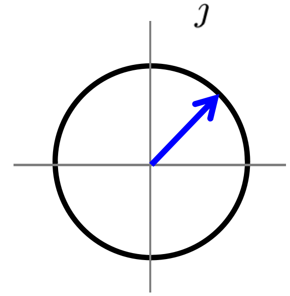
Minimizes # features chosen

$$\|W\|_1 = \sum_j |W_j|$$



Convex compromise

$$\|W\|_2 = \sum_j W_j^2$$



Small weights of features chosen

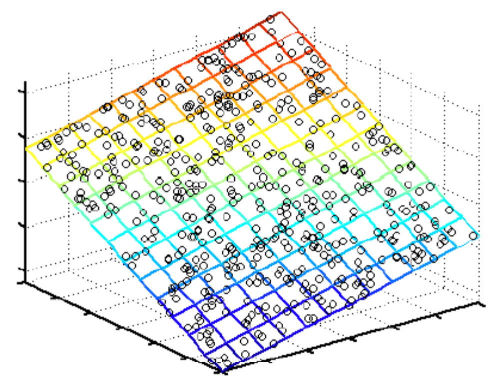
Latent Feature Extraction

Combinations of observed features provide more efficient representation, and capture underlying relations that govern the data

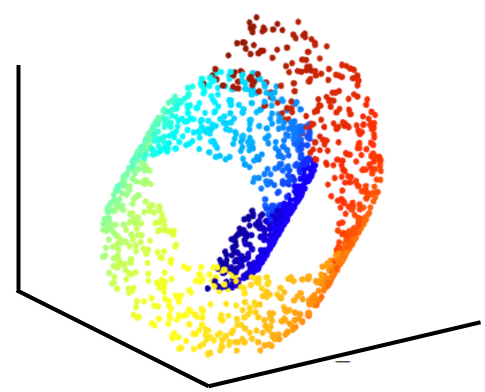
E.g. Ego, personality and intelligence are hidden attributes that characterize human behavior instead of survey questions

Topics (sports, science, news, etc.) instead of documents

Often may not have physical meaning

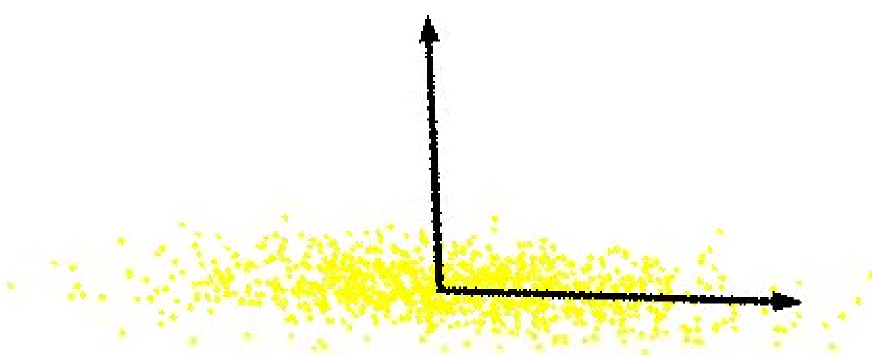


- Linear
  - Principal Component Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis (ICA)

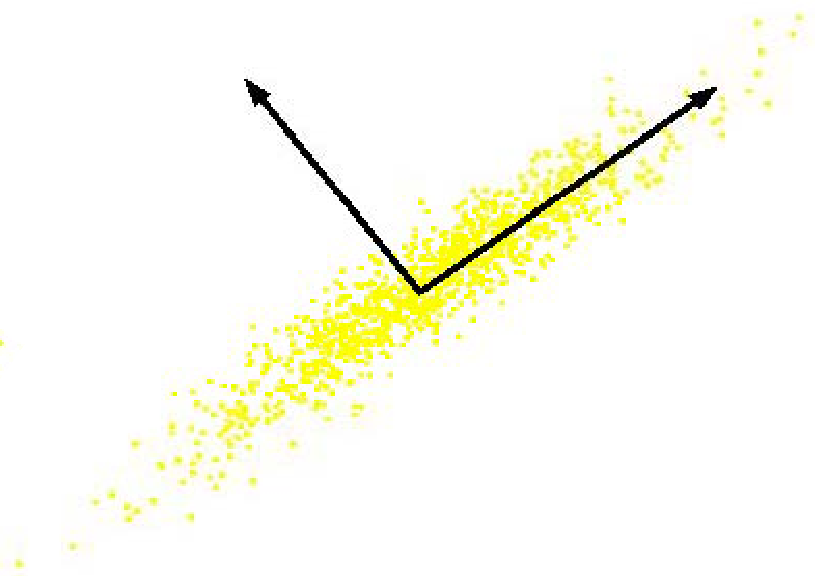


- Nonlinear
  - ISOMAP
  - Local Linear Embedding (LLE)
  - Laplacian Eigenmaps

Principal Component Analysis (PCA)



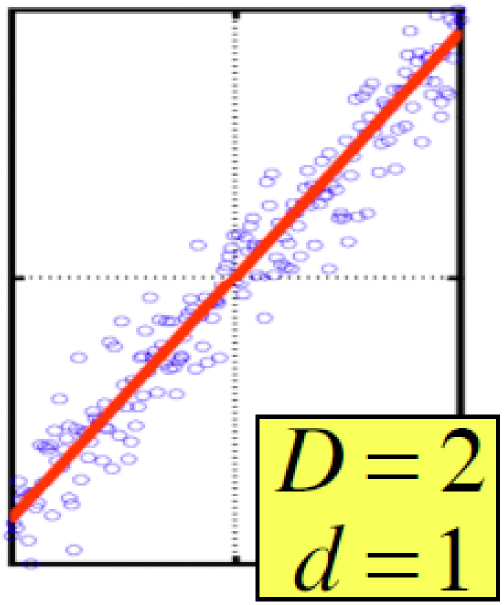
Only one relevant feature

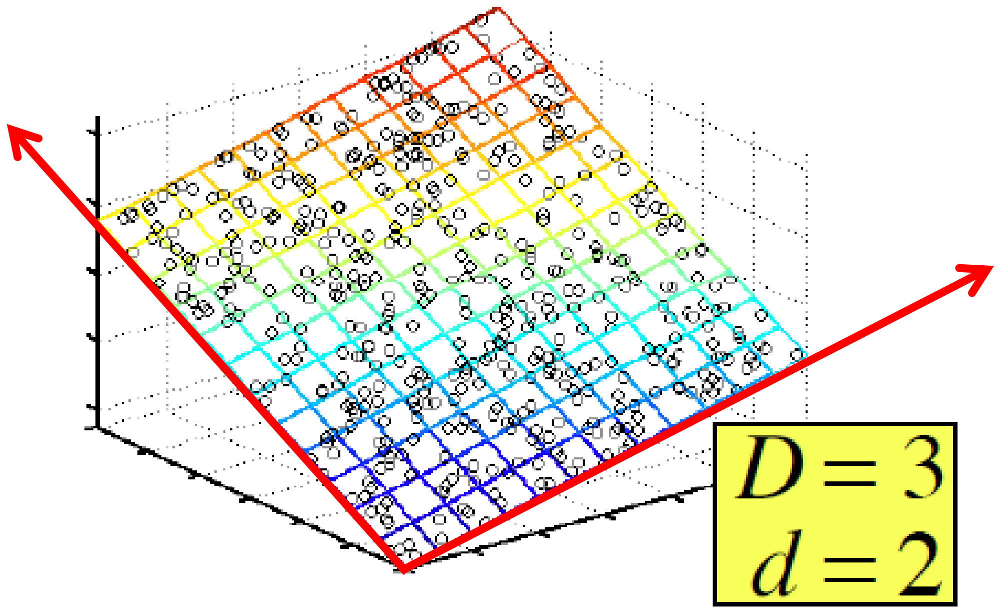


Both features become relevant

Can we transform the features so that we only need to preserve one latent feature? Find linear projection so that projected data is uncorrelated.

Principal Component Analysis (PCA)



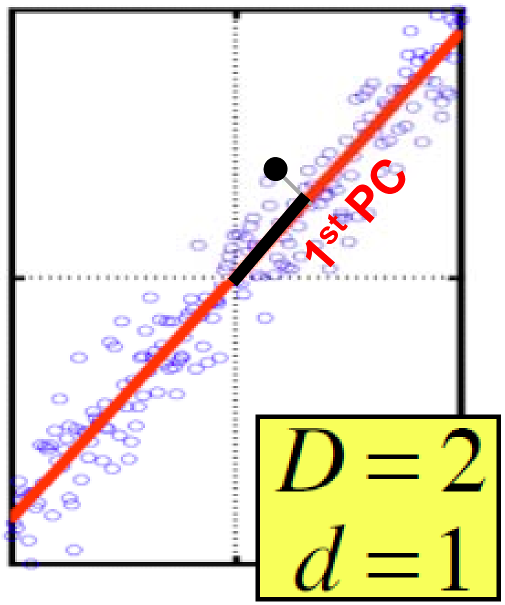


Assumption: Data lies on or near a low d-dimensional linear subspace.

Axes of this subspace are an effective representation of the data

Identifying the axes is known as Principal Components Analysis, and can be obtained by Eigen or Singular value decomposition

Principal Component Analysis (PCA)



Principal Components (PC) are orthogonal directions that capture most of the variance in the data

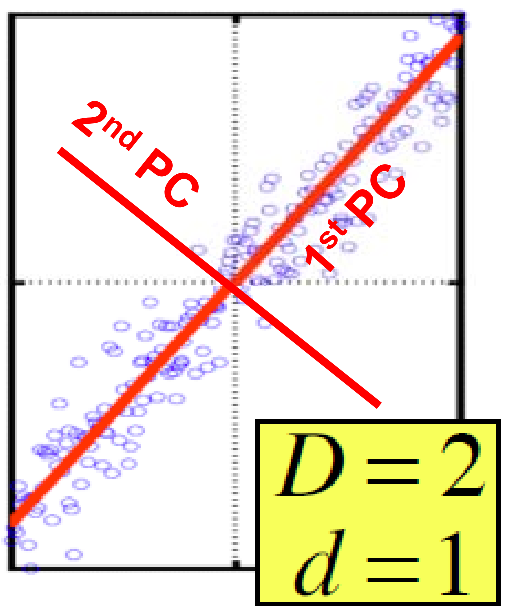
1st PC – direction of greatest variability in data

Projection of data points along 1st PC discriminate the data most along any one direction

Take a data point x_i (D-dimensional vector)

Projection of x_i onto the 1st PC v is $v^T x_i$

Principal Component Analysis (PCA)



Principal Components (PC) are orthogonal directions that capture most of the variance in the data

1st PC – direction of greatest variability in data

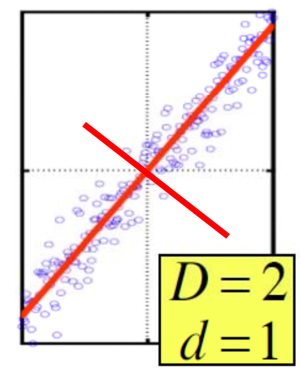
2nd PC – Next orthogonal (uncorrelated) direction of greatest variability

(remove all variability in first direction, then find next direction of greatest variability)

And so on …

Principal Component Analysis (PCA)

Let $v_1, v_2, \ldots, v_d$ denote the principal components



Orthogonal and unit norm $\quad v_i^T v_j = 0 \quad i \neq j$

$v_i^T v_i = 1$

Find vector that maximizes sample variance of projection

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

Assume data are centered
Data points $X = [\, x_1\ x_2\ \ldots\ x_n]$

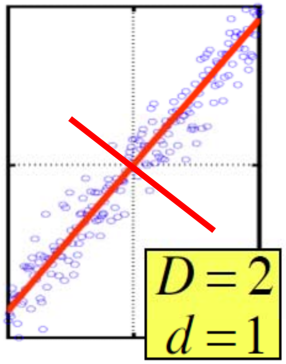
$$\max_{\mathbf{v}} \ \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} \quad \text{s.t.} \quad \mathbf{v}^T\mathbf{v} = 1$$

Lagrangian: $\max_{\mathbf{v}} \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} - \lambda\mathbf{v}^T\mathbf{v}$

Wrap constraints into the objective function

$$\partial/\partial\mathbf{v} = 0 \qquad (\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0 \qquad \Rightarrow \boxed{(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}}$$

Principal Component Analysis (PCA)



$$(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}$$

Therefore, v is the eigenvector of XX^T with eigenvalue λ

Sample variance of projection = $\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}^T\mathbf{v} = \lambda$

Thus, the eigenvalue λ denotes the amount of variability captured along that dimension.

Eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > \ldots$

The 1st Principal component v_1 is the eigenvector of the sample covariance matrix XX^T associated with the largest eigenvalue λ_1

The 2nd Principal component v_2 is the eigenvector of the sample covariance matrix XX^T associated with the second largest eigenvalue λ_2

And so on …

Computing the PCs

Eigenvectors are solutions of the following equation:

$$(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v} \qquad\qquad (\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0$$

Non-zero solution v ≠ 0 possible only if

$$\det(\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I}) = 0$$

Characteristic Equation

This is a D^{th} order equation in λ, can have at most D distinct solutions (roots of the characteristic equation)

Once eigenvalues are computed, solve for eigenvectors (Principal Components) using

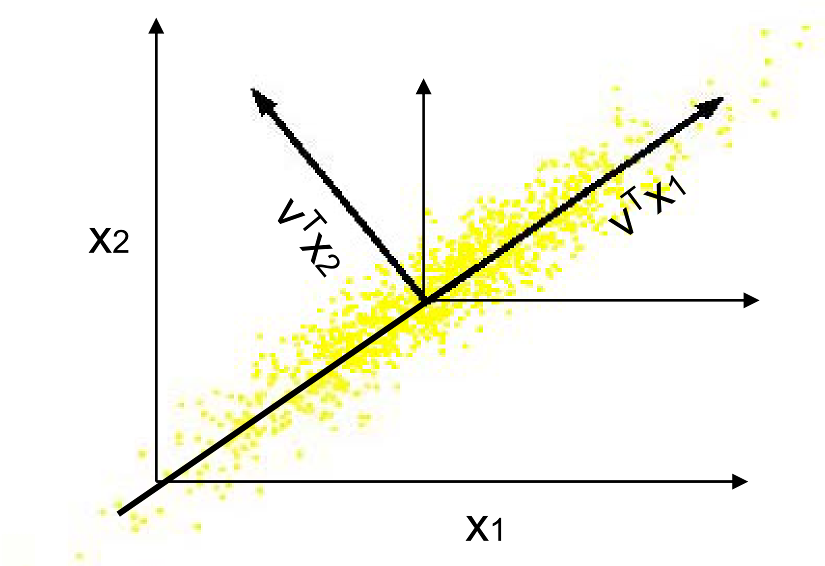
$$(\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0$$

For symmetric matrices, eigenvectors for distinct eigenvalues are orthogonal.

Principal Component Analysis (PCA)

So, the new axes are the eigenvectors of the matrix of sample correlations XX^T of the data, which capture the similarities of the original features based on how data samples project to the new axes.

Transformed features are uncorrelated.



- Geometrically: centering followed by rotation
  - Linear transformation

Another interpretation

Maximum Variance Subspace: PCA finds vectors v such that projections on to the vectors capture maximum variance in the data

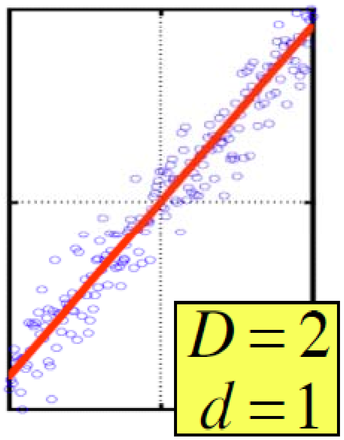
$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

Minimum Reconstruction Error: PCA finds vectors v such that projection on to the vectors yields minimum MSE reconstruction

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{v}^T\mathbf{x}_i\|^2$$

Dimensionality Reduction using PCA

The eigenvalue λ denotes the amount of variability captured along that dimension.



Zero eigenvalues indicate no variability along those directions => data lies exactly on a linear subspace

Only keep data projections onto principal components with non-zero eigenvalues, say $v_1, \ldots, v_d$ where $d = \text{rank}(XX^T)$

Original Representation
data point
$x_i = [x_i^1, x_i^2, \ldots. x_i^D]$
(D-dimensional vector)

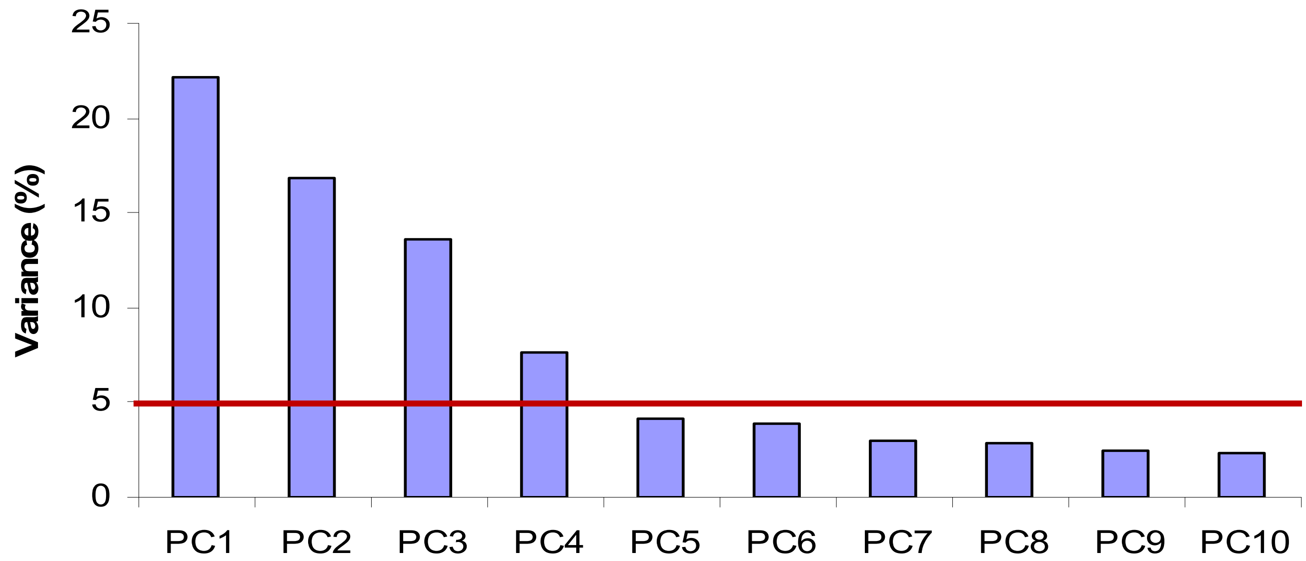
Transformed representation
projections
$[v_1^T x_i, v_2^T x_i, \ldots v_d^T x_i]$
(d-dimensional vector)

Dimensionality Reduction using PCA

Usually data lies near a linear subspace, as noise introduces small variability

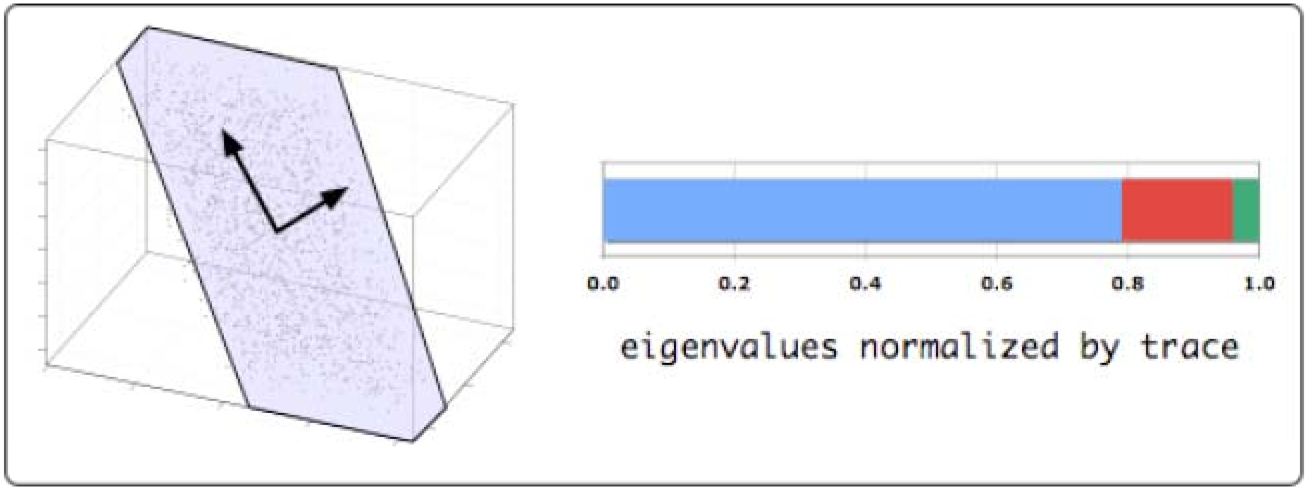
Only keep data projections onto principal components with **large** eigenvalues

Can *ignore* the components of lesser significance.



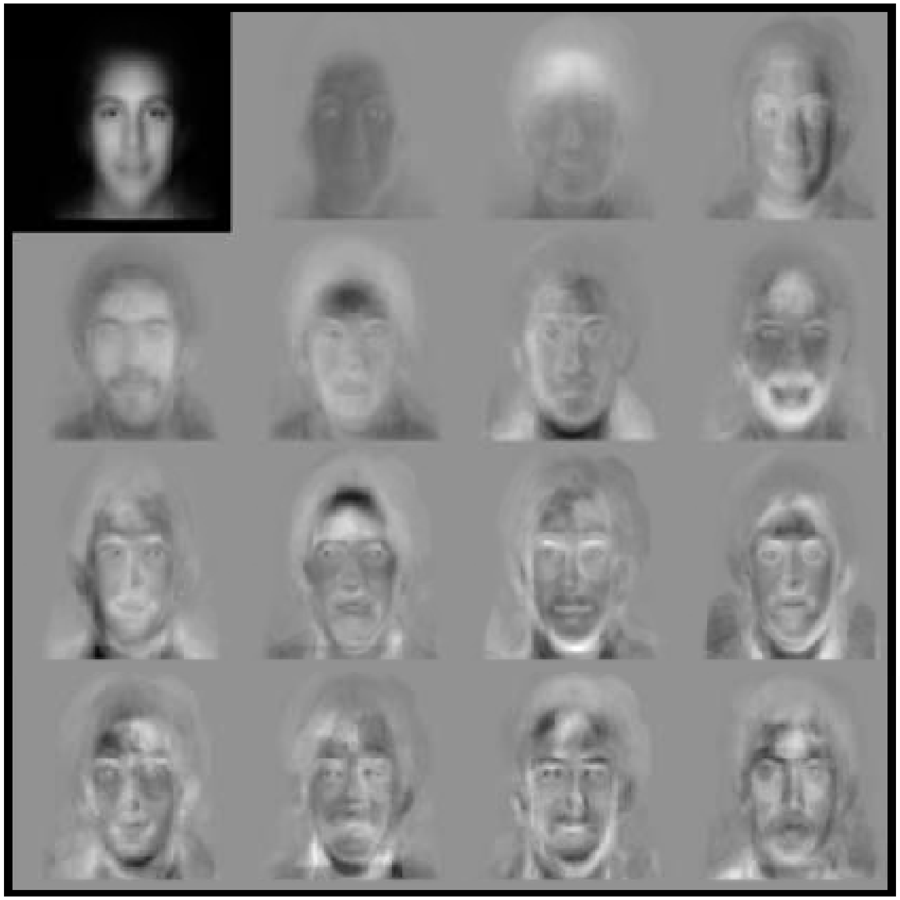
You might lose some information, but if the eigenvalues are small, you don't lose much

Example of PCA



Eigenvectors and eigenvalues of covariance matrix for $n=1600$ inputs in $d=3$ dimensions.

Example: faces



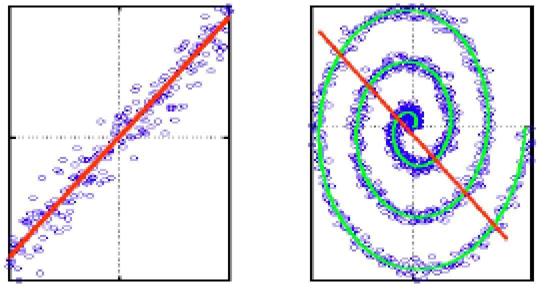
Eigenfaces from 7562 images:

top left image is linear combination of rest.

Sirovich & Kirby (1987)
Turk & Pentland (1991)

Properties of PCA

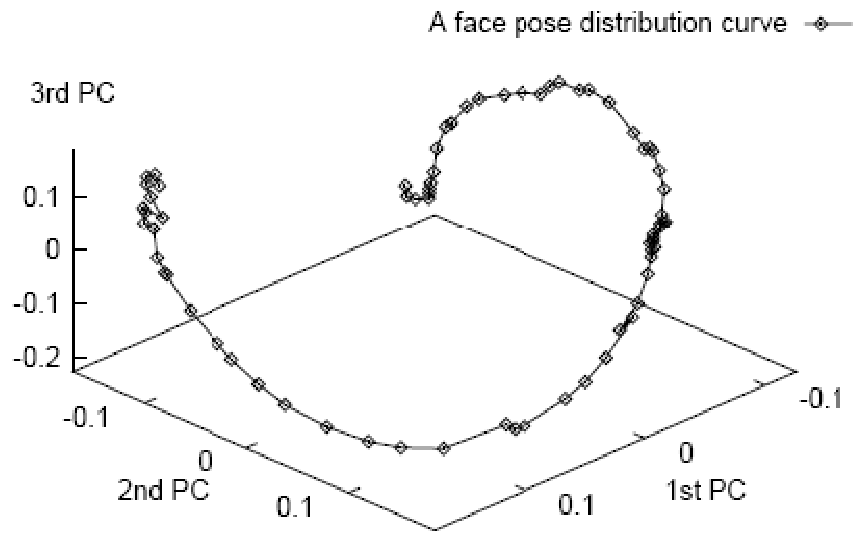
- Strengths
  - Eigenvector method
  - No tuning parameters
  - Non-iterative
  - No local optima



- Weaknesses
  - Limited to second order statistics
  - Limited to linear projections

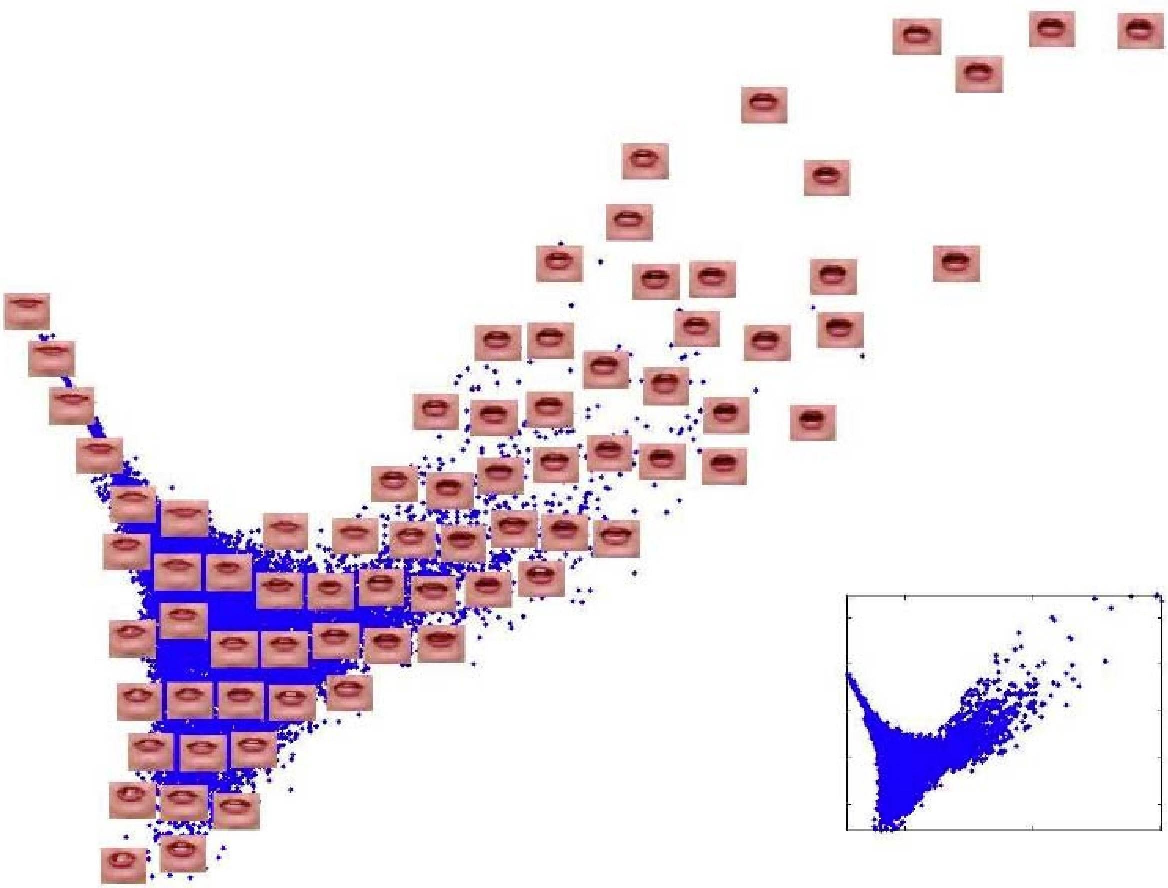
Nonlinear Methods

Data often lies on or near a nonlinear low-dimensional curve aka manifold.



Nonlinear Methods

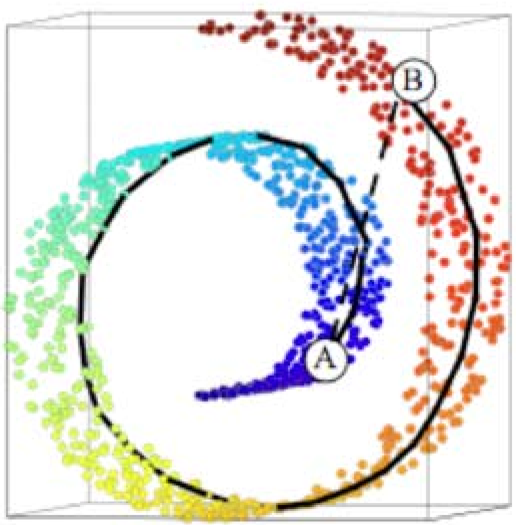
Data often lies on or near a nonlinear low-dimensional curve aka manifold.

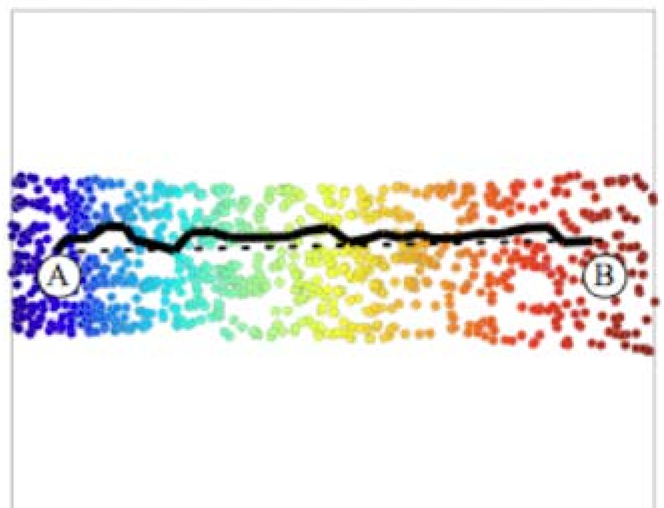


Isomap

Linear methods – Lower-dimensional linear projection that preserves Euclidean distances

ISOMAP basic idea – preserve geodesic distance as measured along the manifold



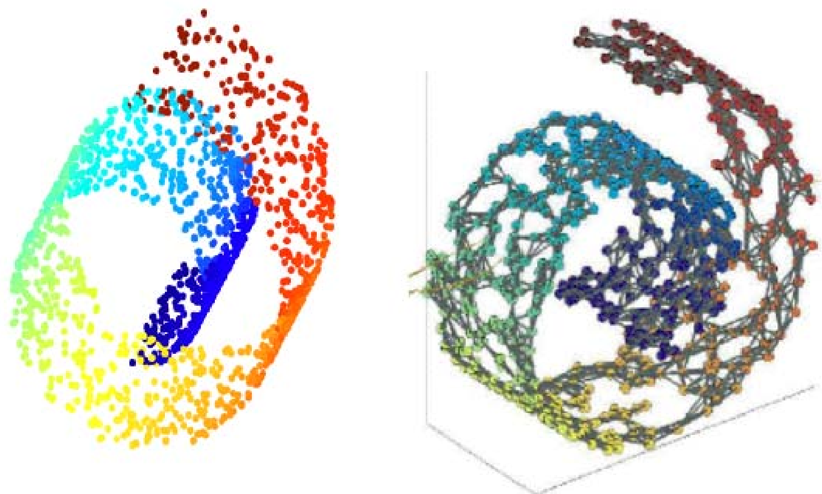


Isomap

Step 1. Build Adjacency graph

Vertices = Data points

Undirected edges connect nearest neighbors (k-NN, eps-NN)



- Graph is discretized approximation of manifold.
- k or eps chosen so that neighborhoods on graphs represent neighborhoods on the manifold (no “shortcuts” connect different arms of the swiss roll)

Isomap

Step 2. Estimate geodesic distances by graph distances

Weight edges by local distances

Compute shortest path through the graph Δ_{ij} (denser sampling => better estimates)

Step 3. Find embedding that preserves graph distances $\Delta_{ij} \sim ||y_i - y_j||$

MDS (Multi Dimensional Scaling)

Preserve dot products G_{ij}

(proxy for distances)

$$G_{ij} = \frac{1}{2}\left[\sum_k \left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \sum_{kl} \Delta_{kl}^2\right]$$

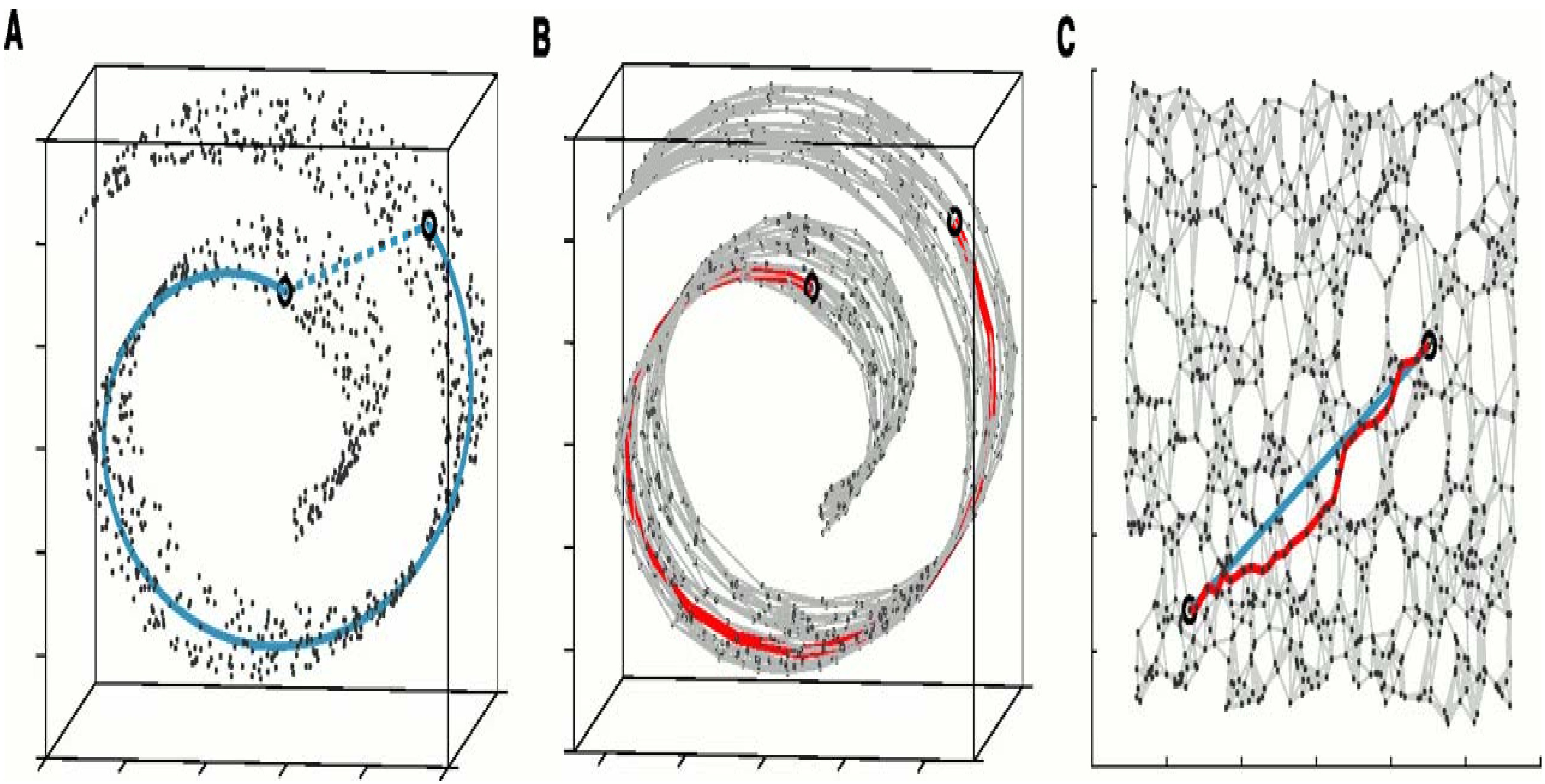
$$\arg\min_{y_1,\dots,y_n} \sum_{i,j} (G_{ij} - y_i^T y_j)$$

Solution - Top d eigenvectors of the Gram matrix G

Eigenvalues measure how each dimension contributes to dot product

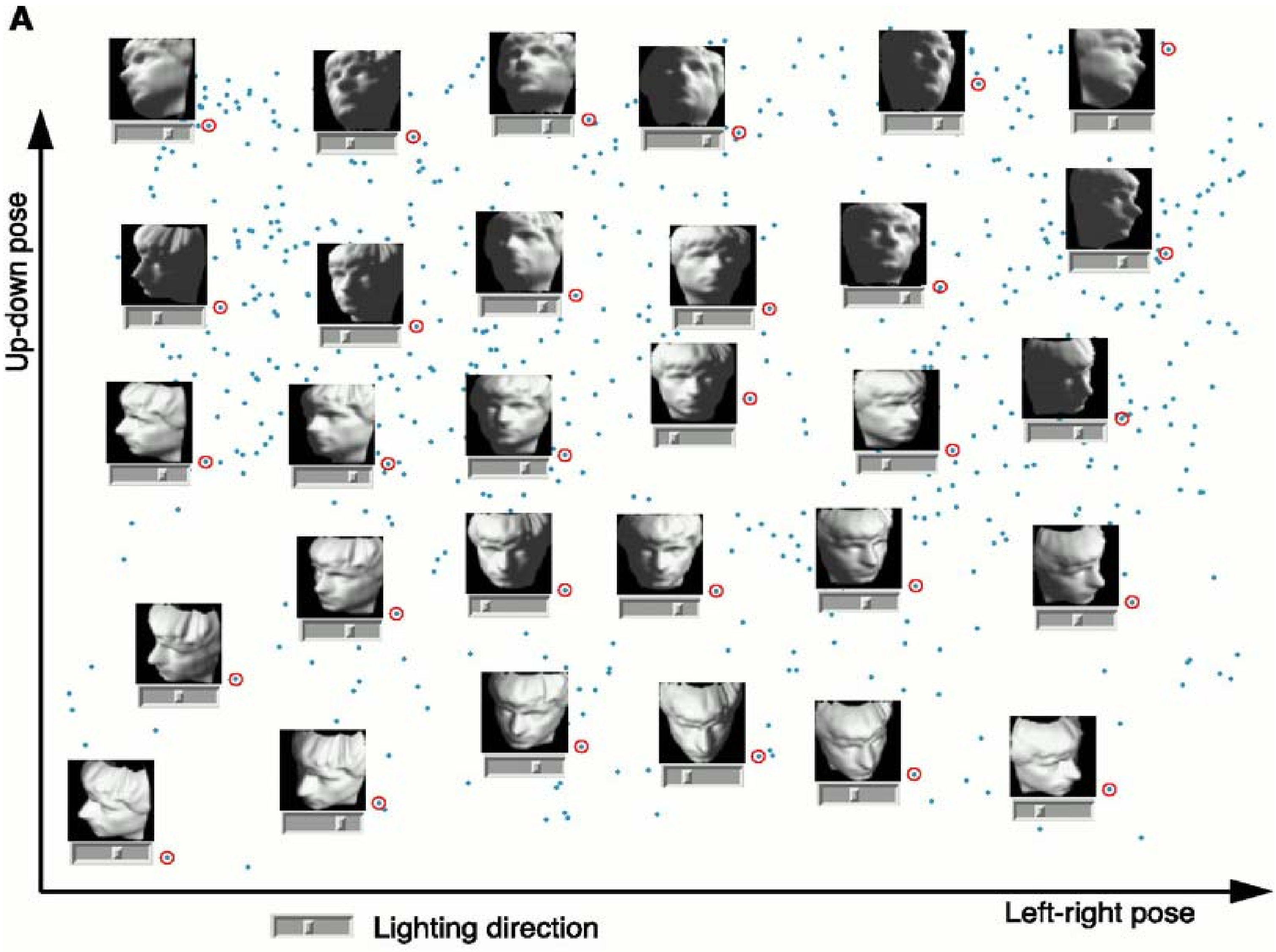
Same as PCA if distances are Euclidean

Isomap

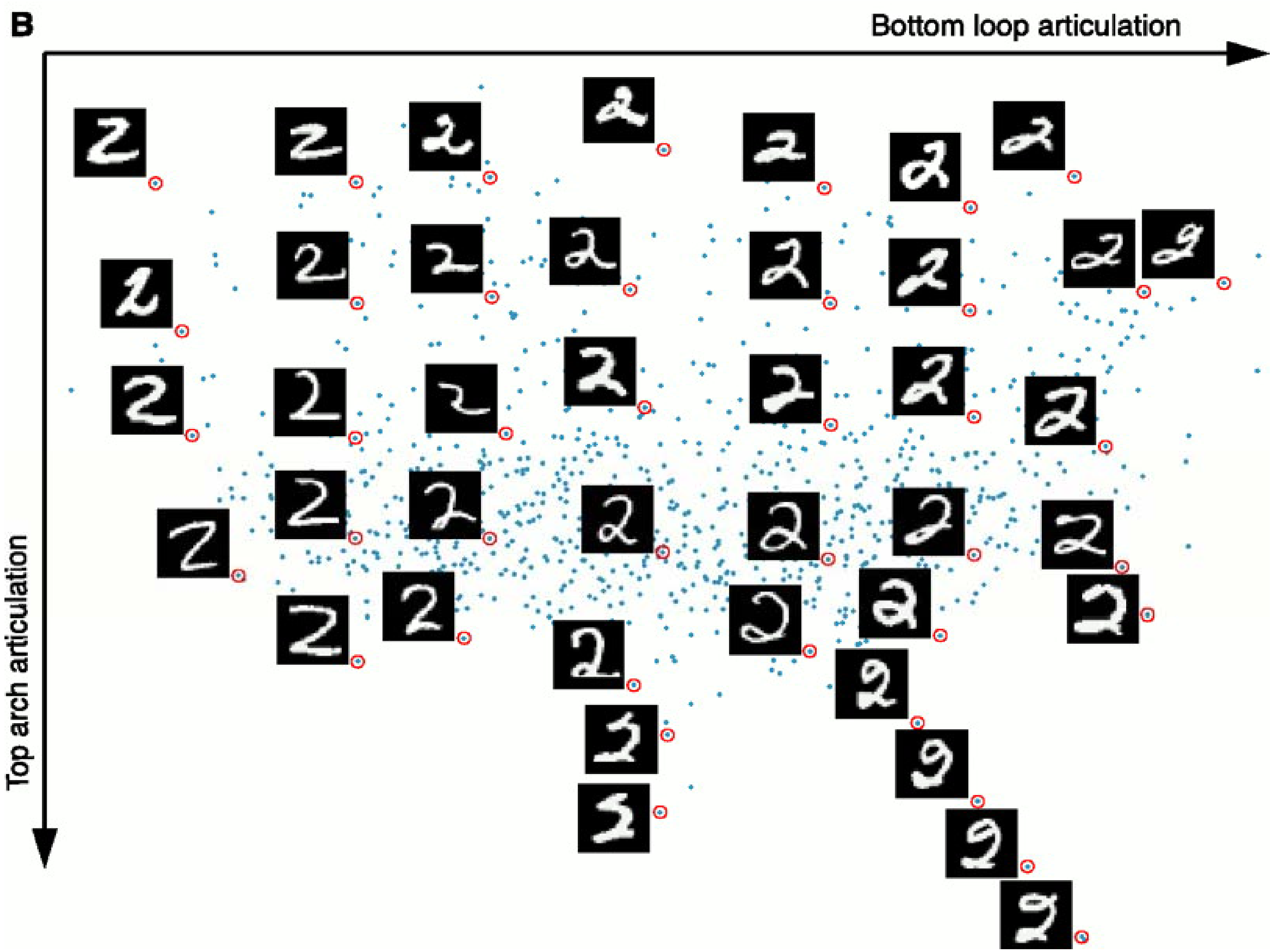


- Theoretically sound, Practically useful

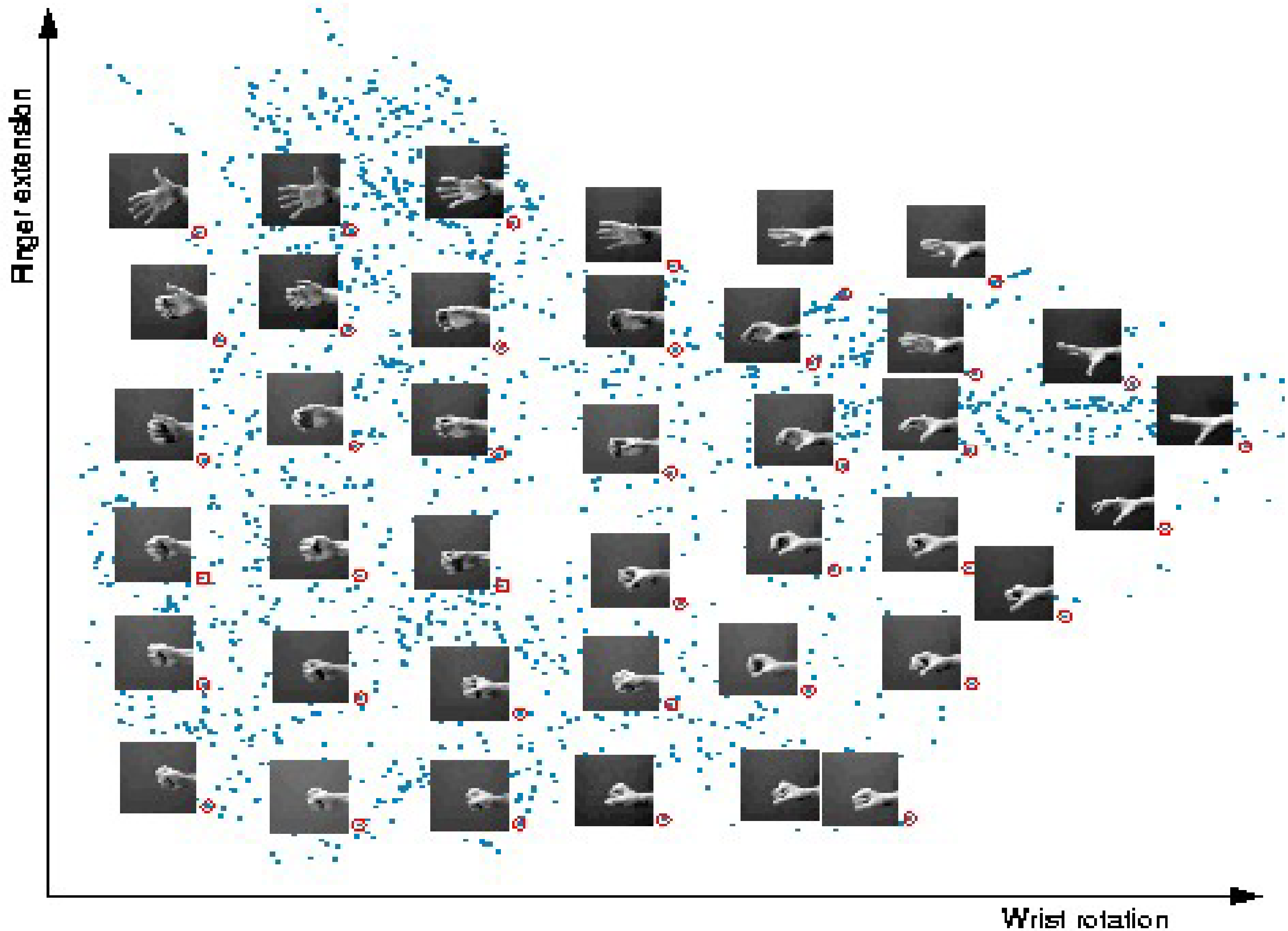
Isomap Results



Isomap Results



Isomap Results



Dimensionality Reduction Methods

- Feature Selection - Only a few features are relevant to the learning task

  Score features (mutual information, prediction accuracy, domain knowledge)
  Regularization

- Latent features – Some linear/nonlinear combination of features provides a more efficient representation than observed feature

  Linear: Low-dimensional linear subspace projection

  PCA (Principal Component Analysis),
  MDS (Multi Dimensional Scaling),
  Factor Analysis, ICA (Independent Component Analysis)

  Nonlinear: Low-dimensional nonlinear projection that preserves geodesic distances along the manifold

  ISOMAP, Kernel PCA,
  LLE (Local Linear Embedding), Laplacian Eigenmaps
  Data-driven linear subspaces (Wavelets)

Some Homework for next time …

- Think about all the (classification) algorithms we have discussed so far
  - What loss functions do they optimize?
  - What decision surfaces do they represent?
  - Pros/Cons?