Dimensionality Reduction
Lecture 23

David Sontag
New York University

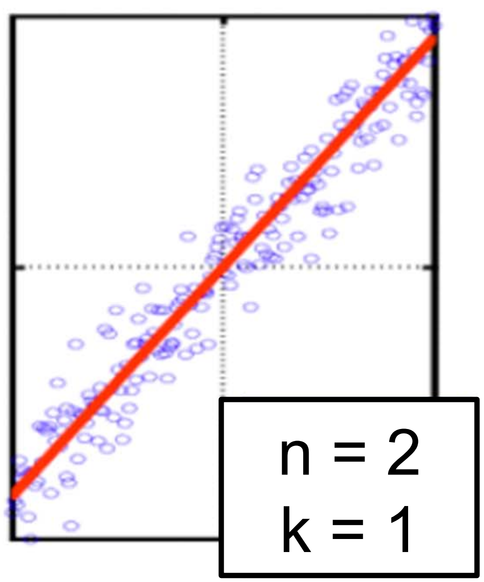
Slides adapted from Carlos Guestrin and Luke Zettlemoyer

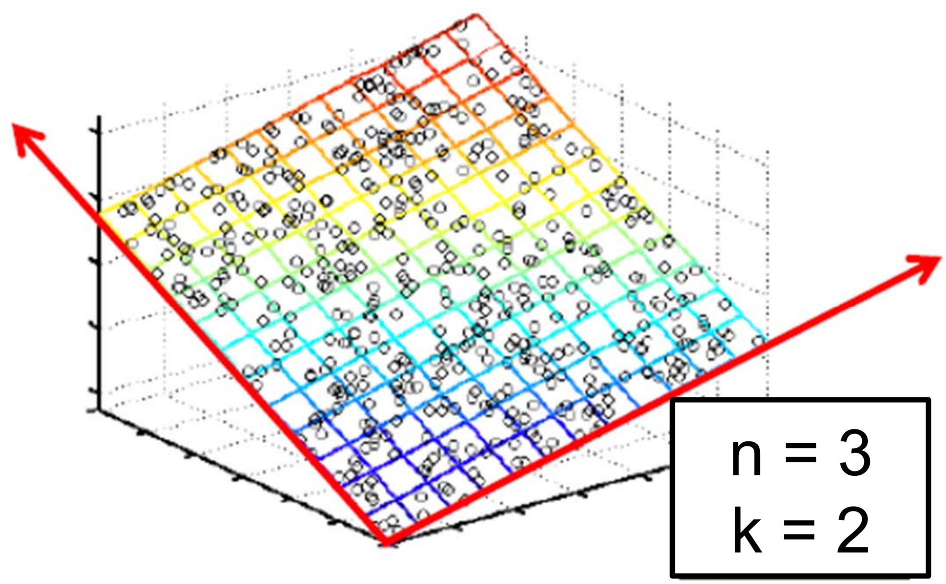
Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has ???, images have ???
- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – show high dimensional data in 2D
  - discover “intrinsic dimensionality” of data
    - high dimensional data that is truly lower dimensional
    - noise reduction

Dimension reduction

- Assumption: data (approximately) lies on a lower dimensional space
- Examples:

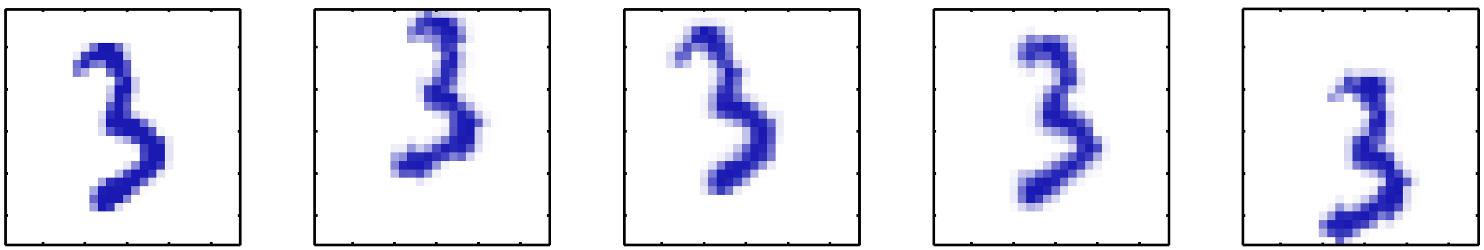




Slide from Yi Zhang

Example (from Bishop)

- Suppose we have a dataset of digits (“3”) perturbed in various ways:



- What operations did I perform? What is the data’s intrinsic dimensionality?
- Here the underlying manifold is *nonlinear*

Lower dimensional projections

- Obtain new feature vector by transforming the original features $x_1 \ldots x_n$

$$z_1 = w_0^{(1)} + \sum_i w_i^{(1)} x_i$$

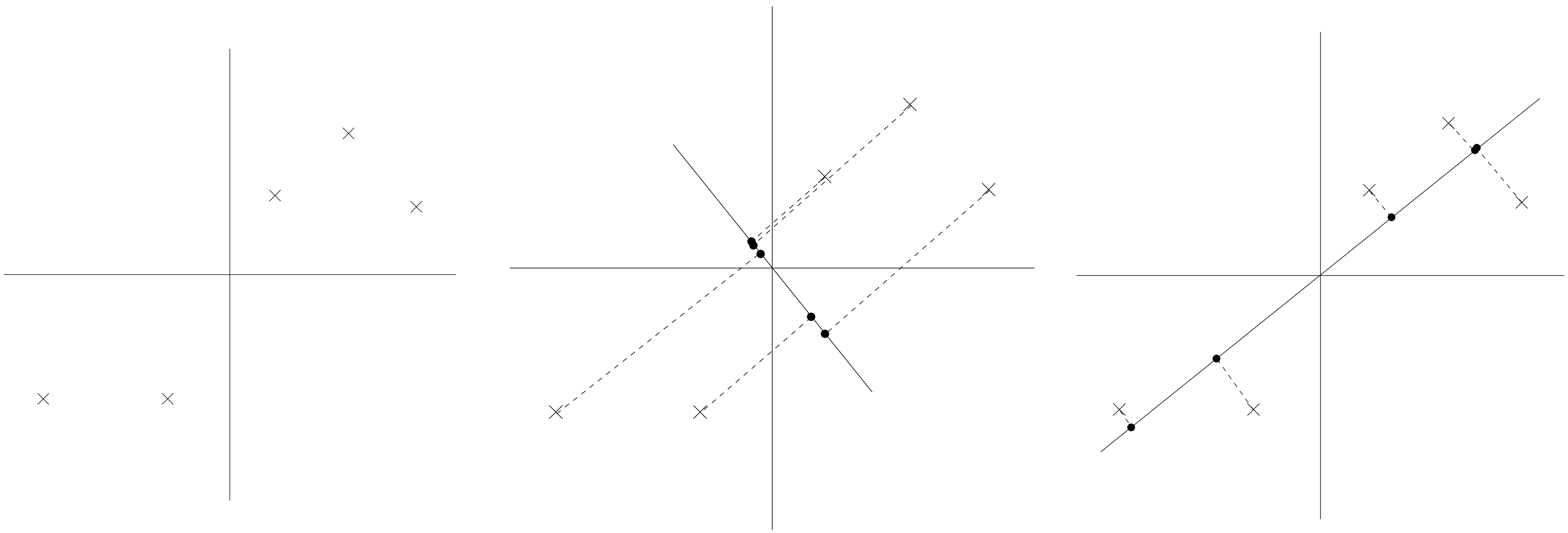
$$\cdots$$

$$z_k = w_0^{(k)} + \sum_i w_i^{(k)} x_i$$

In general will not be invertible – cannot go from *z* back to *x*

- New features are linear combinations of old ones
- Reduces dimension when k<n
- This is typically done in an unsupervised setting
  - just **X**, but no Y

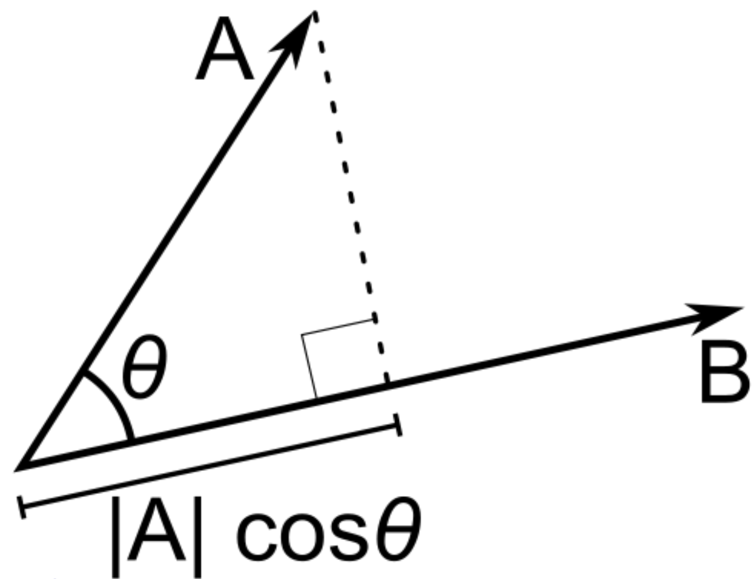
Which projection is better?



From notes by Andrew Ng

Reminder: Vector Projections

- Basic definitions:
  - $A.B = |A||B|\cos\theta$



- Assume $|B|=1$ (unit vector)
  - $A.B = |A|\cos\theta$
  - So, dot product is length of projection!

Using a new basis for the data

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_n)$
  - **select a basis** – set of unit (length 1) basis vectors $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$
    - we consider orthonormal basis:
      - $\mathbf{u}_j \bullet \mathbf{u}_j = 1$, and $\mathbf{u}_j \bullet \mathbf{u}_l = 0$ for $j \neq l$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \bullet \mathbf{u}_j$

$$\boxed{\hat{\mathrm{x}}^i = \bar{\mathrm{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j}$$

Maximize variance of projection

Let $x^{(i)}$ be the i^{th} data point minus the mean.

Choose unit-length u to maximize:

$$\frac{1}{m}\sum_{i=1}^{m}({x^{(i)}}^T u)^2 \quad = \quad \frac{1}{m}\sum_{i=1}^{m} u^T x^{(i)} {x^{(i)}}^T u$$

$$= \quad u^T \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)} {x^{(i)}}^T\right) u.$$

Covariance matrix Σ

Let ||u||=1 and maximize. Using the method of Lagrange multipliers, can show that the solution is given by the principal eigenvector of the covariance matrix! **(shown on board)**

Basic PCA algorithm

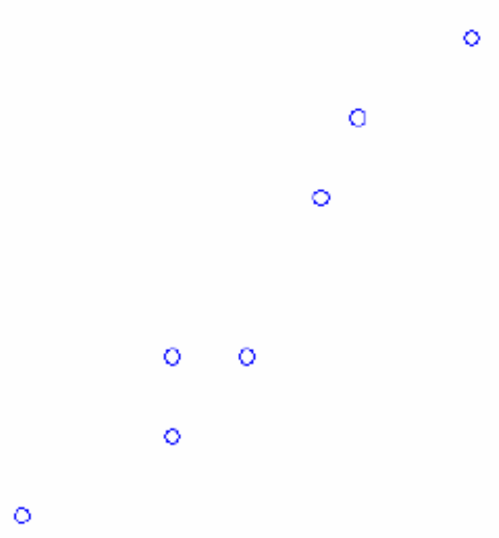
[Pearson 1901,
Hotelling, 1933]

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X_c} \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Compute covariance** matrix:
  - $\Sigma \leftarrow 1/m\ \mathbf{X_c}^T \mathbf{X_c}$
- Find **eigen vectors and values** of Σ
- **Principal components:** k eigen vectors with highest eigen values

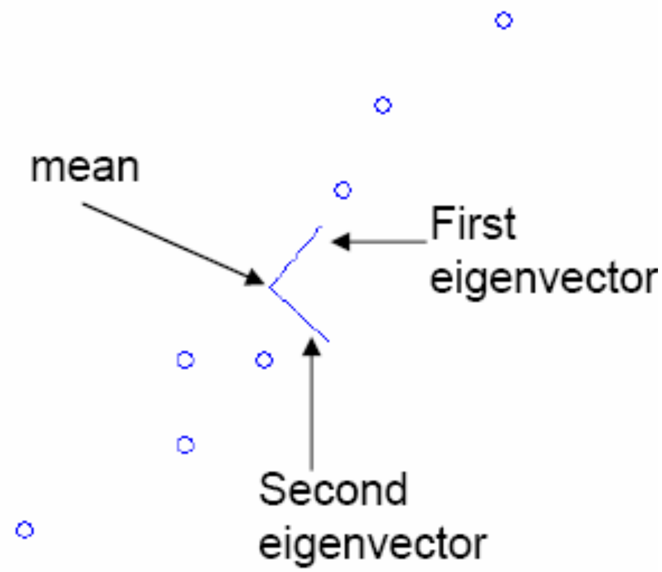
PCA example

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

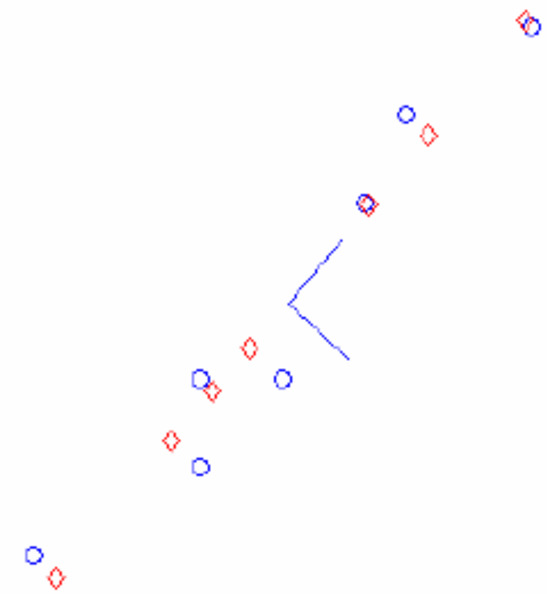
Data:



Projection:



Reconstruction:



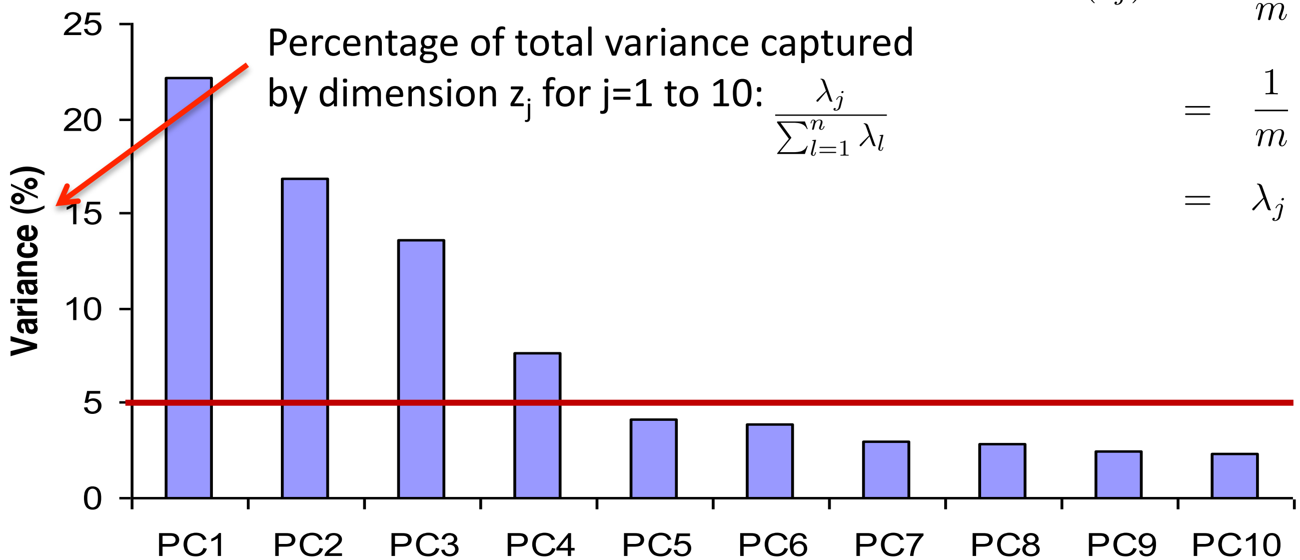
Dimensionality reduction with PCA

In high-dimensional problem, data usually 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.

$$
\begin{aligned}
\mathrm{var}(z_j) &= \frac{1}{m}\sum_{i=1}^{m}(z_j^i)^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}(x^i \cdot u_j)^2 \\
&= \lambda_j
\end{aligned}
$$



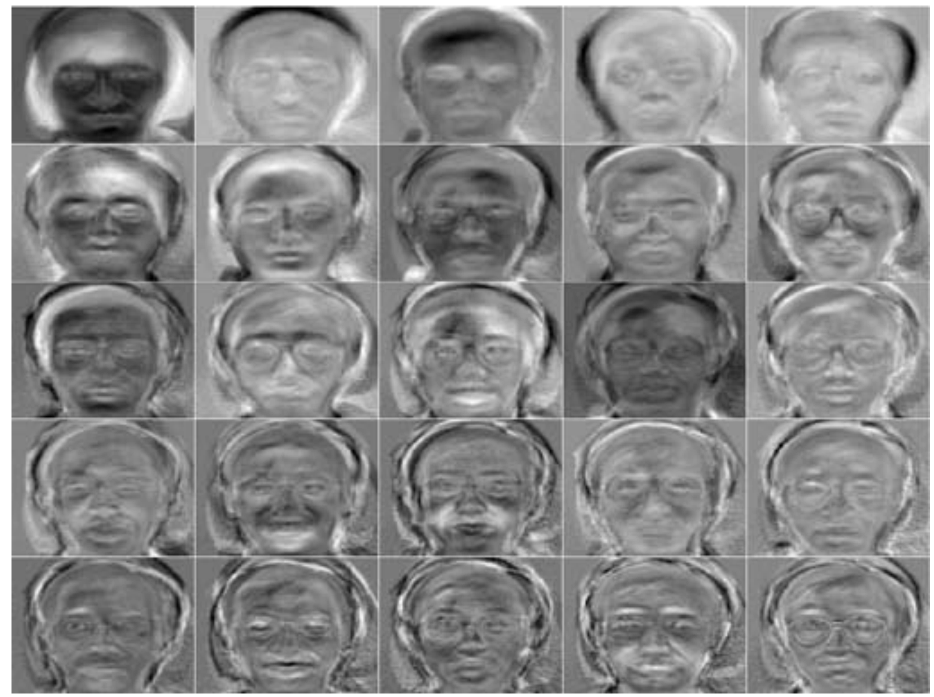
You might lose some information, but if the eigenvalues are small, you don't lose much

Slide from Aarti Singh

Eigenfaces [Turk, Pentland '91]

- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding together (weighted versions of) the principal components:

Scaling up

- Covariance matrix can be really big!
  - Σ is n by n
  - 10000 features can be common!
  - finding eigenvectors is very slow…
- Use singular value decomposition (SVD)
  - Finds k eigenvectors
  - great implementations available, e.g., Matlab svd

SVD

- Write $\mathbf{X} = \mathbf{Z}\ \mathbf{S}\ \mathbf{U}^T$
  - $\mathbf{X} \leftarrow$ data matrix, one row per datapoint
  - $\mathbf{S} \leftarrow$ singular value matrix, diagonal matrix with entries σ_i
    - Relationship between singular values of **X** and eigenvalues of Σ given by $\lambda_i = \sigma_i^2/m$
  - $\mathbf{Z} \leftarrow$ weight matrix, one row per datapoint
    - **Z** times **S** gives coordinate of x_i in eigenspace
  - $\mathbf{U}^T \leftarrow$ singular vector matrix
    - In our setting, each row is eigenvector $\mathbf{u}_j$

PCA using SVD algorithm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X_c} \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Call SVD** algorithm on $\mathbf{X_c}$ – ask for k singular vectors
- **Principal components:** k singular vectors with highest singular values (rows of $\mathbf{U}^T$)
  - **Coefficients:** project each point onto the new vectors

Non-linear methods

- Linear

    Principal Component Analysis (PCA)

    Factor Analysis

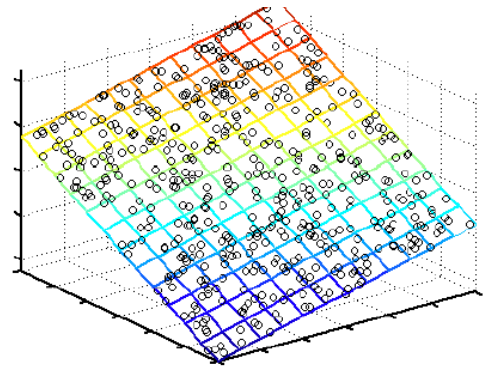
    Independent Component Analysis (ICA)

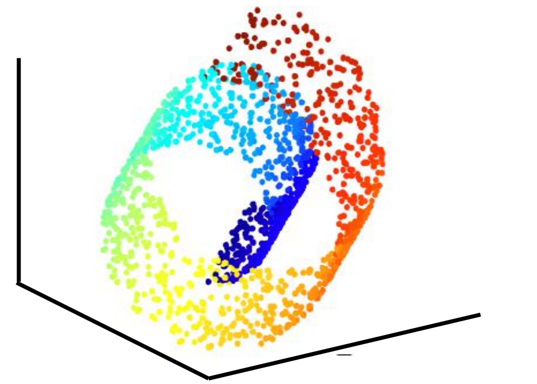
- Nonlinear

    Laplacian Eigenmaps

    ISOMAP

    Local Linear Embedding (LLE)





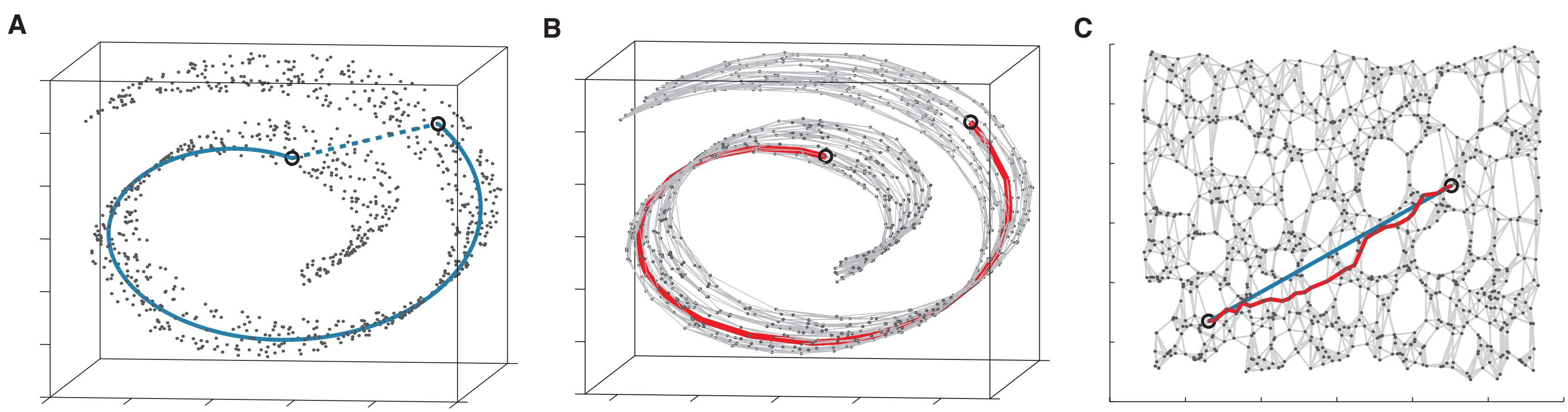
Slide from Aarti Singh

Isomap

Goal: use *geodesic* distance between points (with respect to manifold)

Estimate manifold using graph. Distance between points given by distance of shortest path

Embed onto 2D plane so that Euclidean distance approximates graph distance



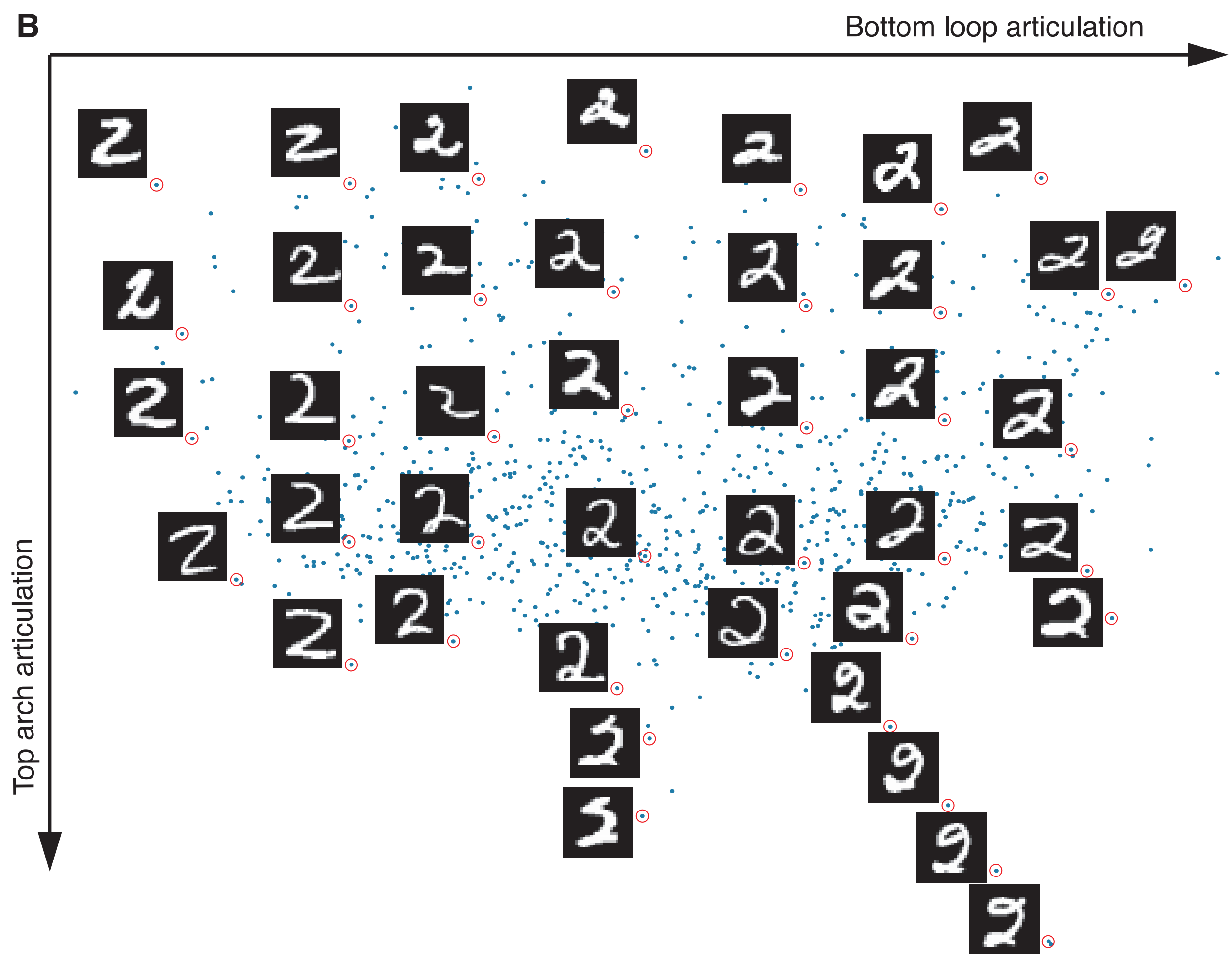
[Tenenbaum, Silva, Langford. Science 2000]

Isomap

Table 1. The Isomap algorithm takes as input the distances $d_X(i,j)$ between all pairs i,j from N data points

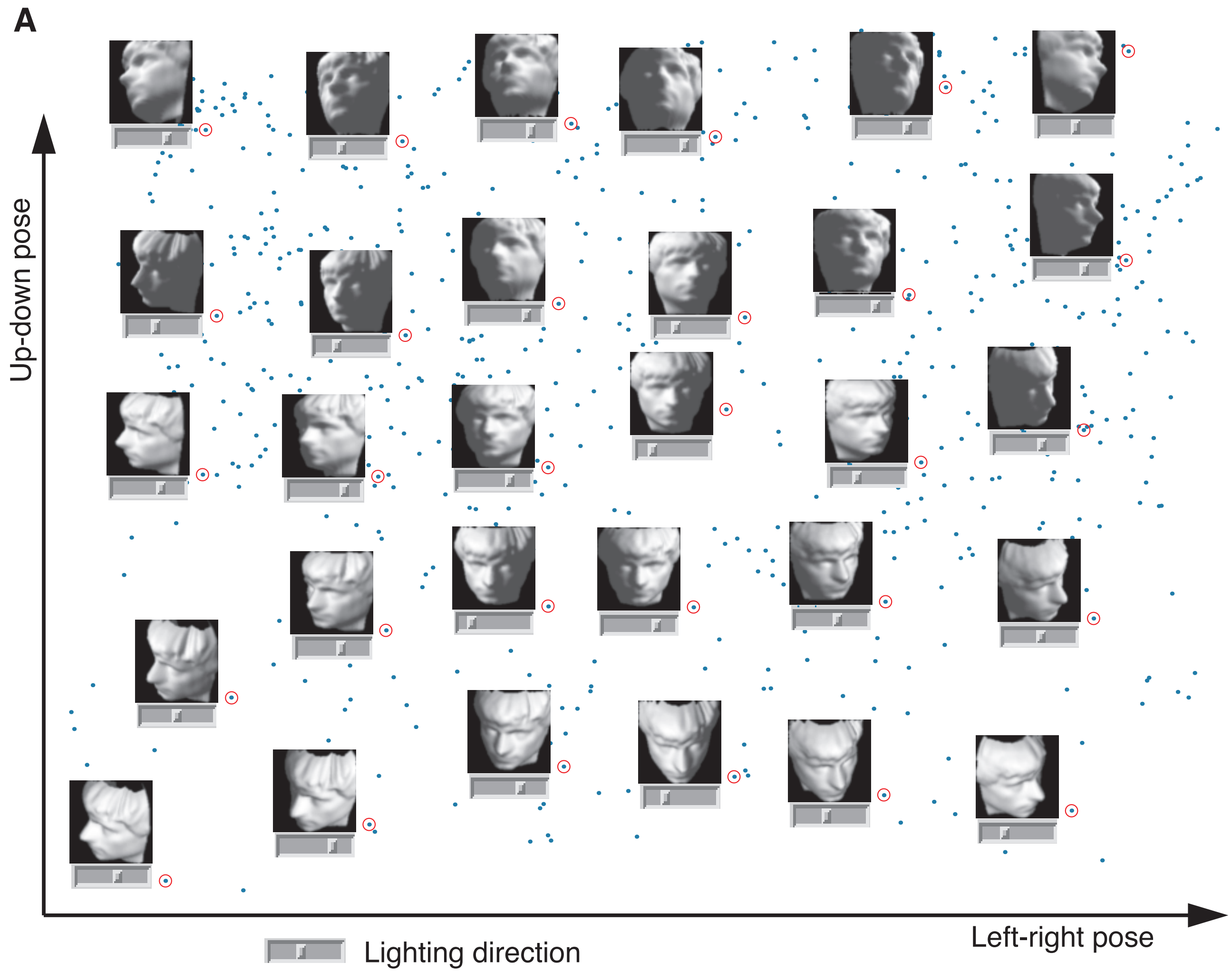
Step		
1	Construct neighborhood graph	Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than ϵ (ϵ-Isomap), or if i is one of the K nearest neighbors of j (K-Isomap). Set edge lengths equal to $d_X(i,j)$.

Isomap



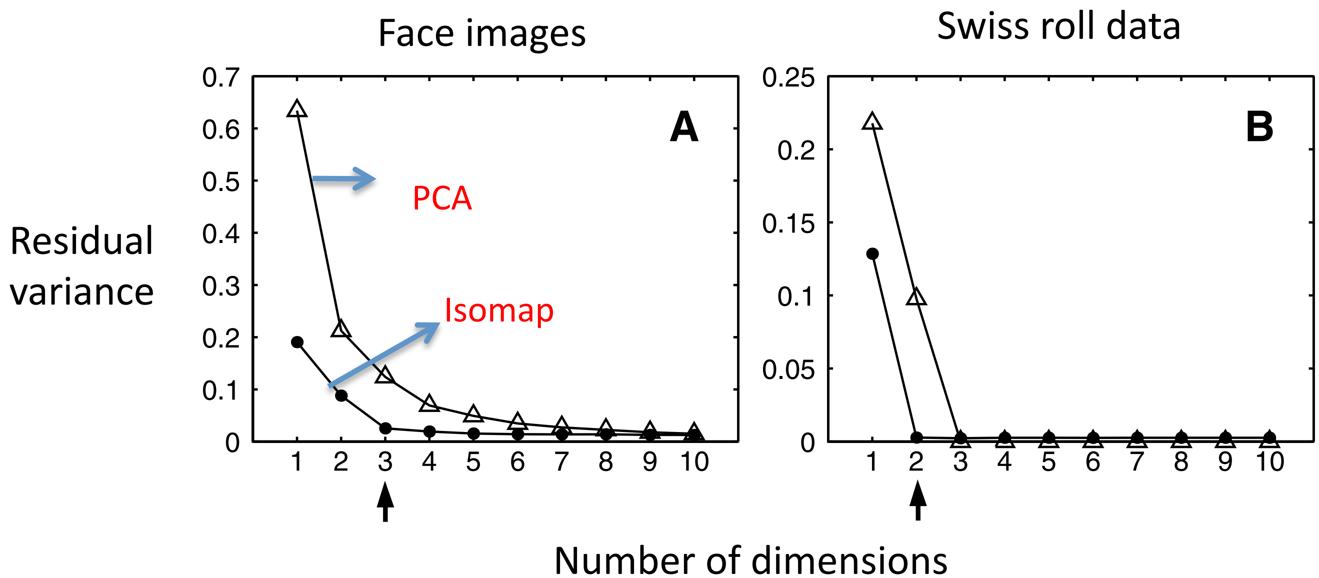
[Tenenbaum, Silva, Langford. Science 2000]

Isomap



[Tenenbaum, Silva, Langford. Science 2000]

Isomap



Face images
Swiss roll data
Residual variance
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
A
PCA
Isomap
1 2 3 4 5 6 7 8 9 10
0.25
0.2
0.15
0.1
0.05
0
B
1 2 3 4 5 6 7 8 9 10
Number of dimensions

What you need to know

- Dimensionality reduction
  - why and when it's important
- Principal component analysis
  - minimizing reconstruction error
  - relationship to covariance matrix and eigenvectors
  - using SVD
- Non-linear dimensionality reduction