Applications of SVD: PCA & MDS

Szymon Rusinkiewicz

COS 302, Fall 2020

PRINCETON UNIVERSITY

Last Time

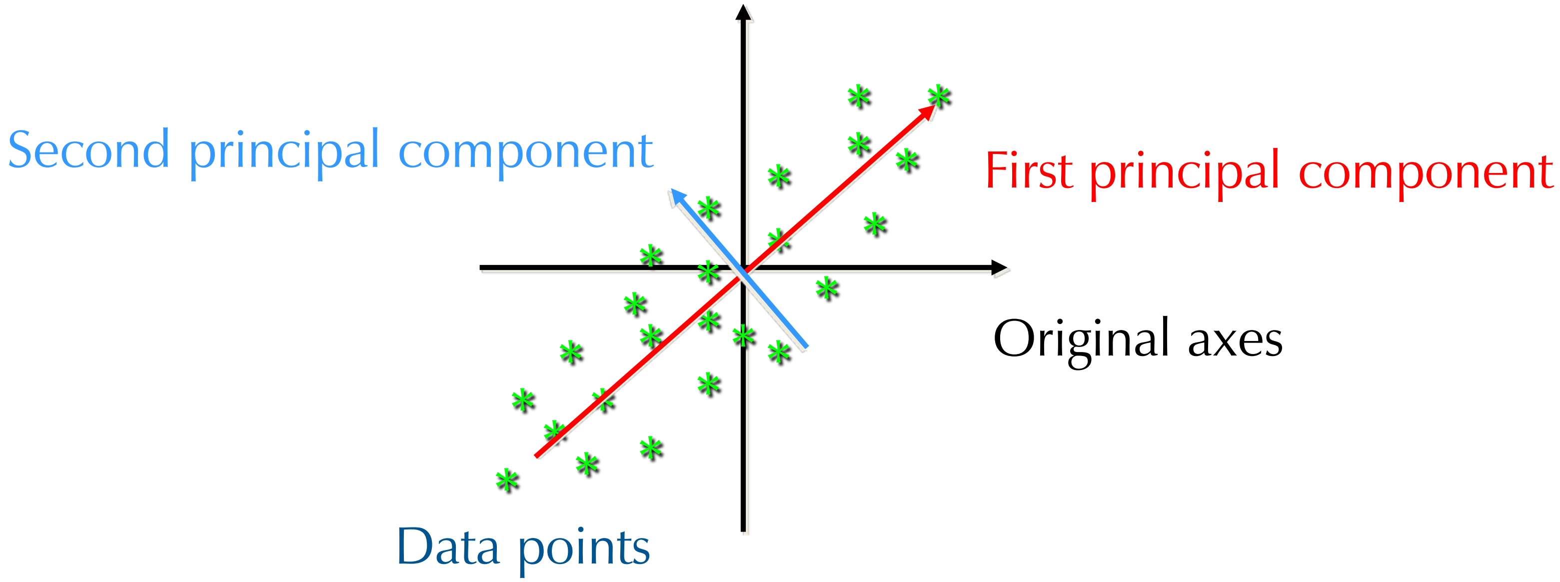
- Singular Value Decomposition
- Solving linear least-squares…
  - without incurring condition-squaring effect of normal equations ($\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$)
  - when $\boldsymbol{A}$ is singular, "fat", or otherwise poorly-specified?
- Total least squares

Today: More Applications of SVD

- Principal Components Analysis
- Multi-dimensional Scaling

Principal Components Analysis (PCA)

- Approximating a high-dimensional data set with a lower-dimensional linear subspace
- Also converts possibly-correlated attributes into uncorrelated attributes



SVD and PCA

- Data matrix with points/examples as rows
- Center data by subtracting mean
- Compute (reduced) SVD
- Columns of $\boldsymbol{V}$ are normalized principal components
- Each w_i indicates importance of corresponding component
- Rows of $\boldsymbol{U}$ are data points expressed in terms of principal components

Dimensionality Reduction

- Map points in high-dimensional space to lower number of dimensions
- (Try to) preserve structure: pairwise distances, etc.
- Useful for further processing:
  - Less computation, fewer parameters
  - Easier to understand, visualize

SVD for Rank-k approximation

- $\boldsymbol{A}$ is $m \times n$ matrix of rank $> k$
- Suppose you want to find best rank-k approximation to $\boldsymbol{A}$
- Take SVD: $\boldsymbol{A} = \boldsymbol{UWV}^{\mathrm{T}}$
- Set all but the largest k singular values of $\boldsymbol{W}$ to 0
- Can form compact representation by eliminating columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ corresponding to zeroed w_i

PCA on Images

- Compression: each new image can be approximated by projection onto first few principal components
- Recognition: for a new image, project onto first few principal components, match feature vectors
- Generation: Adjust contributions of a few principal components to generate new plausible data points

PCA on Images

$$\begin{bmatrix} & & \\ & \boldsymbol{A} & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & \boldsymbol{U} & \\ & & \end{bmatrix} \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_m \end{bmatrix} \begin{bmatrix} & & \\ & \boldsymbol{V} & \\ & & \end{bmatrix}^{\mathrm{T}}$$

Unrolled Images

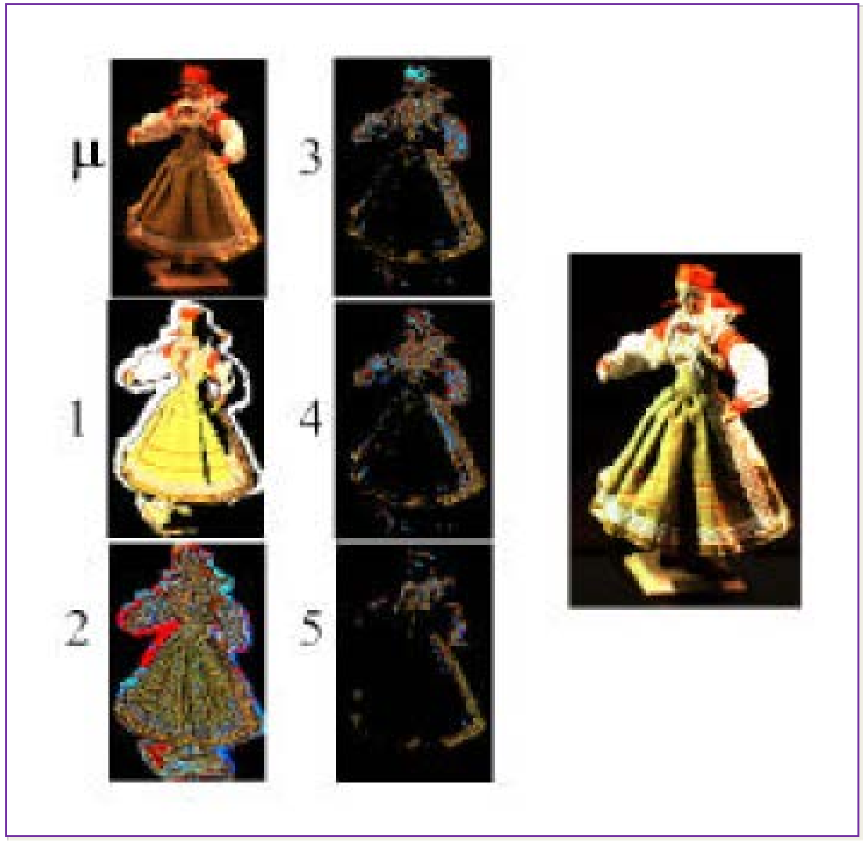
PCA for Relighting

- Images under different illumination

[Matusik & McMillan]

PCA for Relighting

- Images under different illumination
- Most variation captured by first 5 principal components – can re-illuminate by combining only a few images



[Matusik & McMillan]

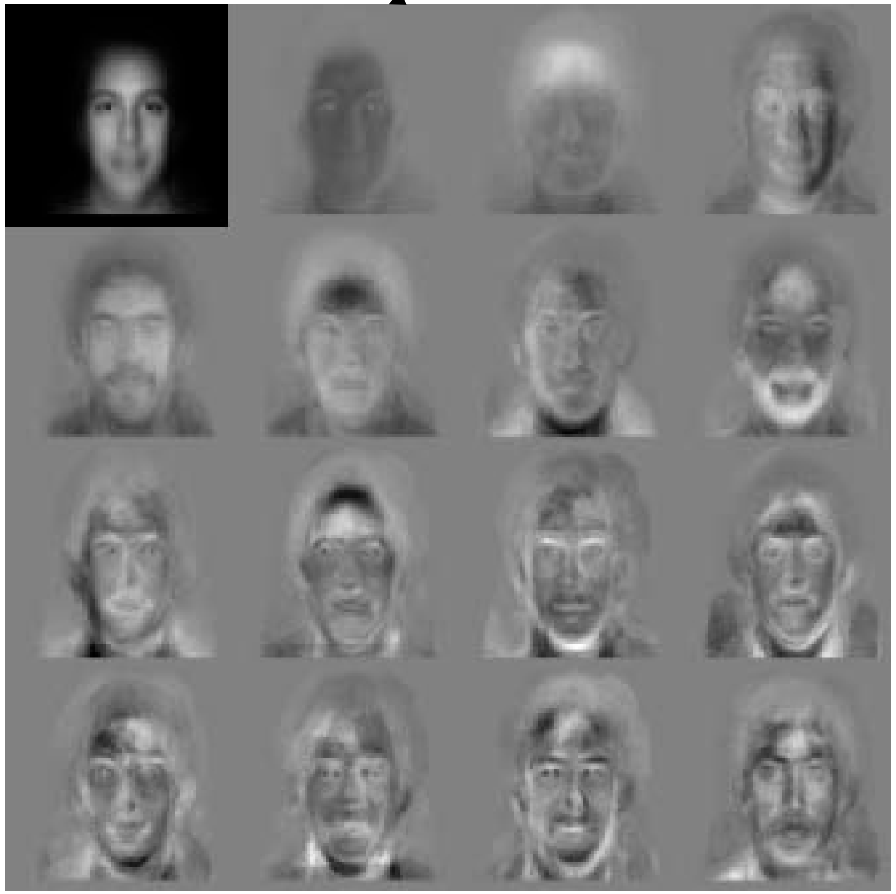
Face Recognition

- Suppose you want to recognize a *particular* face
- How does *this* face differ from average face
  - Not all variations equally important
    (variation in a single pixel relatively unimportant)
- If images are high-dimensional vectors, want to find directions in this space with high variation
  - PCA!

PCA on Faces: “Eigenfaces”

First principal component

Average face



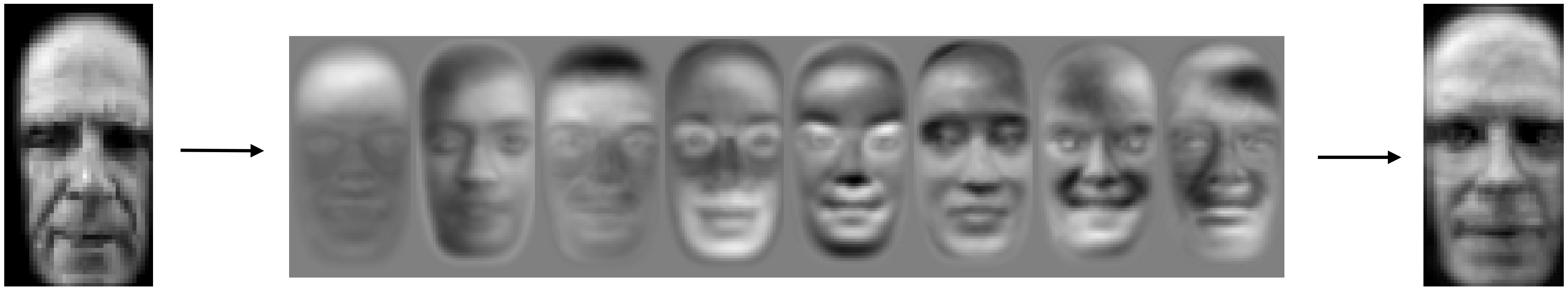
Other components

For all except average,
“gray” = 0,
“white” > 0,
“black” < 0

Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components

$$\text{image} = \text{average} + \sum_{i=1}^{i_{max}} a_i \text{Eigenface}_i$$



Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components
- For a new image: calculate coefficients

$$a_i = (\text{image} - \text{average}) \cdot \text{Eigenface}_i$$

Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components
- For a new image: calculate coefficients
- Is it a face?

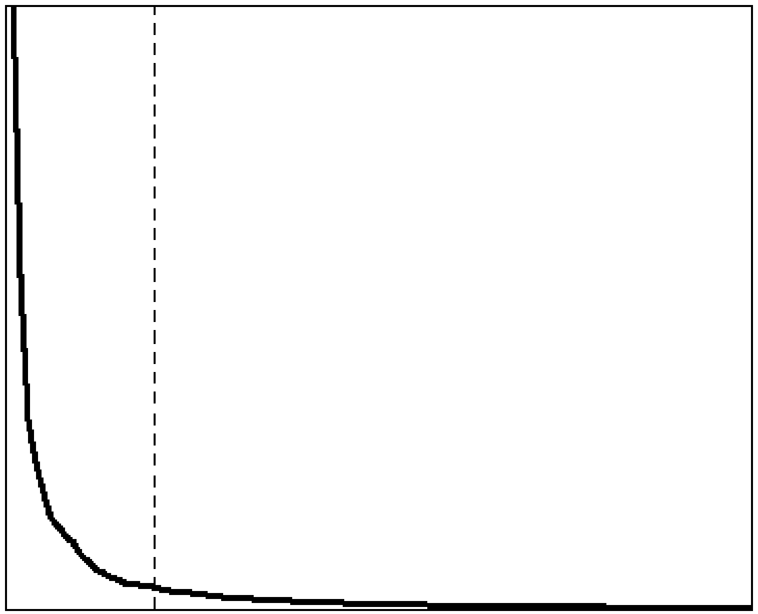
$$\left\| \text{image} - \left(\text{average} + \sum_{i=1}^{i_{max}} a_i \text{Eigenface}_i \right) \right\| < \text{threshold?}$$

Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components
- For a new image: calculate coefficients
- Is it a face?
- If a face, find image in database with closest a_i
  - "Nearest-neighbor classifier"

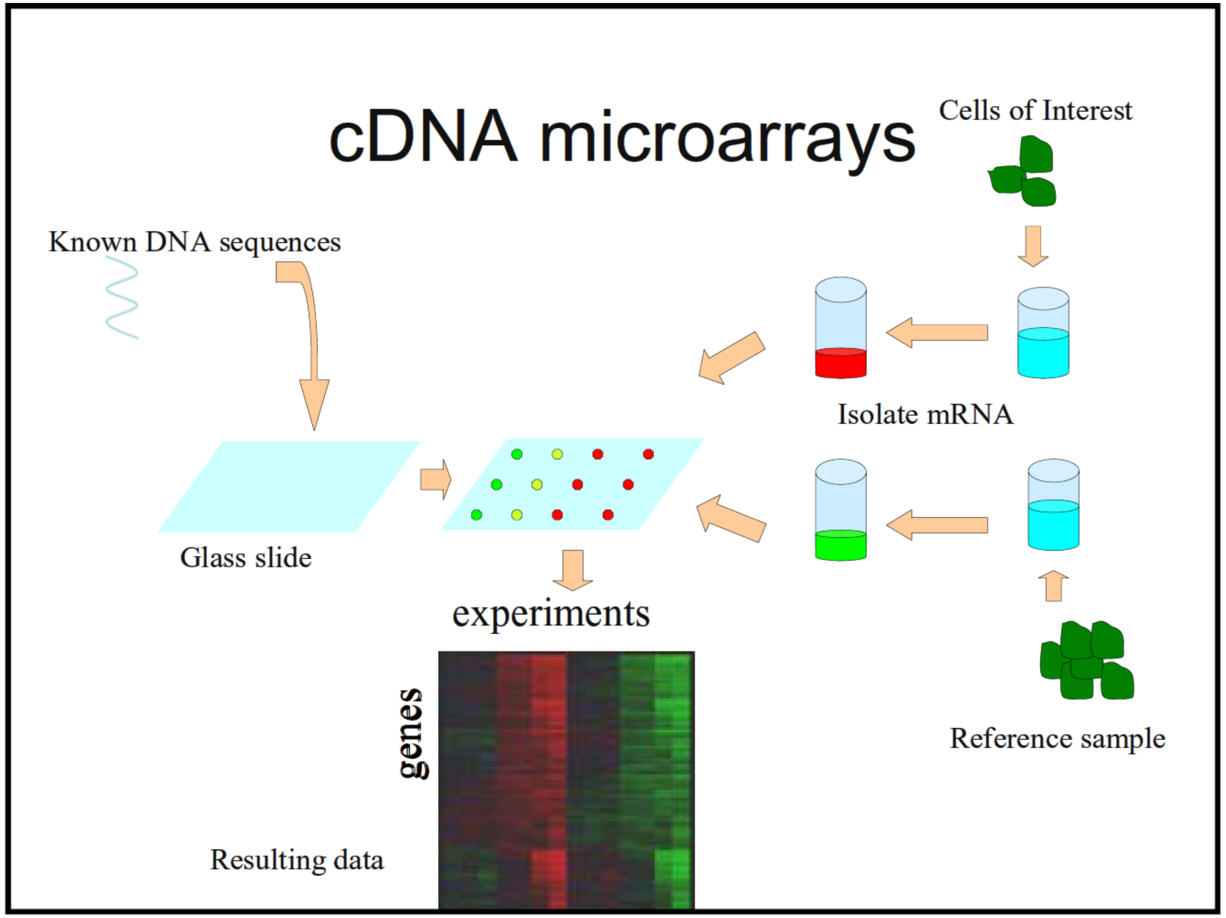
Choosing the Dimension k

- How many eigenfaces to use?
- Look at the decay of the singular values
  - Singular value gives the amount of variance "in the direction" of that eigenface



PCA for DNA Microarrays

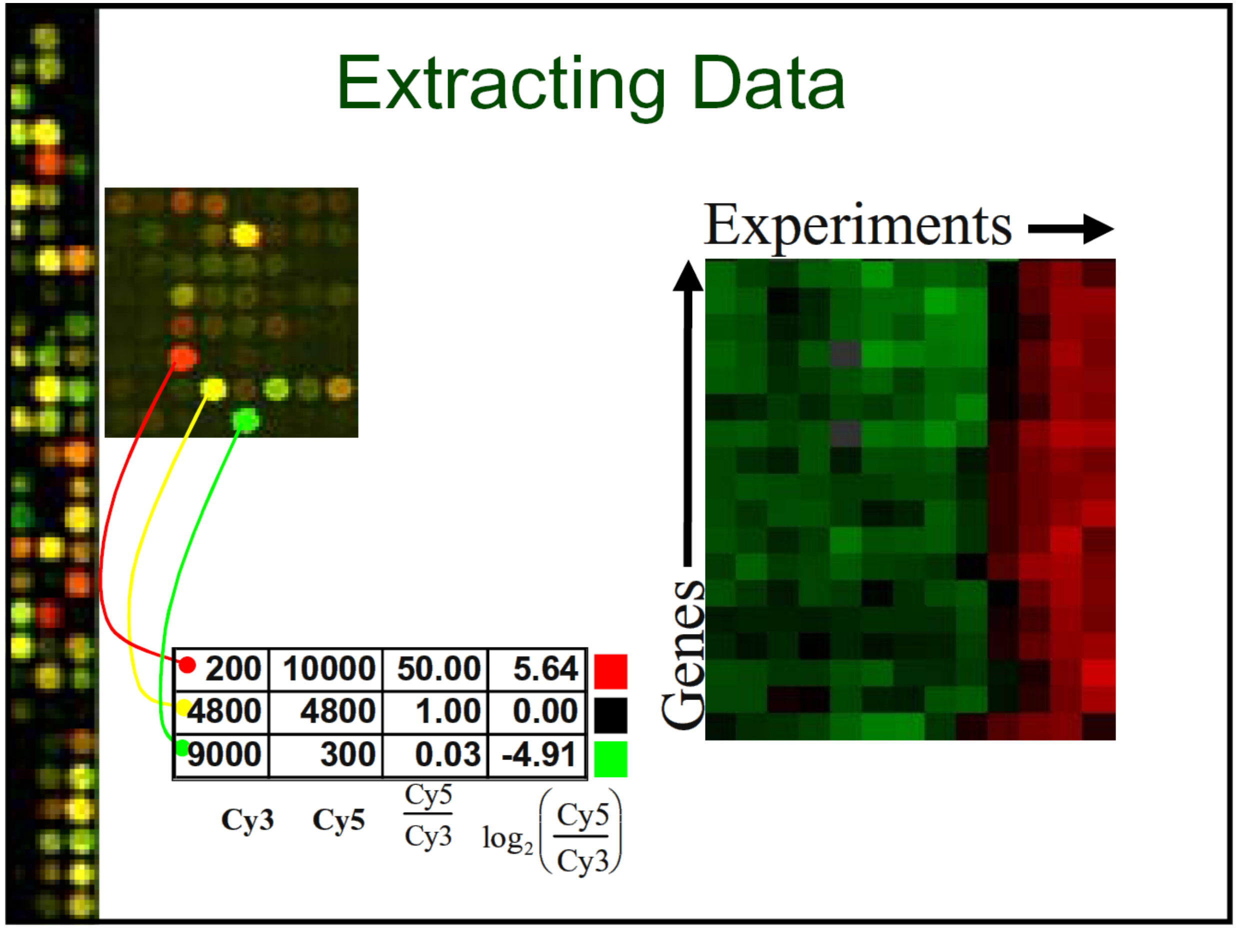
- Measure gene activation under different conditions



[Troyanskaya]

PCA for DNA Microarrays

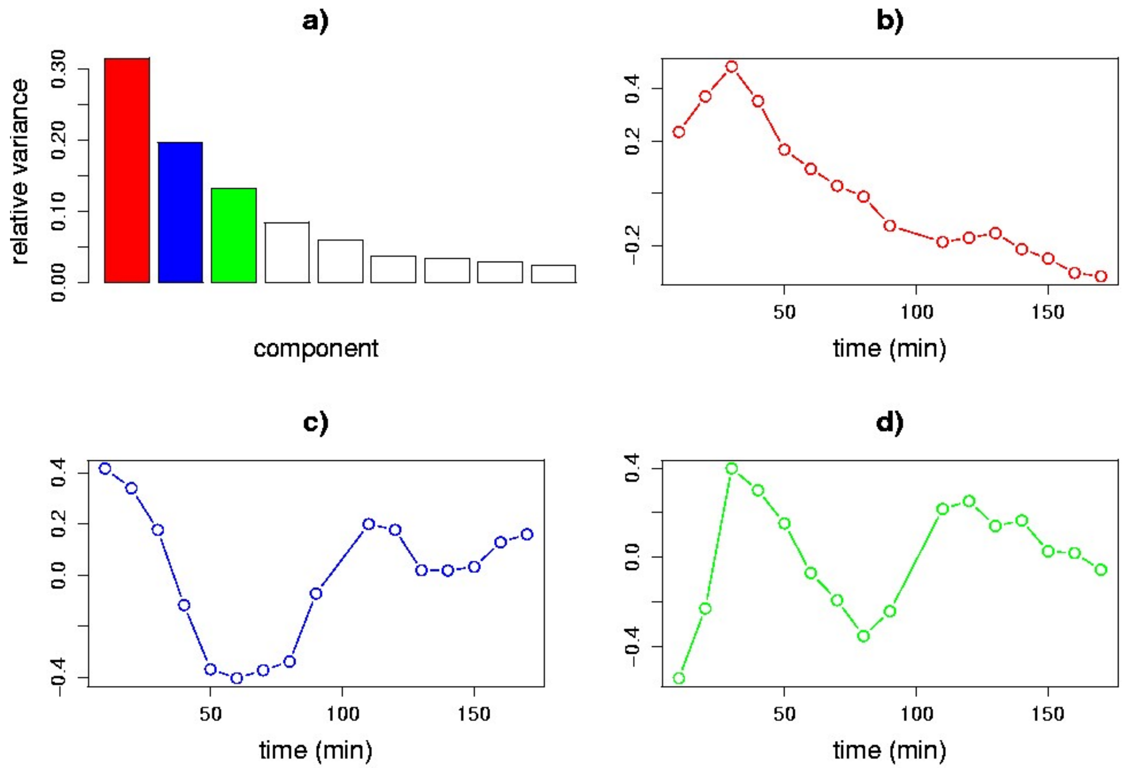
- Measure gene activation under different conditions



[Troyanskaya]

PCA for DNA Microarrays

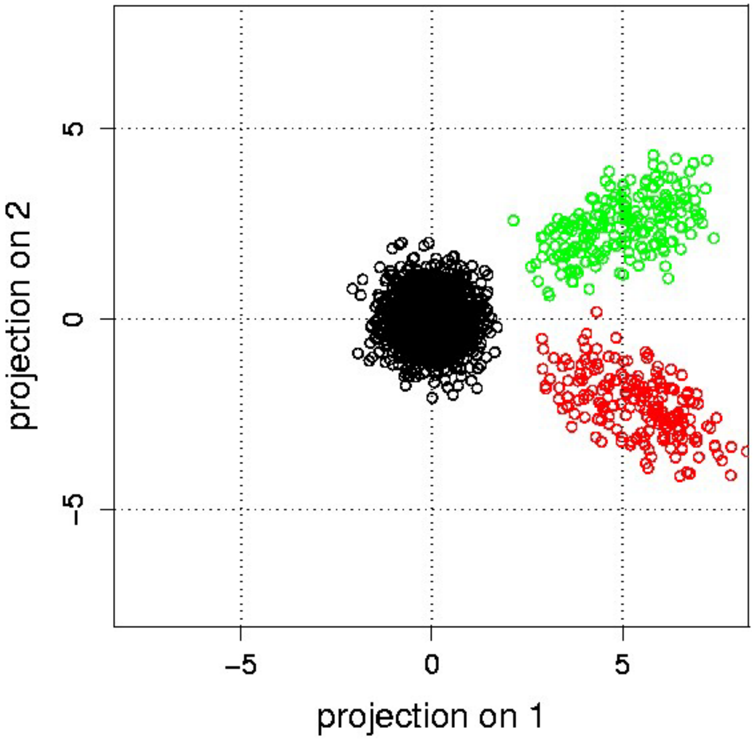
- PCA shows patterns of correlated activation
  - Genes with same pattern might have similar function



[Wall et al.]

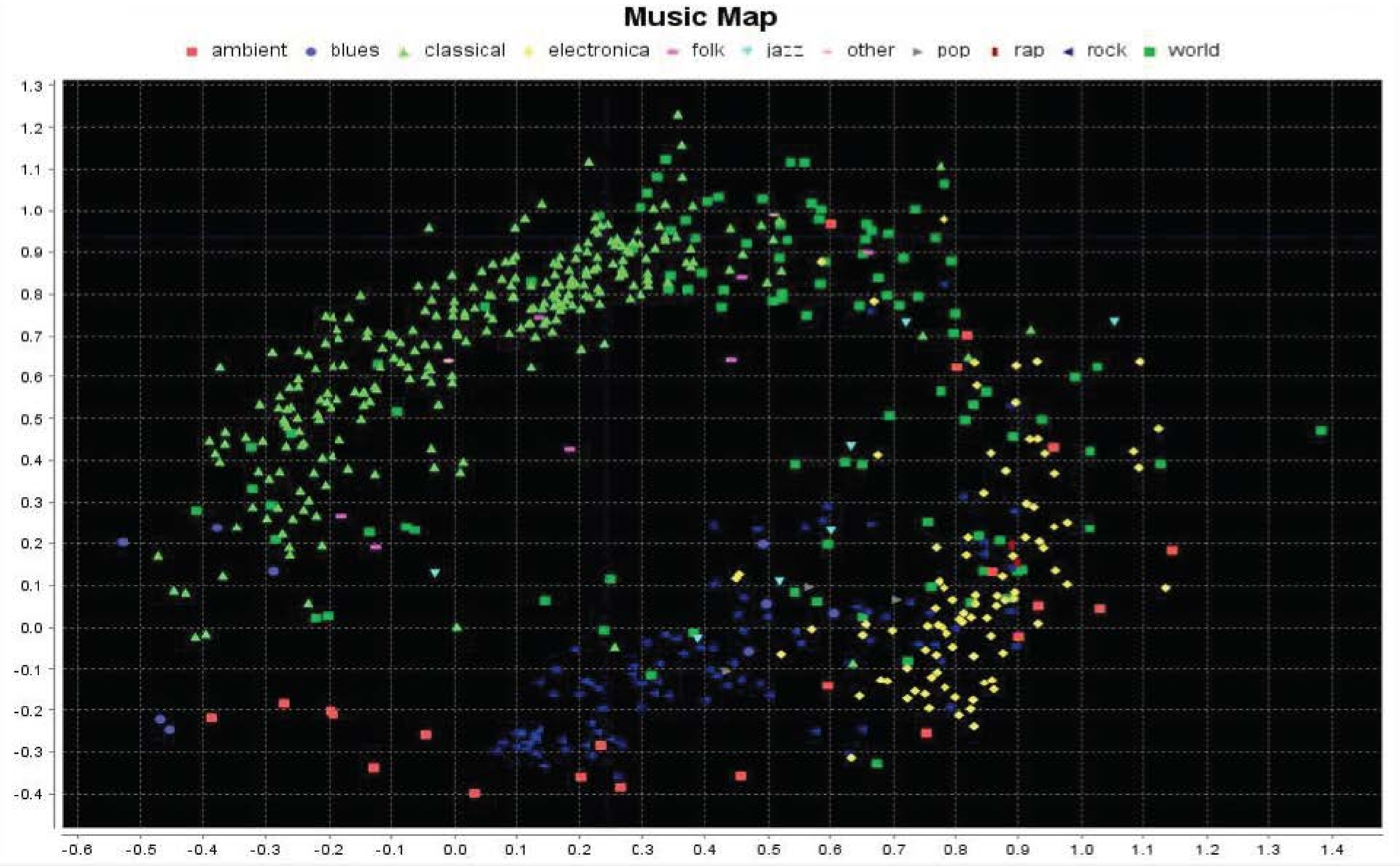
PCA for DNA Microarrays

- PCA shows patterns of correlated activation
  - Genes with same pattern might have similar function



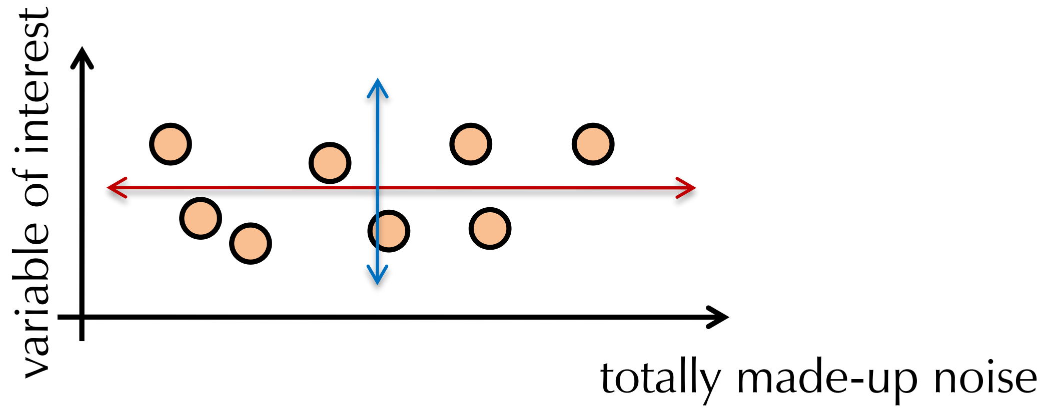
[Wall et al.]

PCA for Music



Practical Considerations for PCA

- Sensitive to scale of each attribute (column)
  - In practice, may "standardize" by scaling each attribute to have unit variance
- Sensitive to noisy attributes
  - Just because a dimension is highly weighted by PCA doesn't mean it's relevant, informative, etc.



Multidimensional Scaling

Multidimensional Scaling

- In some experiments, can only measure similarity or dissimilarity
  - e.g., are responses to stimuli similar or different? How different are they?
  - Frequent in psychophysical experiments, preference surveys, etc.
- Want to recover absolute positions in k-dimensional space

Multidimensional Scaling

- Example: given pairwise distances between cities

	Atl	Chi	Den	Hou	LA	Mia	NYC	SF	Sea	DC
Atlanta	0									
Chicago	587	0								
Denver	1212	920	0							
Houston	701	940	879	0						
LA	1936	1745	831	1374	0					
Miami	604	1188	1726	968	2339	0				
NYC	748	713	1631	1420	2451	1092	0			
SF	2139	1858	949	1645	347	2594	2571	0		
Seattle	2182	1737	1021	1891	959	2734	2406	678	0	
DC	543	597	1494	1220	2300	923	205	2442	2329	0

  – Want to recover (x,y) locations

[Pellacini et al.]

Euclidean MDS

- Formally, let's say we have $n \times n$ matrix $\boldsymbol{D}$ consisting of squared distances $d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$
- Want to recover $n \times k$ matrix $\boldsymbol{X}$ of positions in k-dimensional space

$$D = \begin{pmatrix} 0 & (x_1 - x_2)^2 & (x_1 - x_3)^2 & \\ (x_1 - x_2)^2 & 0 & (x_2 - x_3)^2 & \\ (x_1 - x_3)^2 & (x_2 - x_3)^2 & 0 & \\ & & & \ddots \end{pmatrix}$$

$$X = \begin{pmatrix} (\cdots x_1 \cdots) \\ (\cdots x_2 \cdots) \\ \vdots \end{pmatrix}$$

Euclidean MDS

- Observe that

$$d_{ij}^2 = (x_i - x_j)^2 = x_i^2 - 2x_i x_j + x_j^2$$

- Strategy: convert matrix ***D*** of d_{ij}^2 into matrix ***B*** of $x_i x_j$
  - "Centered" distance matrix
  - Then decompose $\boldsymbol{B} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$

Euclidean MDS

- Centering:
  - Sum of row *i* of ***D*** = sum of column *i* of ***D*** =

$$s_i = \sum_j d_{ij}^2 = \sum_j x_i^2 - 2x_i x_j + x_j^2$$

$$= nx_i^2 - 2x_i \sum_j x_j + \sum_j x_j^2$$

  - Sum of all entries in ***D*** =

$$s = \sum_i s_i = 2n \sum_i x_i^2 - 2\left(\sum_i x_i\right)^2$$

Euclidean MDS

- Choose $\Sigma x_i = 0$
  - Solution will have average position at origin
  - Then,

$$s_i = nx_i^2 + \sum_j x_j^2, \quad s = 2n\sum_j x_j^2$$

$$d_{ij}^2 - \tfrac{1}{n}s_i - \tfrac{1}{n}s_j + \tfrac{1}{n^2}s = -2x_i x_j$$

- So, to get ***B***:
  - compute row (or column) sums
  - compute sum of sums
  - apply above formula to each entry of ***D***
  - Divide by –2

Factoring $B = XX^{\mathrm{T}}$ using SVD

- Now have $\boldsymbol{B}$, want to factor into $\boldsymbol{XX}^{\mathrm{T}}$
- If $\boldsymbol{X}$ is $n \times k$, $\boldsymbol{B}$ must have rank k
- Take SVD, set all but top k singular values to 0
  - Eliminate corresponding columns of $\boldsymbol{U}$ and $\boldsymbol{V}$
  - Have $\boldsymbol{B'} = \boldsymbol{U'W'V'}^{\mathrm{T}}$
  - $\boldsymbol{B'}$ is square and symmetric, so $\boldsymbol{U'} = \boldsymbol{V'}$
  - Take $\boldsymbol{X} = \boldsymbol{U'}$ times square root of $\boldsymbol{W'}$

Multidimensional Scaling

- Result ($k = 2$):

[Pellacini et al.]

Another application

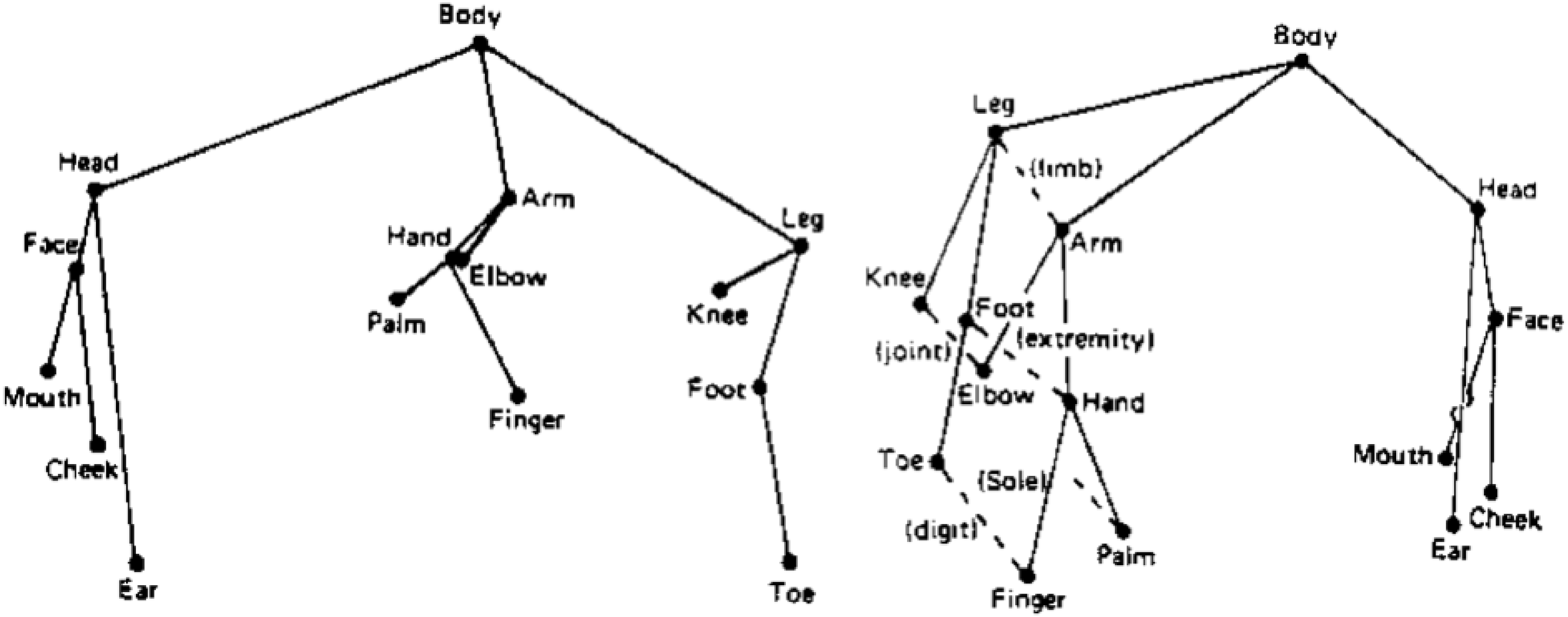
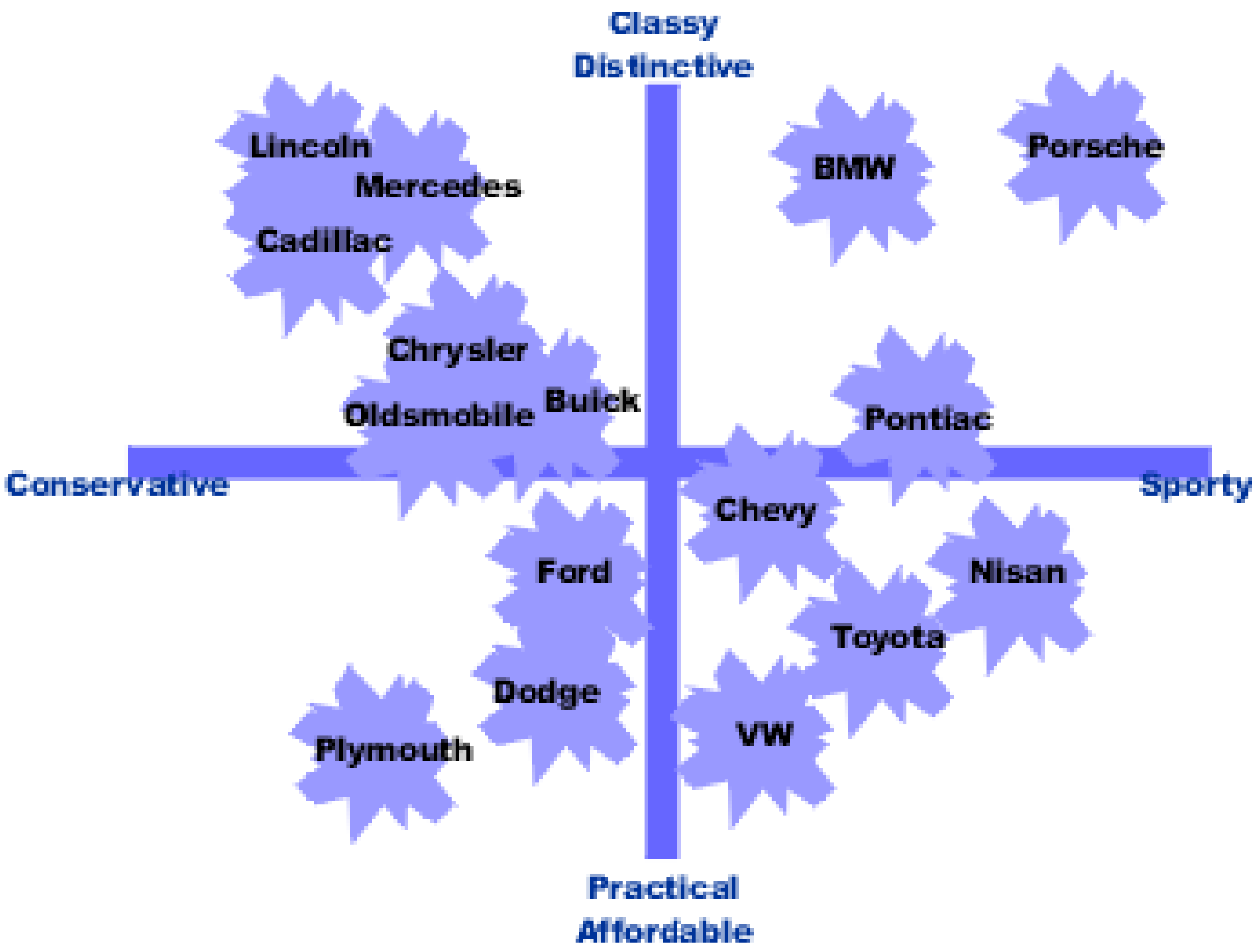


Figure 2 (a) RMDS of children's similarity judgments about 15 body parts: (b) RMDS of adults' similarity judgments about 15 body parts.

From Young 1985 / Jacobowitz 1973

Perceptual Mapping for Marketing



Multidimensional Scaling

- Caveat: actual axes, center not necessarily what you want (can't recover them!)
- This is "classical" or "Euclidean" MDS [Torgerson 52]
  - Distance matrix assumed to be actual Euclidean distance
- More sophisticated versions available
  - "Non-metric MDS": not Euclidean distance, sometimes just *inequalities*
  - Replicated MDS: for multiple data sources (e.g. people)
  - "Weighted MDS": account for observer bias