UNIVERSITY OF MINNESOTA TWIN CITIES

Dimension reduction methods: Algorithms and Applications

Yousef Saad
Department of Computer Science and Engineering

University of Minnesota

Université du Littoral- Calais

July 11, 2016

First..

- ... to the memory of Mohammed Bellalij

Introduction, background, and motivation

Common goal of data mining methods: to extract meaningful information or patterns from data. Very broad area – includes: data analysis, machine learning, pattern recognition, information retrieval, ...

- Main tools used: linear algebra; graph theory; approximation theory; optimization; ...
- In this talk: emphasis on dimension reduction techniques and the interrelations between techniques

Introduction: a few factoids

- Data is growing exponentially at an “alarming” rate:
  - 90% of data in world today was created in last two years
  - Every day, 2.3 Million terabytes (2.3 $\times 10^{18}$ bytes) created
- Mixed blessing: Opportunities & big challenges.
- Trend is re-shaping & energizing many research areas ...
- ... including my own: numerical linear algebra

Topics

➤ Focus on two main problems

– Information retrieval

– Face recognition

➤ and 2 types of dimension reduction methods

– Standard subspace methods [SVD, Lanczos]

– Graph-based methods

Major tool of Data Mining: Dimension reduction

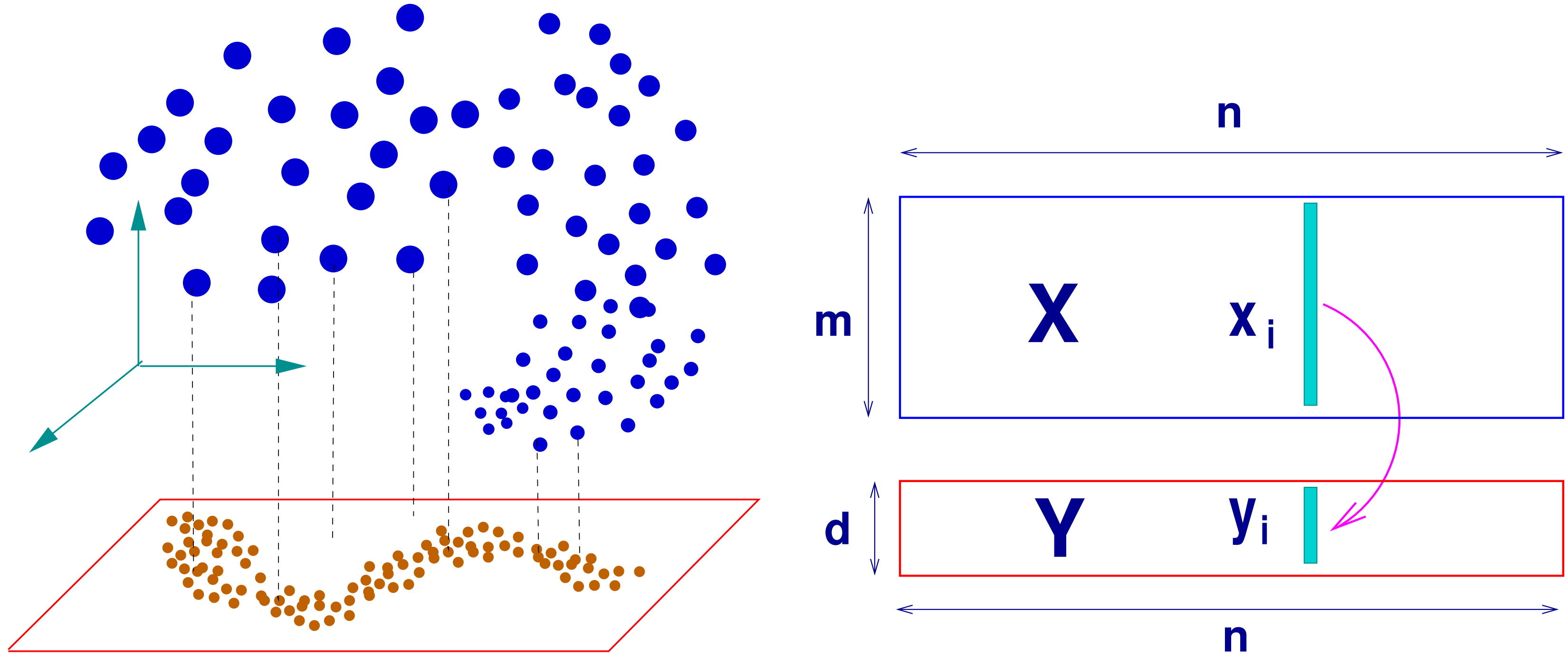
- Goal is not as much to reduce size (& cost) but to:

  - Reduce noise and redundancy in data before performing a task [e.g., classification as in digit/face recognition]
  - Discover important 'features' or 'paramaters'

The problem: Given: $X = [x_1, \cdots, x_n] \in \mathbb{R}^{m \times n}$, find a low-dimens. representation $Y = [y_1, \cdots, y_n] \in \mathbb{R}^{d \times n}$ of X

- Achieved by a mapping $\Phi : x \in \mathbb{R}^m \longrightarrow y \in \mathbb{R}^d$ so:

$$\phi(x_i) = y_i, \quad i = 1, \cdots, n$$



- Φ may be linear : $y_i = W^\top x_i$, i.e., $Y = W^\top X$, ..
- ... or nonlinear (implicit).
- Mapping Φ required to: Preserve proximity? Maximize variance? Preserve a certain graph?

Example: Principal Component Analysis (PCA)

In *Principal Component Analysis* $\boldsymbol{W}$ is computed to maximize variance of projected data:

$$\max_{\boldsymbol{W}\in\mathbb{R}^{m\times d};\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}} \quad \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{y}_j \right\|_2^2, \quad \boldsymbol{y}_i = \boldsymbol{W}^\top \boldsymbol{x}_i.$$

- Leads to maximizing

$$\text{Tr}\left[\boldsymbol{W}^\top(\boldsymbol{X}-\boldsymbol{\mu}\boldsymbol{e}^\top)(\boldsymbol{X}-\boldsymbol{\mu}\boldsymbol{e}^\top)^\top\boldsymbol{W}\right], \quad \boldsymbol{\mu} = \tfrac{1}{n}\Sigma_{i=1}^{n}\boldsymbol{x}_i$$

- Solution $\boldsymbol{W} = \{$ dominant eigenvectors $\}$ of the covariance matrix $\equiv$ Set of left singular vectors of $\bar{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{e}^\top$

SVD:

$$\bar{X} = U\Sigma V^\top, \quad U^\top U = I, \quad V^\top V = I, \quad \Sigma = \text{Diag}$$

- Optimal $W = U_d \equiv$ matrix of first d columns of U
- Solution W also minimizes 'reconstruction error' ..

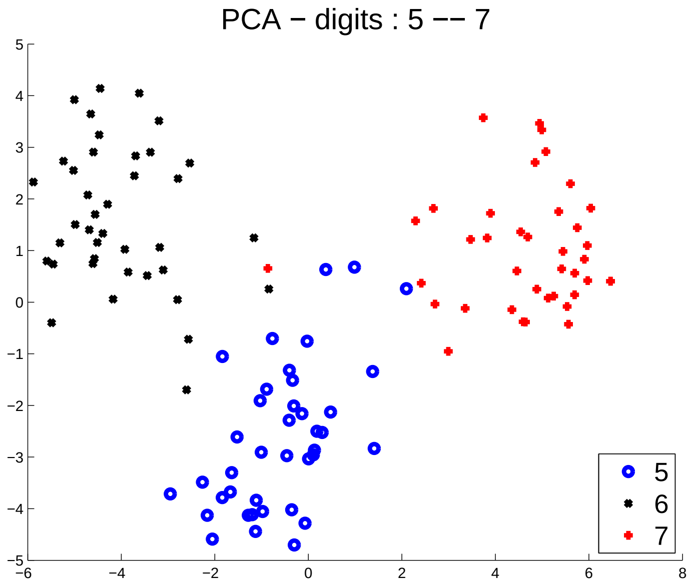
$$\sum_i \|x_i - WW^T x_i\|^2 = \sum_i \|x_i - W y_i\|^2$$

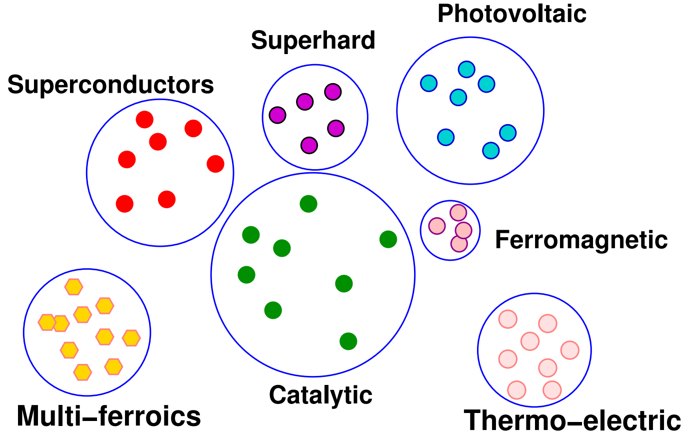
- In some methods recentering to zero is not done, i.e., $\bar{X}$ replaced by X.

Unsupervised learning

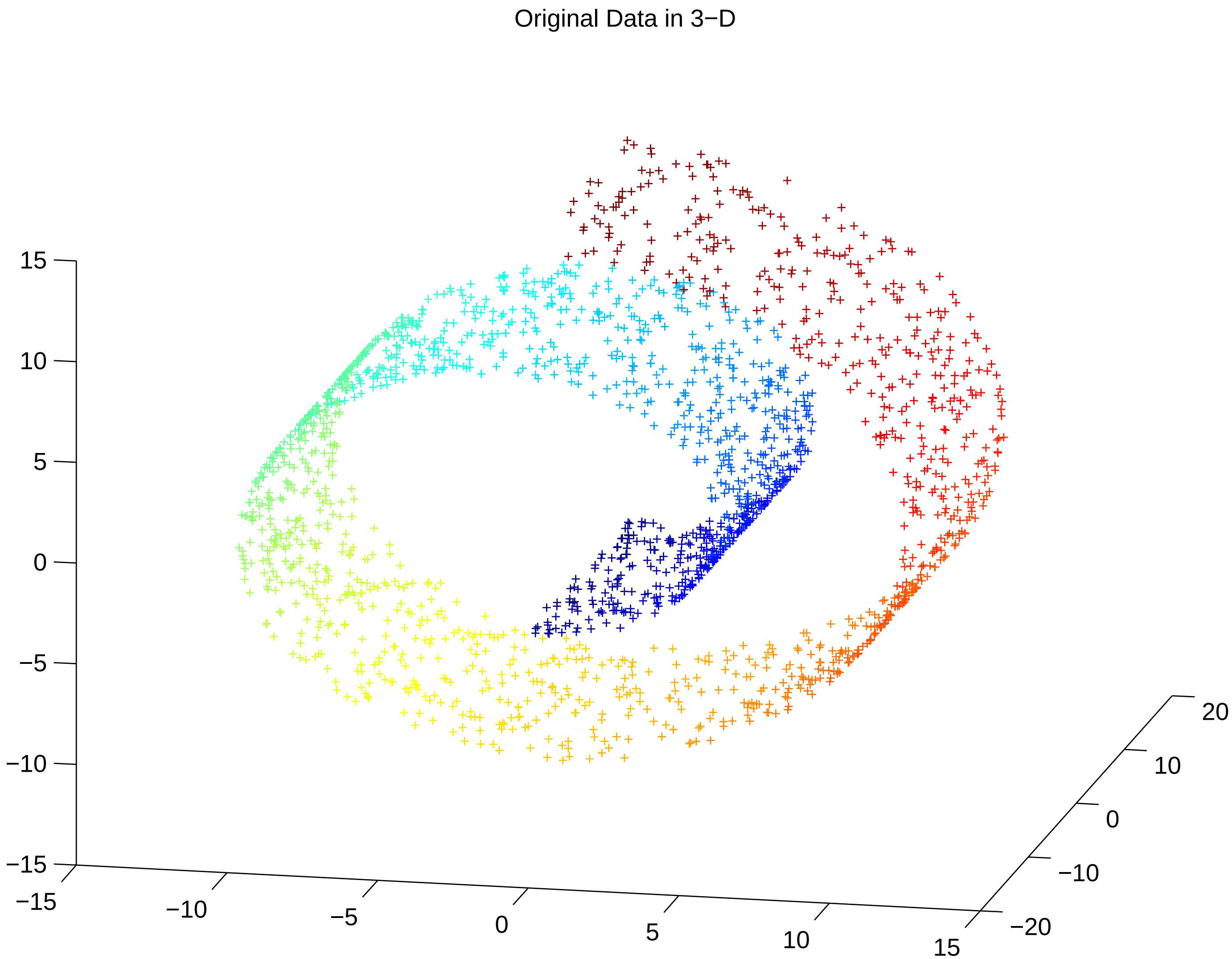
"Unsupervised learning" : methods that do not exploit known labels

- Example of digits: perform a 2-D projection
- Images of same digit tend to cluster (more or less)
- Such 2-D representations are popular for visualization
- Can also try to find natural clusters in data, e.g., in materials
- Basic clusterning technique: K-means

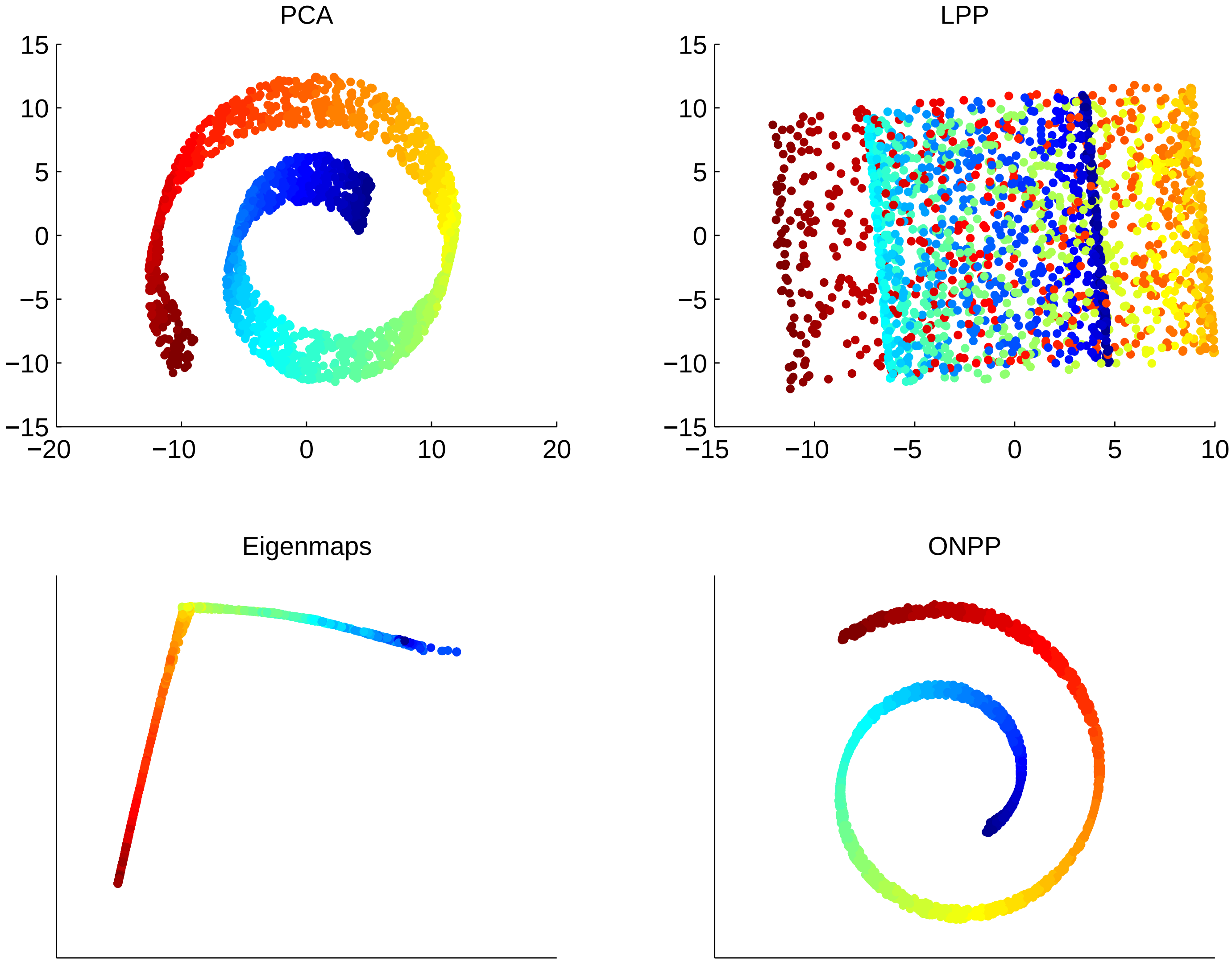




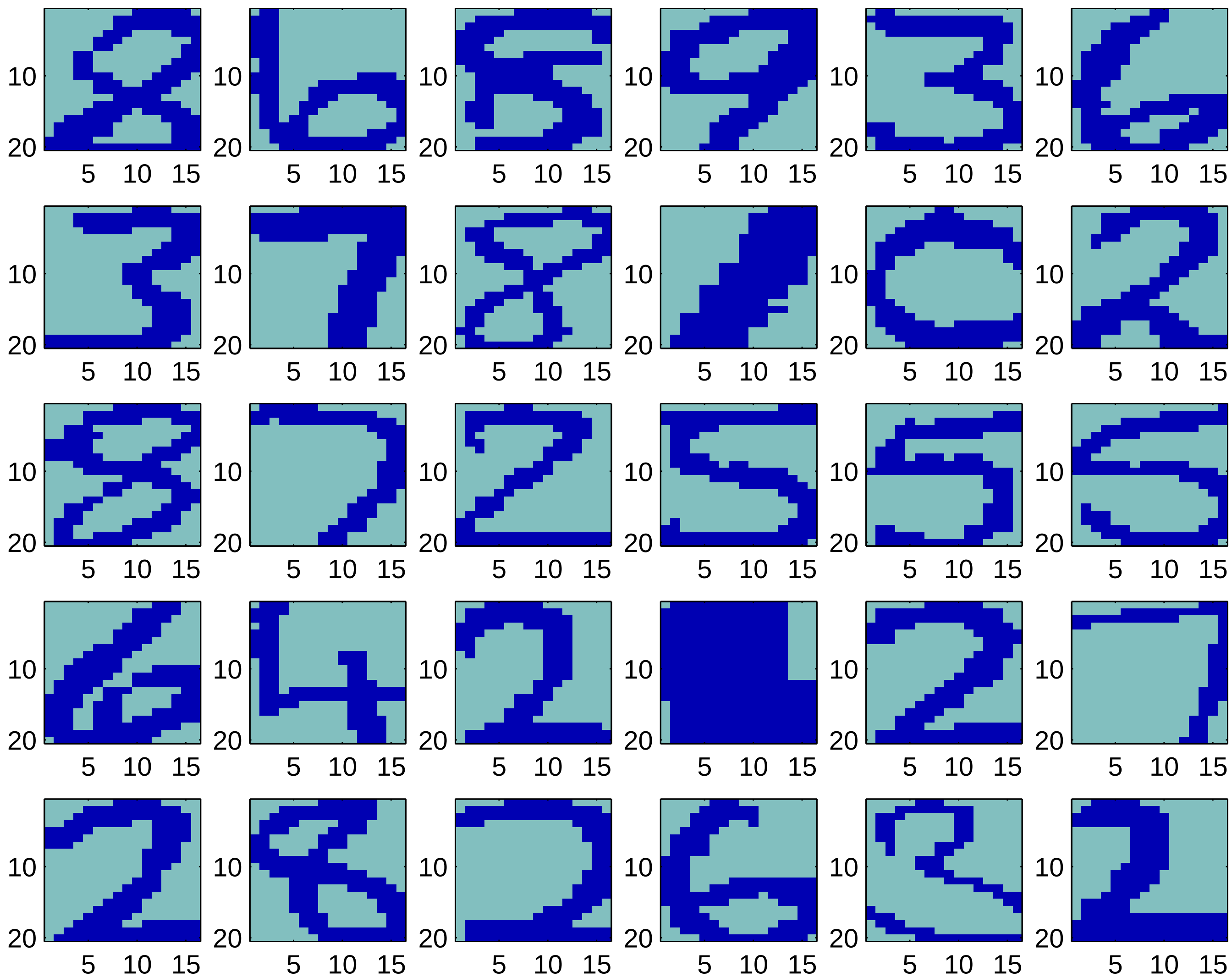
Example: The 'Swill-Roll' (2000 points in 3-D)



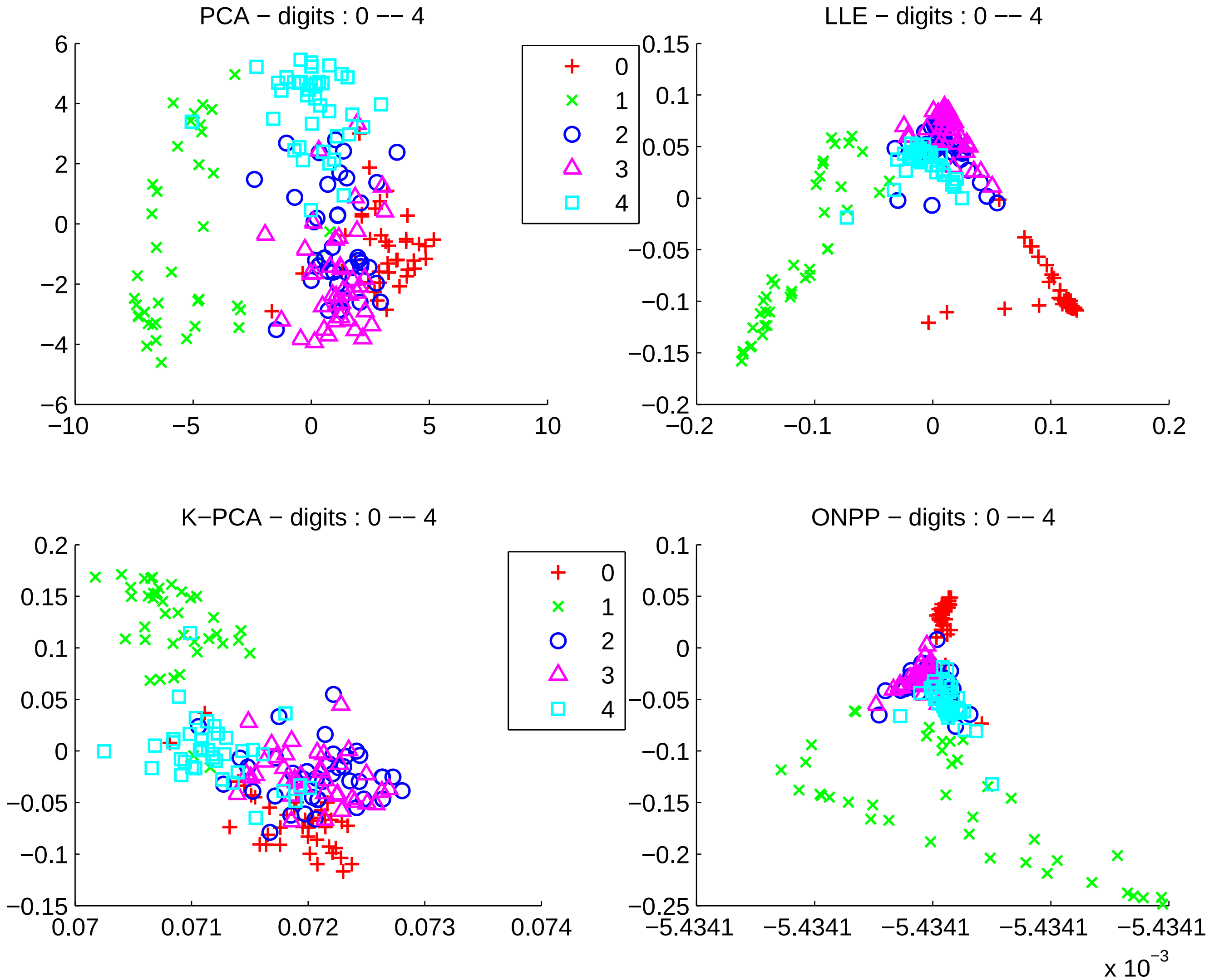
2-D 'reductions':



Example: Digit images (a random sample of 30)

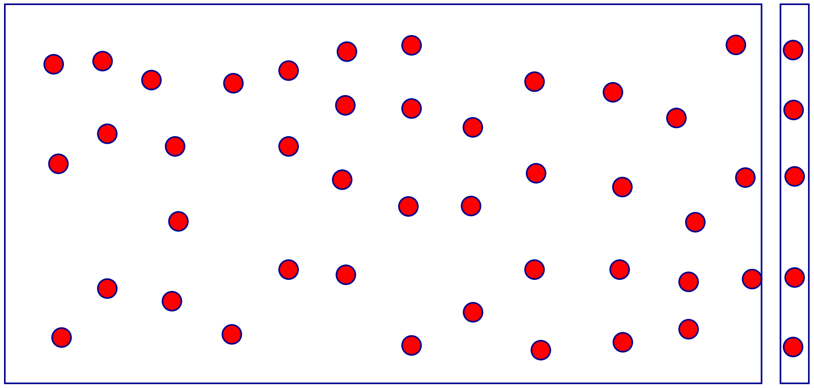


2-D 'reductions':



APPLICATION: INFORMATION RETRIEVAL

Application: Information Retrieval



- Given: collection of documents (columns of a matrix $\boldsymbol{A}$) and a query vector $\boldsymbol{q}$.
- Representation: $\boldsymbol{m} \times \boldsymbol{n}$ term by document matrix

- A query $\boldsymbol{q}$ is a (sparse) vector in $\mathbb{R}^m$ (‘pseudo-document’)

Problem: find a column of $\boldsymbol{A}$ that best matches $\boldsymbol{q}$

- *Vector space model:* use $\cos\langle(\boldsymbol{A}(:,\boldsymbol{j}),\boldsymbol{q}),\boldsymbol{j}=\boldsymbol{1}:\boldsymbol{n}$
- Requires the computation of $\boldsymbol{A}^T\boldsymbol{q}$
- Literal Matching $\rightarrow$ ineffective

Common approach: Dimension reduction (SVD)

- LSI: replace A by a low rank approximation [from SVD]

$$A = U\Sigma V^T \quad \rightarrow \quad A_k = U_k \Sigma_k V_k^T$$

- Replace similarity vector: $s = A^T q$ by $s_k = A_k^T q$
- Main issues: 1) computational cost 2) Updates

Idea: Replace A_k by $A\phi(A^T A)$, where ϕ == a filter function

Consider the step-function (Heaviside):

$$\phi(x) = \begin{cases} 0, & 0 \le x \le \sigma_k^2 \\ 1, & \sigma_k^2 \le x \le \sigma_1^2 \end{cases}$$

- Would yield the same result as TSVD but not practical

Use of polynomial filters

- Solution : use a polynomial approximation to ϕ
- Note: $\boxed{s^T = q^T A \phi(A^T A)}$, requires only Mat-Vec's
- Ideal for situations where data must be explored once or a small number of times only –
- Details skipped – see:

E. Kokiopoulou and YS, Polynomial Filtering in Latent Semantic Indexing for Information Retrieval, ACM-SIGIR, 2004.

IR: Use of the Lanczos algorithm (J. Chen, YS '09)

- Lanczos algorithm = Projection method on Krylov subspace Span$\{v, Av, \cdots, A^{m-1}v\}$
- Can get singular vectors with Lanczos, & use them in LSI
- Better: Use the Lanczos vectors directly for the projection
- K. Blom and A. Ruhe [SIMAX, vol. 26, 2005] perform a Lanczos run for each query [expensive].

> - Proposed: One Lanczos run- random initial vector. Then use Lanczos vectors in place of singular vectors.

- In short: Results comparable to those of SVD at a much lower cost.

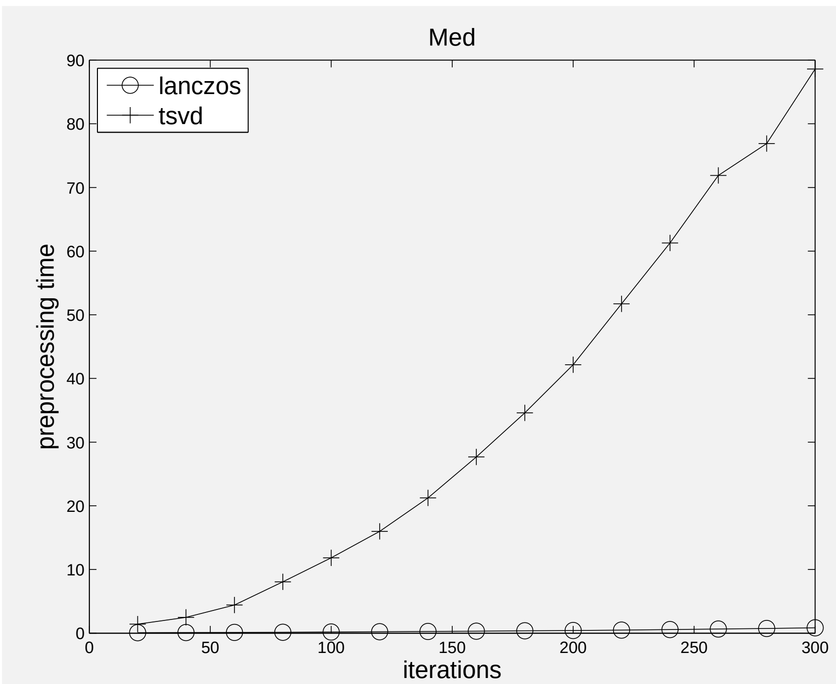
Tests: IR

Information retrieval datasets

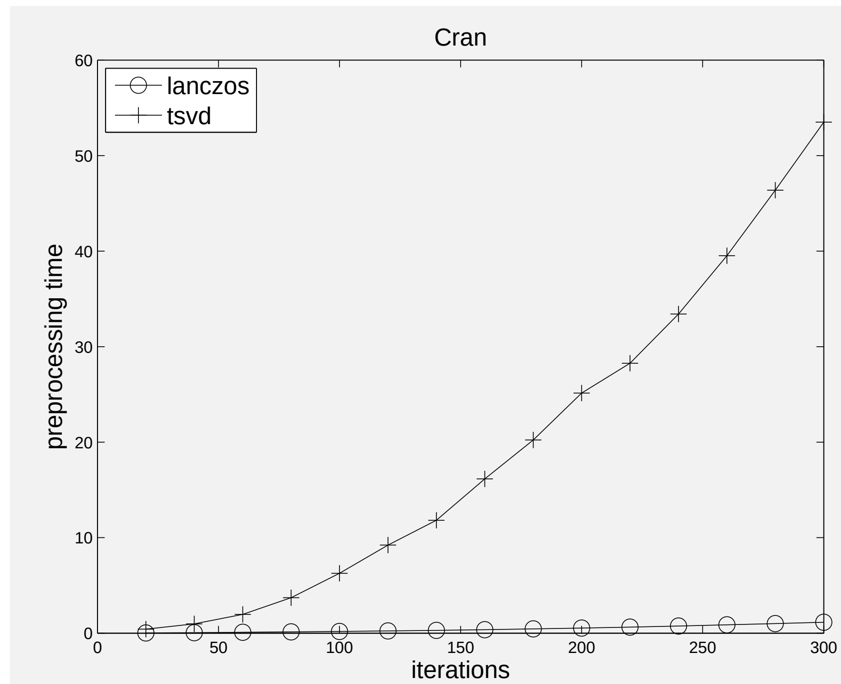
	# Terms	# Docs	# queries	sparsity
MED	7,014	1,033	30	0.735
CRAN	3,763	1,398	225	1.412

Preprocessing times

Med dataset.



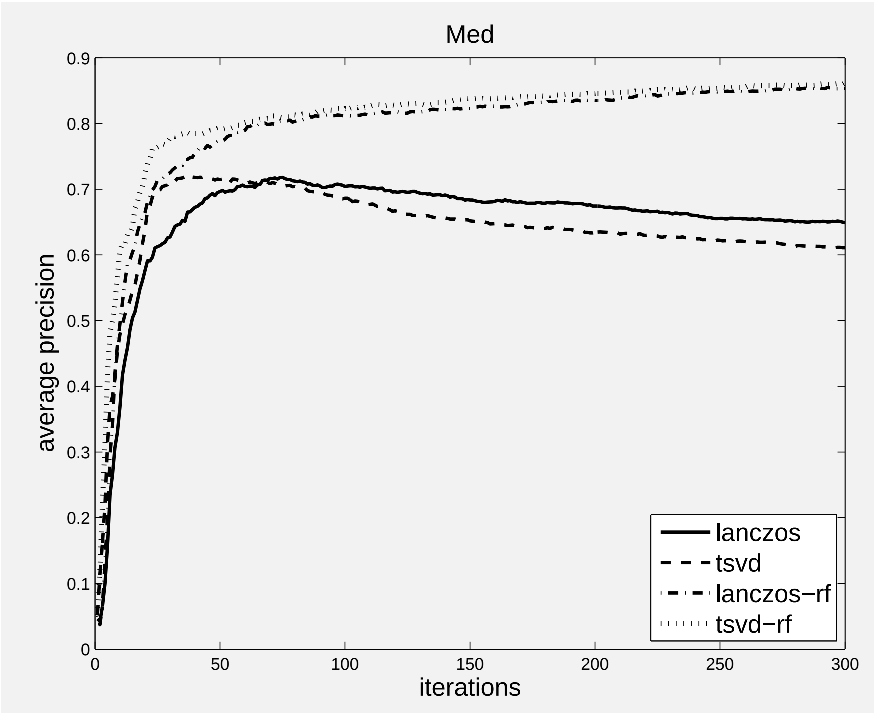
Cran dataset.

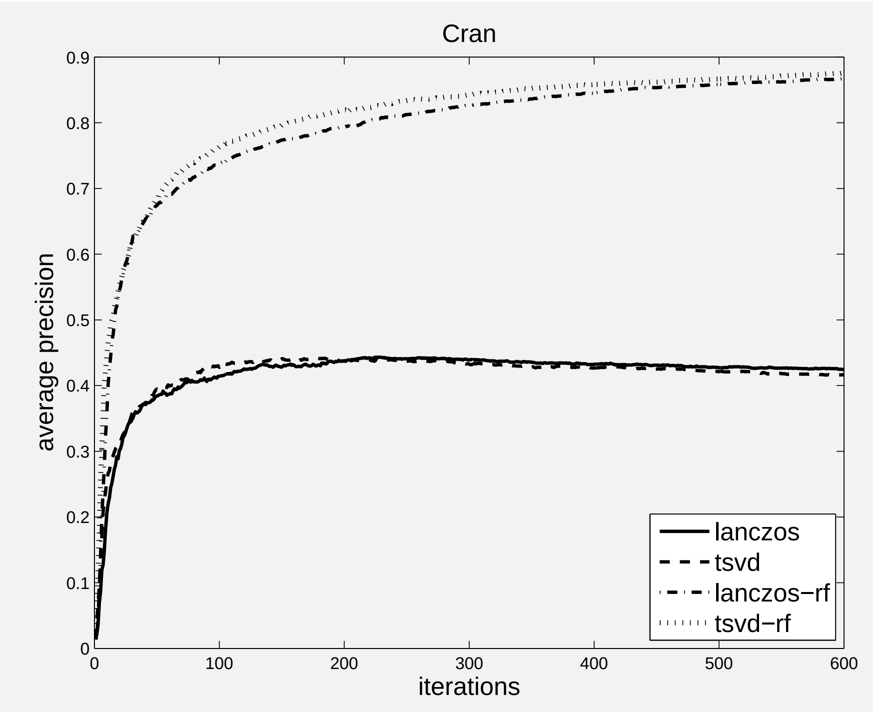


Average retrieval precision

Med dataset

Cran dataset

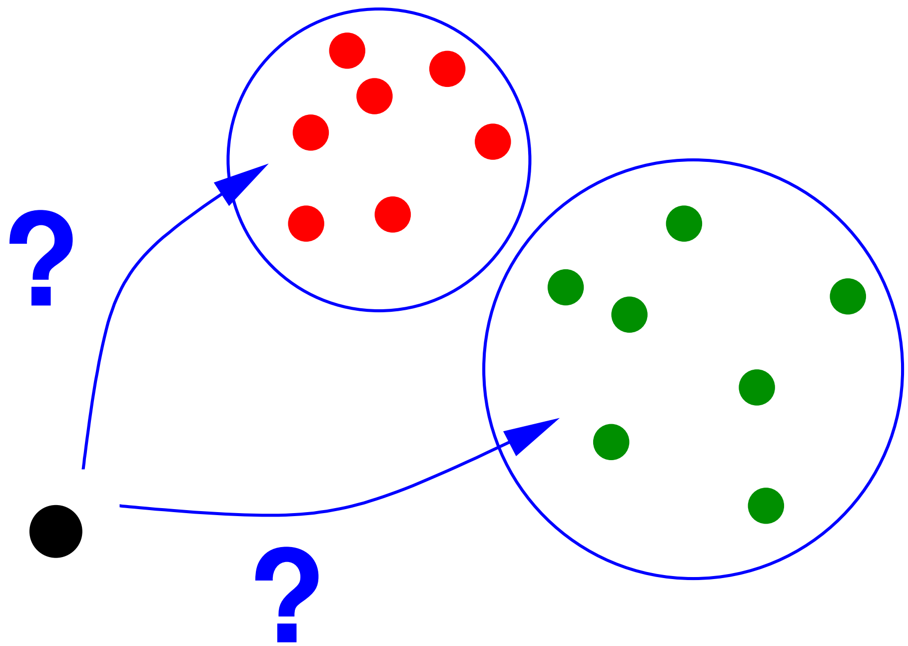




Retrieval precision comparisons

Supervised learning: classification

Problem: Given labels (say “A” and “B”) for each item of a given set, find a mechanism to classify an unlabelled item into either the “A” or the “B" class.

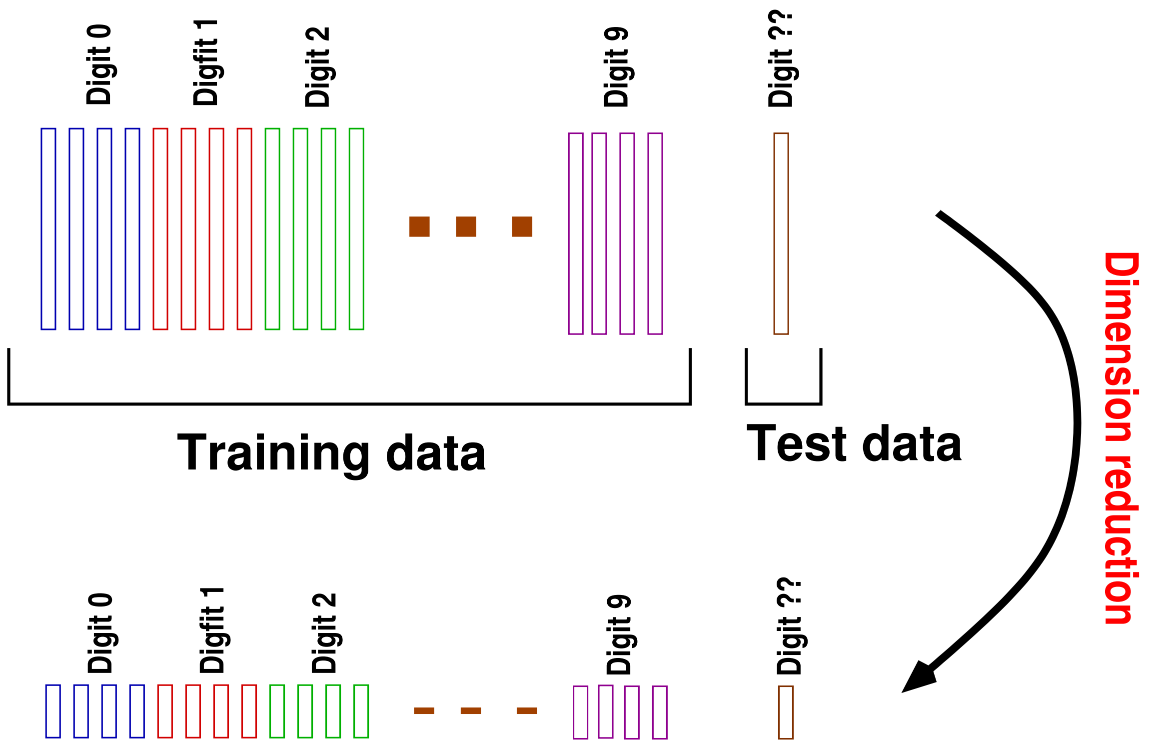


- Many applications.
- Example: distinguish SPAM and non-SPAM messages
- Can be extended to more than 2 classes.

Supervised learning: classification

- Best illustration: written digits recognition example

Given: a set of labeled samples (training set), and an (unlabeled) test image.
Problem: find label of test image

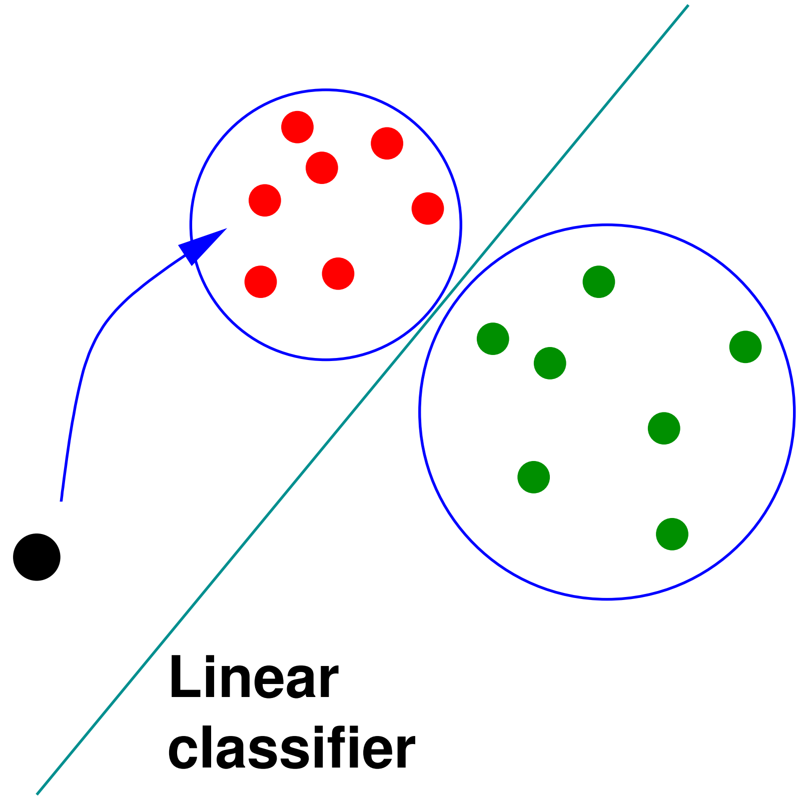


- Roughly speaking: we seek dimension reduction so that recognition is 'more effective' in low-dim. space

Supervised learning: Linear classification

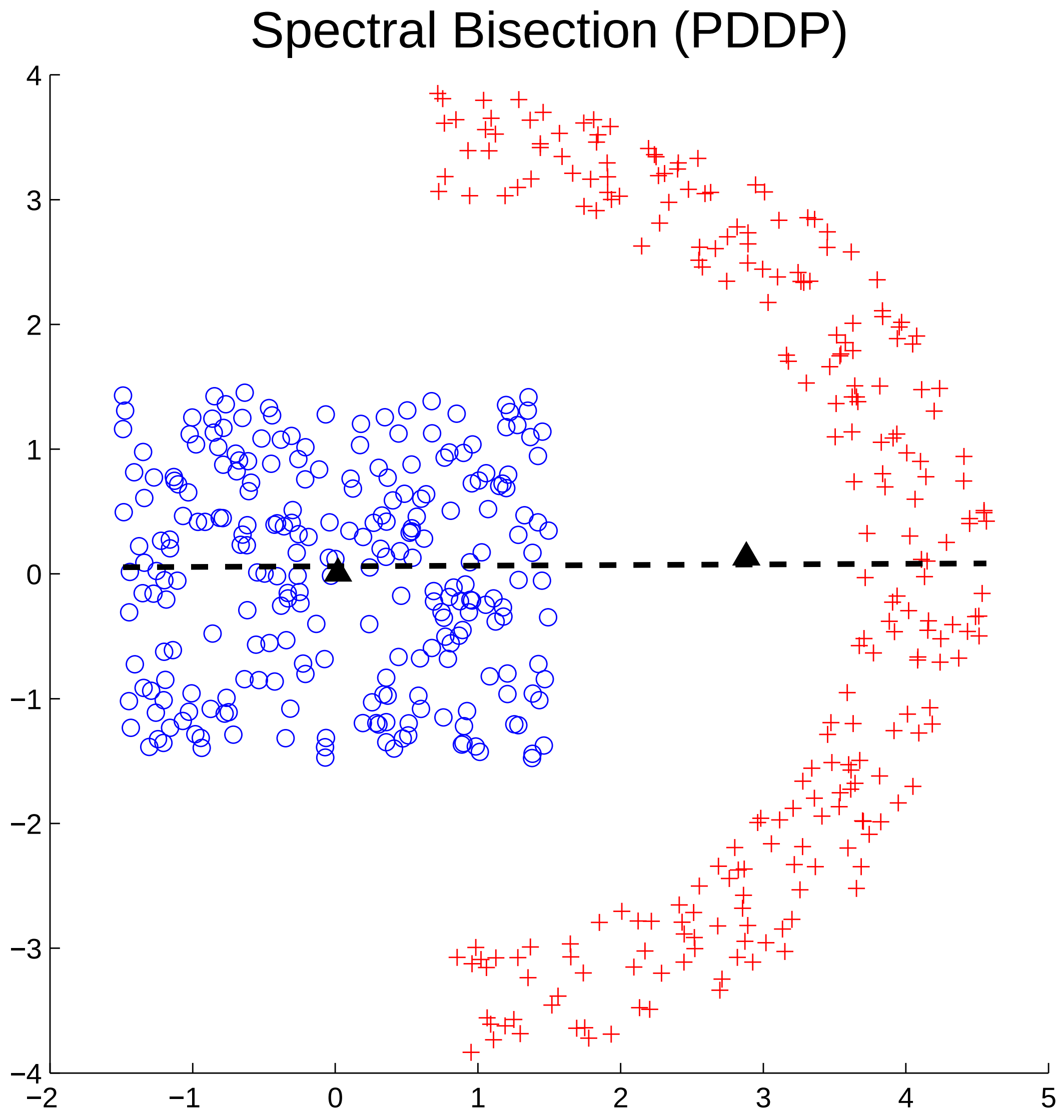
Linear classifiers: Find a hyperplane which best separates the data in classes A and B.

➤ Example of application: Distinguish between SPAM and non-SPAM e-mails

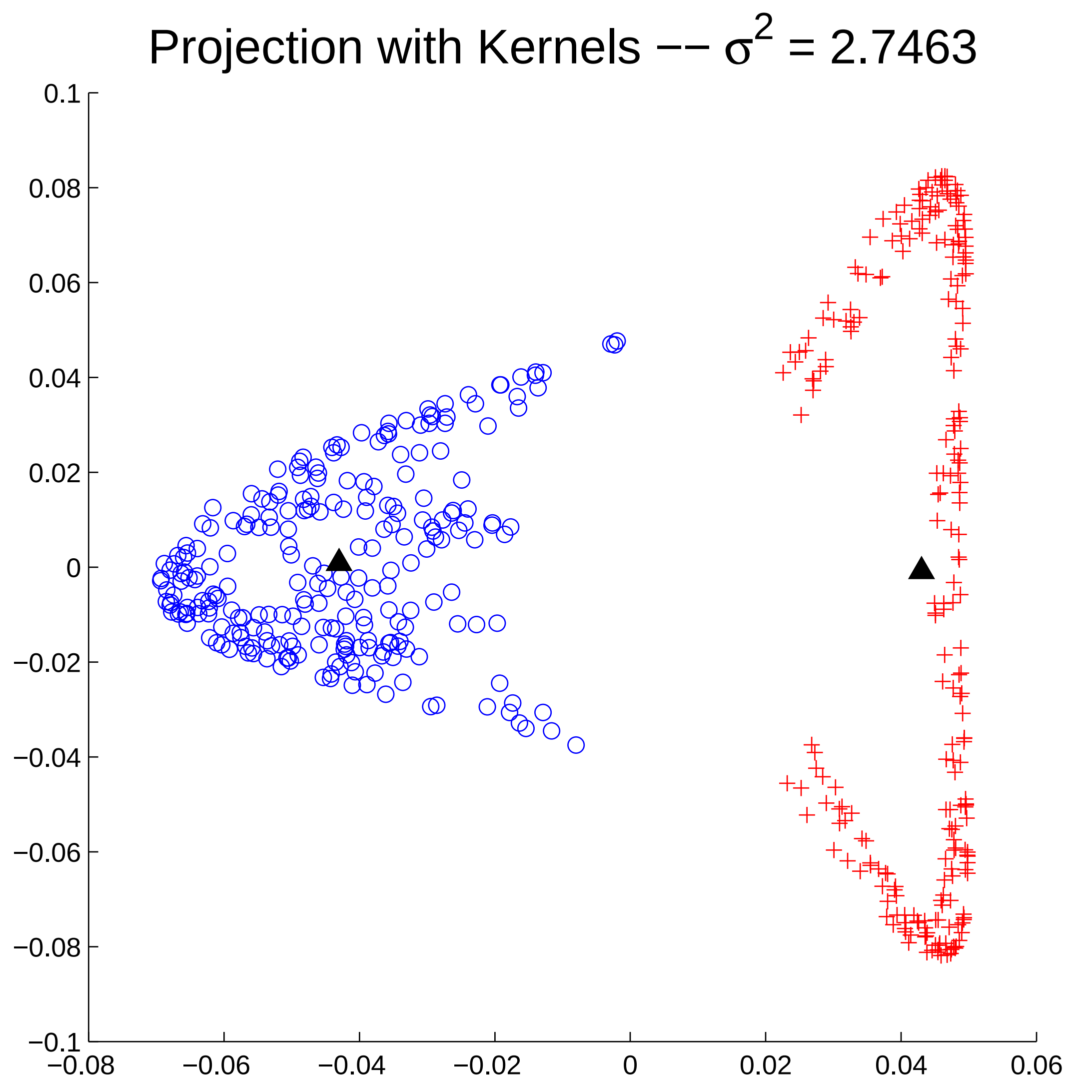


➤ Note: The world in non-linear. Often this is combined with Kernels – amounts to changing the inner product

A harder case:



- Use kernels to transform



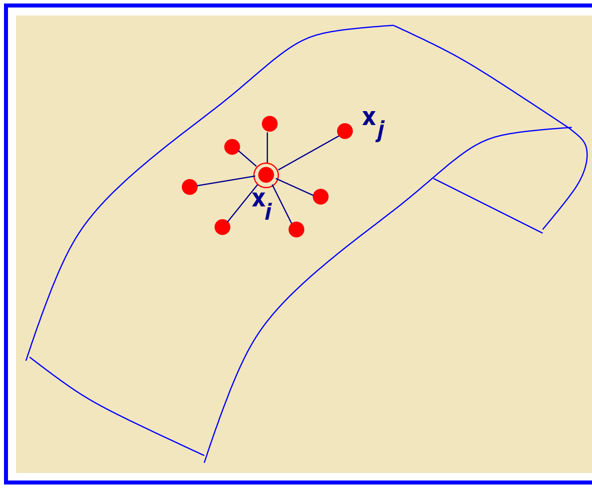
Transformed data with a Gaussian Kernel

GRAPH-BASED TECHNIQUES

Graph-based methods

- Start with a graph of data. e.g.: graph of k nearest neighbors (k-NN graph)

Want: Perform a projection which preserves the graph in some sense



- Define a *graph Laplacean:*

$$L = D - W$$

e.g.,: $w_{ij} = \begin{cases} 1 & \text{if } j \in Adj(i) \\ 0 & \text{else} \end{cases}$ $\quad D = \text{diag}\left[d_{ii} = \sum_{j \neq i} w_{ij} \right]$

with $Adj(i)$ = neighborhood of i (excluding i)

A side note: Graph partitioning

If x is a vector of signs (± 1) then

$$x^\top L x = 4 \times \text{('number of edge cuts')}$$

edge-cut = pair (i, j) with $x_i \neq x_j$

➤ Consequence: Can be used for partitioning graphs, or ‘clustering’ [take $p = sign(u_2)$, where u_2 = 2nd smallest eigenvector..]

Example: The Laplacean eigenmaps approach

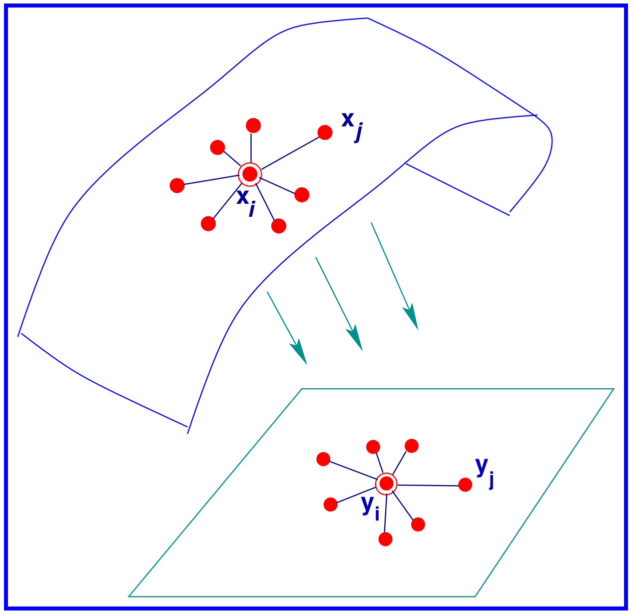
Laplacean Eigenmaps [Belkin-Niyogi '01] *minimizes*

$$\mathcal{F}(Y) = \sum_{i,j=1}^{n} w_{ij} \|y_i - y_j\|^2 \quad \text{subject to} \quad Y D Y^\top = I$$

Motivation: if $\|x_i - x_j\|$ is small (orig. data), we want $\|y_i - y_j\|$ to be also small (low-Dim. data)

➤ Original data used indirectly through its graph

➤ Leads to $n \times n$ sparse eigenvalue problem [In 'sample' space]



- Problem translates to:

$$\min_{\begin{cases} Y \in \mathbb{R}^{d\times n} \\ Y D Y^\top = I \end{cases}} \text{Tr}\left[Y(D-W)Y^\top\right].$$

- Solution (sort eigenvalues increasingly):

$$(D-W)u_i = \lambda_i D u_i\,; \quad y_i = u_i^\top; \quad i = 1,\cdots,d$$

- Note: can assume $D = I$. Amounts to rescaling data. Problem becomes

$$(I-W)u_i = \lambda_i u_i\,; \quad y_i = u_i^\top; \quad i = 1,\cdots,d$$

Locally Linear Embedding (Roweis-Saul-00)

➤ LLE is very similar to Eigenmaps. Main differences:

1) Graph Laplacean matrix is replaced by an 'affinity' graph

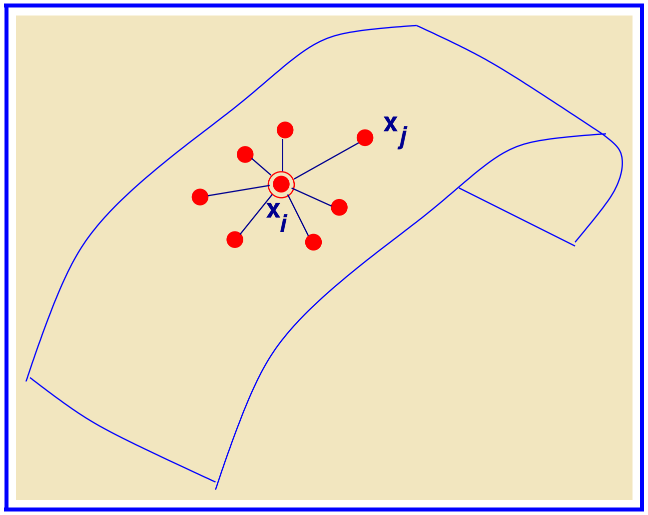
2) Objective function is changed.

1. Graph: Each x_i is written as a convex combination of its k nearest neighbors:

$$x_i \approx \Sigma w_{ij} x_j, \quad \sum_{j \in N_i} w_{ij} = 1$$

➤ Optimal weights computed ('local calculation') by minimizing

$$\|x_i - \Sigma w_{ij} x_j\| \quad \text{for} \quad i = 1, \cdots, n$$



2. Mapping:

The y_i's should obey the same 'affinity' as x_i's $\rightsquigarrow$

Minimize:

$$\sum_i \left\| y_i - \sum_j w_{ij} y_j \right\|^2 \quad \text{subject to:} \quad Y\mathbf{1} = 0, \quad YY^\top = I$$

Solution:

$$\boxed{(I - W^\top)(I - W)u_i = \lambda_i u_i; \qquad y_i = u_i^\top .}$$

➤ $(I - W^\top)(I - W)$ replaces the graph Laplacean of eigenmaps

ONPP (Kokiopoulou and YS '05)

- Orthogonal Neighborhood Preserving Projections
- A linear (orthogonoal) version of LLE obtained by writing $\boldsymbol{Y}$ in the form $\boldsymbol{Y} = \boldsymbol{V}^\top \boldsymbol{X}$
- Same graph as LLE. Objective: preserve the affinity graph (as in LEE) *but* with the constraint $\boldsymbol{Y} = \boldsymbol{V}^\top \boldsymbol{X}$
- Problem solved to obtain mapping:

$$\min_{\boldsymbol{V}} \text{Tr}\left[\boldsymbol{V}^\top \boldsymbol{X}(\boldsymbol{I} - \boldsymbol{W}^\top)(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{X}^\top \boldsymbol{V}\right]$$

s.t. $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$

- In LLE replace $\boldsymbol{V}^\top \boldsymbol{X}$ by $\boldsymbol{Y}$

Implicit vs explicit mappings

- In PCA the mapping Φ from high-dimensional space ($\mathbb{R}^m$) to low-dimensional space ($\mathbb{R}^d$) is explicitly known:

$$\boldsymbol{y} = \Phi(\boldsymbol{x}) \equiv \boldsymbol{V}^T \boldsymbol{x}$$

- In Eigenmaps and LLE we only know

$$\boldsymbol{y_i} = \phi(\boldsymbol{x_i}), \boldsymbol{i} = 1, \cdots, \boldsymbol{n}$$

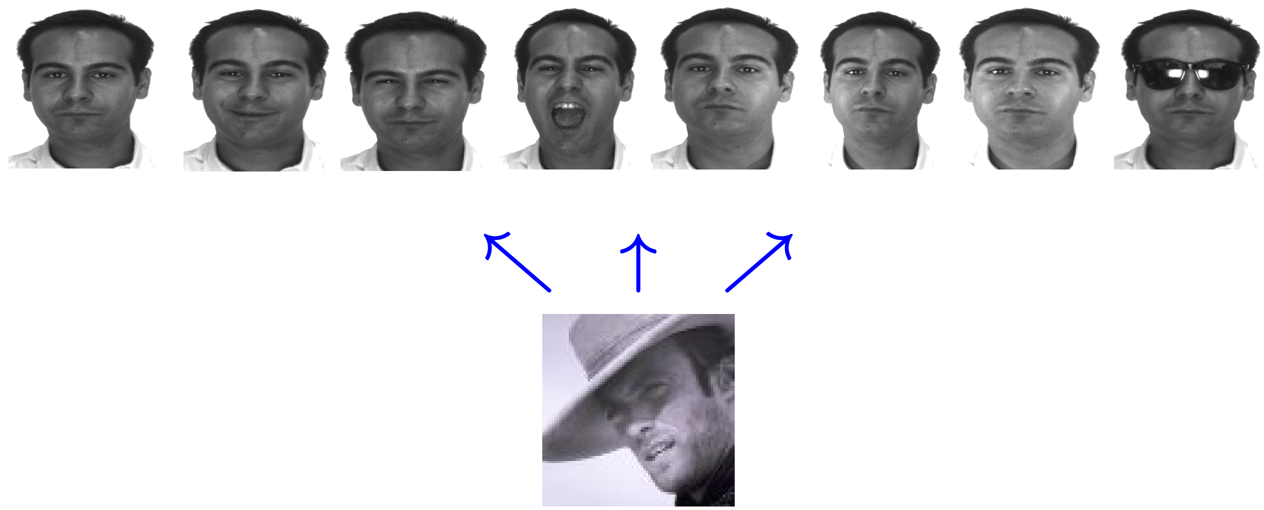
- Mapping ϕ is complex, i.e.,
- Difficult to get $\phi(x)$ for an arbitrary x not in the sample.
- Inconvenient for classification
- “The out-of-sample extension” problem

Face Recognition – background

Problem: We are given a database of images: [arrays of pixel values]. And a test (new) image.

Face Recognition – background

Problem: We are given a database of images: [arrays of pixel values]. And a test (new) image.



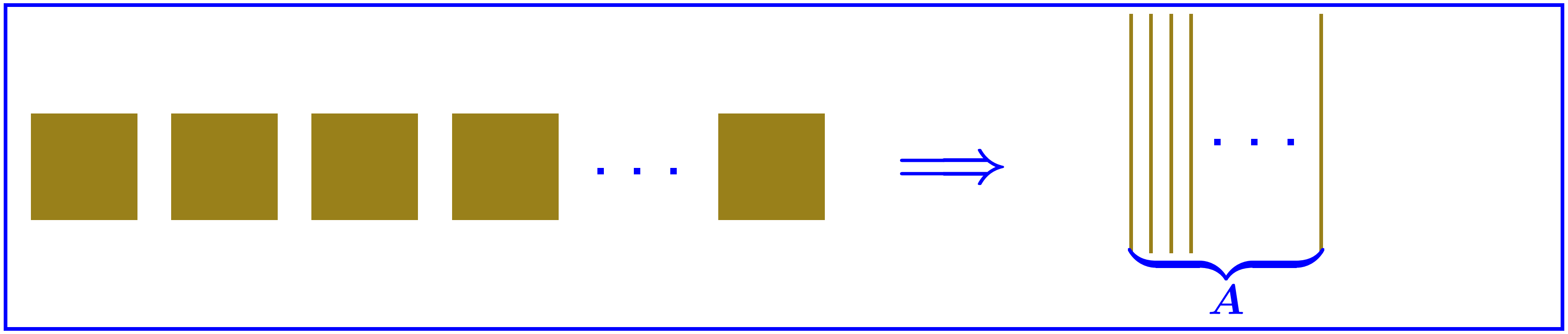
Question: Does this new image correspond to one of those in the database?

Difficulty Positions, Expressions, Lighting, ...,

Example: Eigenfaces [Turk-Pentland, '91]

➤ Idea identical with the one we saw for digits:

– Consider each picture as a (1-D) column of all pixels
– Put together into an array A of size $\#\_pixels \times \#\_images$.



– Do an SVD of A and perform comparison with any test image in low-dim. space

Graph-based methods in a supervised setting

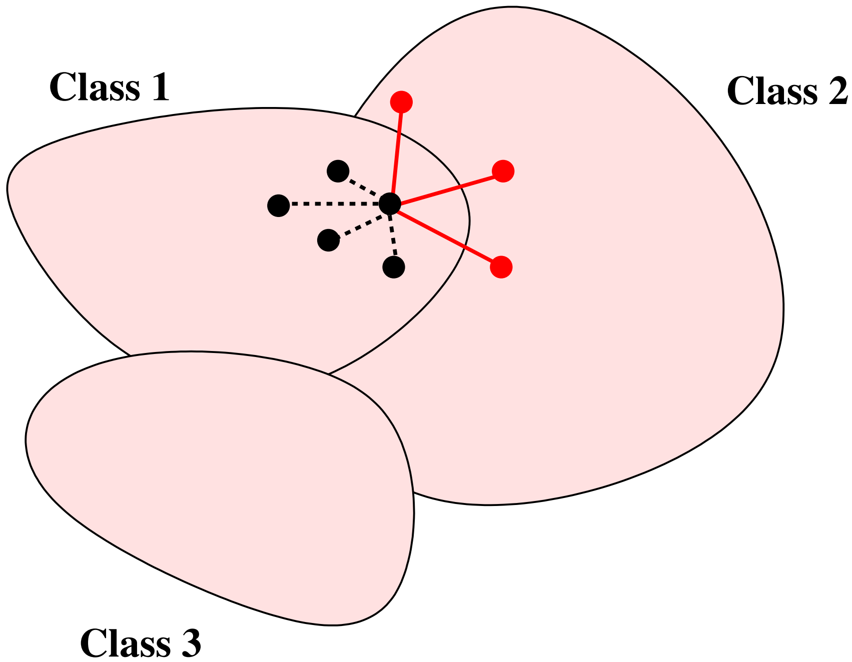
Graph-based methods can be adapted to supervised mode. Idea: Build G so that nodes in the same class are neighbors. If c = # classes, G consists of c cliques.

- Weight matrix W =block-diagonal
- Note: $\text{rank}(W) = n - c$.
- As before, graph Laplacean:

$$L_c = D - W$$

$$W = \begin{pmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_c \end{pmatrix}$$

- Can be used for ONPP and other graph based methods
- Improvement: add repulsion Laplacean [Kokiopoulou, YS 09]



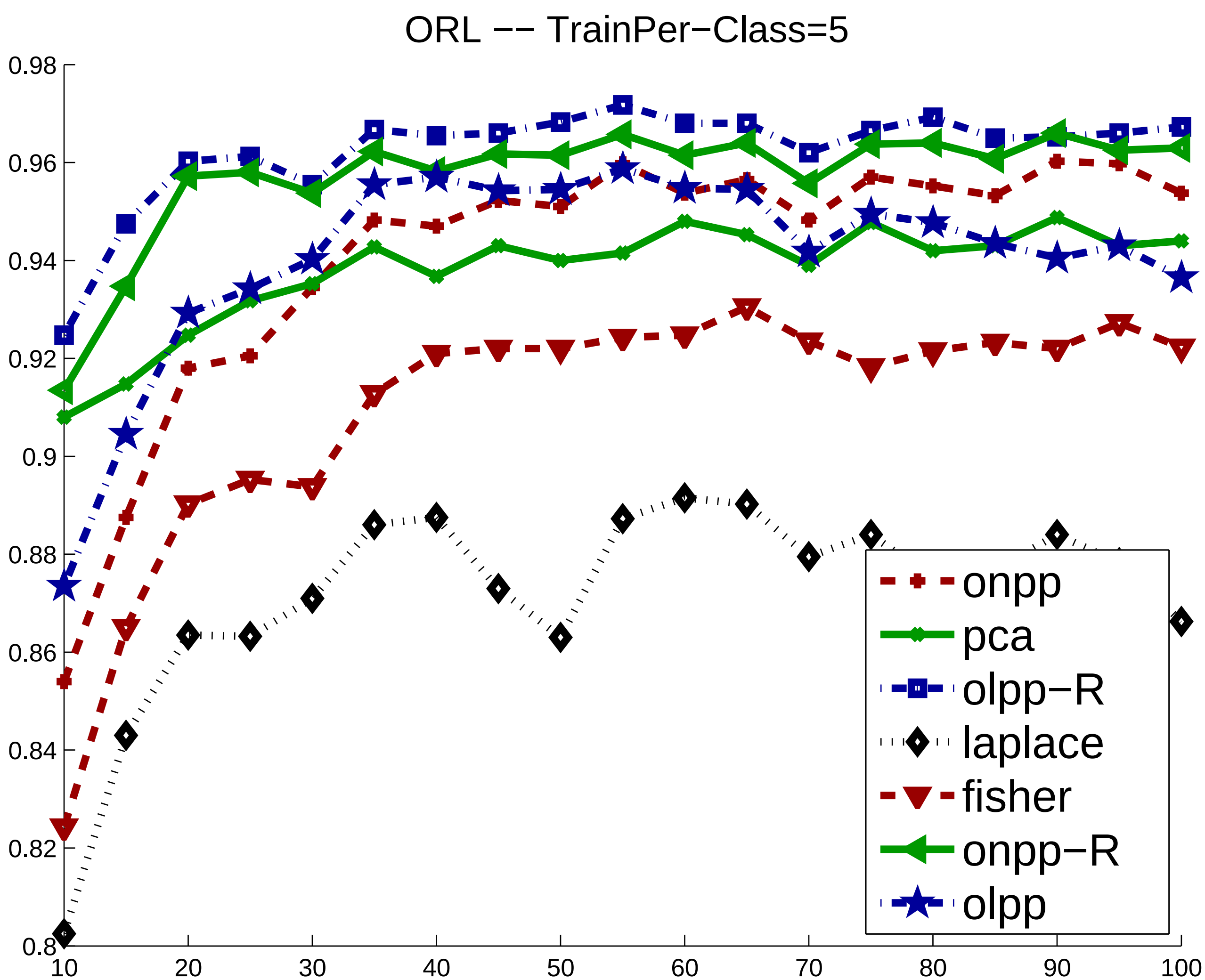
Leads to eigenvalue problem with matrix:

$$L_c - \rho L_R$$

- L_c = class-Laplacean,
- L_R = repulsion Laplacean,
- $\rho =$ parameter

Test: ORL 40 subjects, 10 sample images each – example:

# of pixels : 112×92; TOT. # images : 400



➤ Observation: some values of ρ yield better results than using the optimum ρ obtained from maximizing trace ratio

Conclusion

- Interesting new matrix problems in areas that involve the effective mining of data
- Among the most pressing issues is that of reducing computational cost - [SVD, SDP, ..., too costly]
- Many online resources available
- Huge potential in areas like materials science though inertia has to be overcome
- To a researcher in computational linear algebra : big tide of change on types or problems, algorithms, frameworks, culture,..
- But change should be welcome

When one door closes, another opens; but we often look so long and so regretfully upon the closed door that we do not see the one which has opened for us.

Alexander Graham Bell (1847-1922)

➤ In the words of “Who Moved My Cheese?” [Spencer Johnson, 2002]:

“If you do not change, you can become extinct !”