



# Linear algebra methods for data mining

*Yousef Saad*

*Department of Computer Science  
and Engineering*

*University of Minnesota*

*NCSU*

*Jan 31, 2014*

## *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

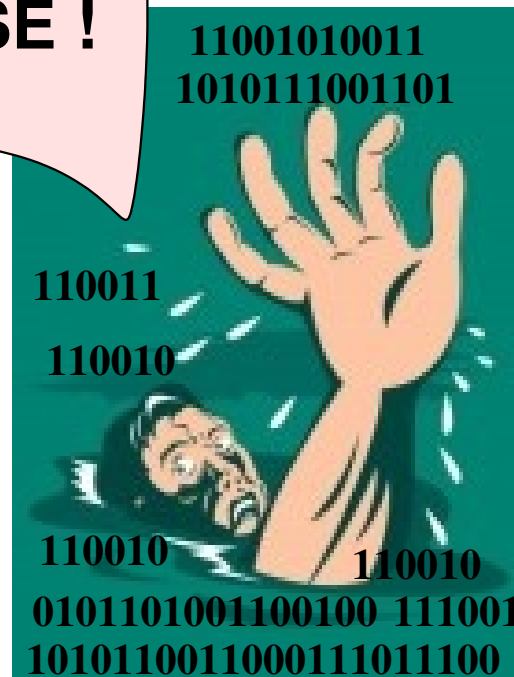
## *Introduction: What is data mining?*

Set of methods and tools to extract meaningful information or patterns from data. Broad area : data analysis, machine learning, pattern recognition, information retrieval, ...

- Tools used: linear algebra; Statistics; Graph theory; Approximation theory; Optimization; ...
- This talk: brief introduction – with emphasis on linear algebra viewpoint
- + our initial work on materials.
- Emphasis on “Dimension reduction methods”

# Drowning in data

**Dimension  
Reduction  
PLEASE !**



Picture modified from [http://www.123rf.com/photo\\_7238007\\_man-drowning-reaching-out-for-help.html](http://www.123rf.com/photo_7238007_man-drowning-reaching-out-for-help.html)

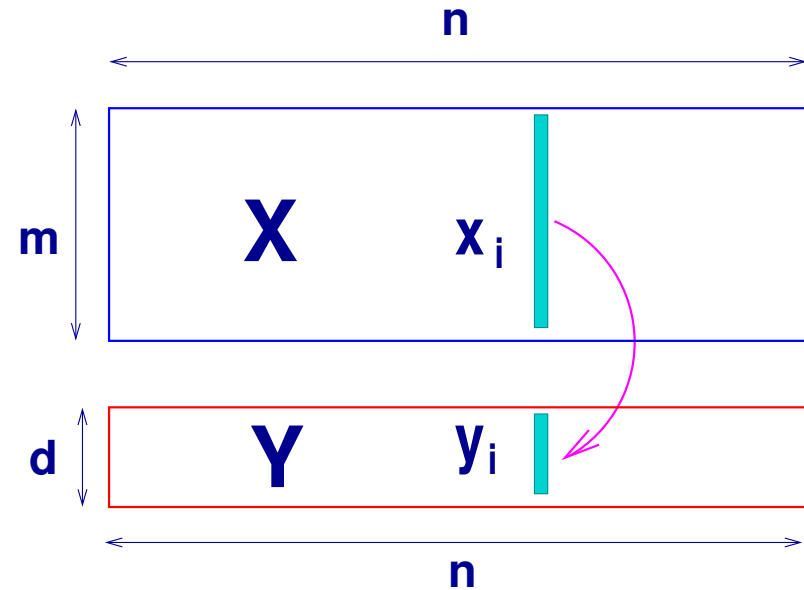
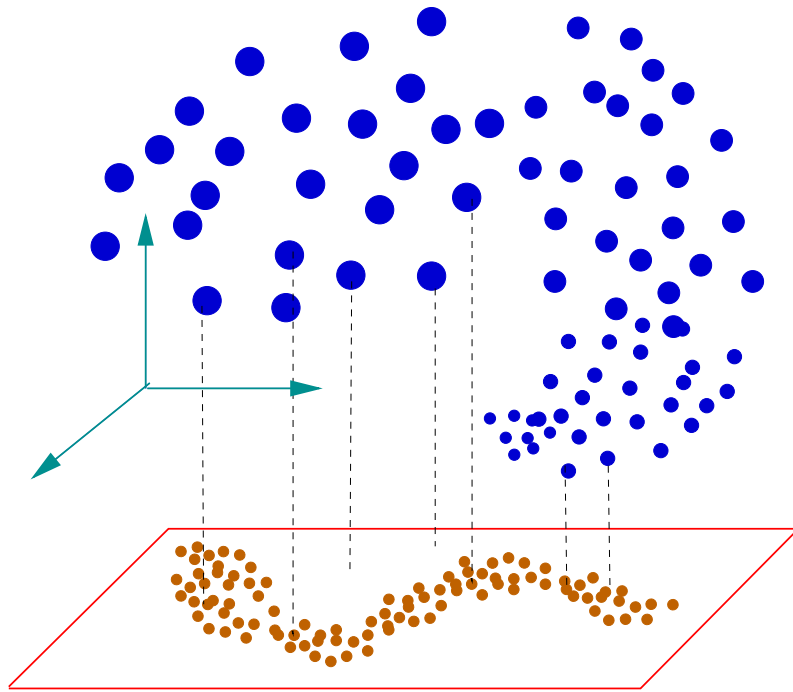
## Major tool of Data Mining: Dimension reduction

- 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 'parameters'

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

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

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



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

## Example: Principal Component Analysis (PCA)

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

$$\begin{array}{l} \max \\ W \in \mathbb{R}^{m \times d} \\ W^\top W = I \end{array} \sum_{i=1}^d \left\| y_i - \frac{1}{n} \sum_{j=1}^n y_j \right\|_2^2, \quad y_i = W^\top x_i.$$

➤ Leads to maximizing

$$\text{Tr} [W^\top (X - \mu e^\top)(X - \mu e^\top)^\top W], \quad \mu = \frac{1}{n} \sum_{i=1}^n x_i$$

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

**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’ ..

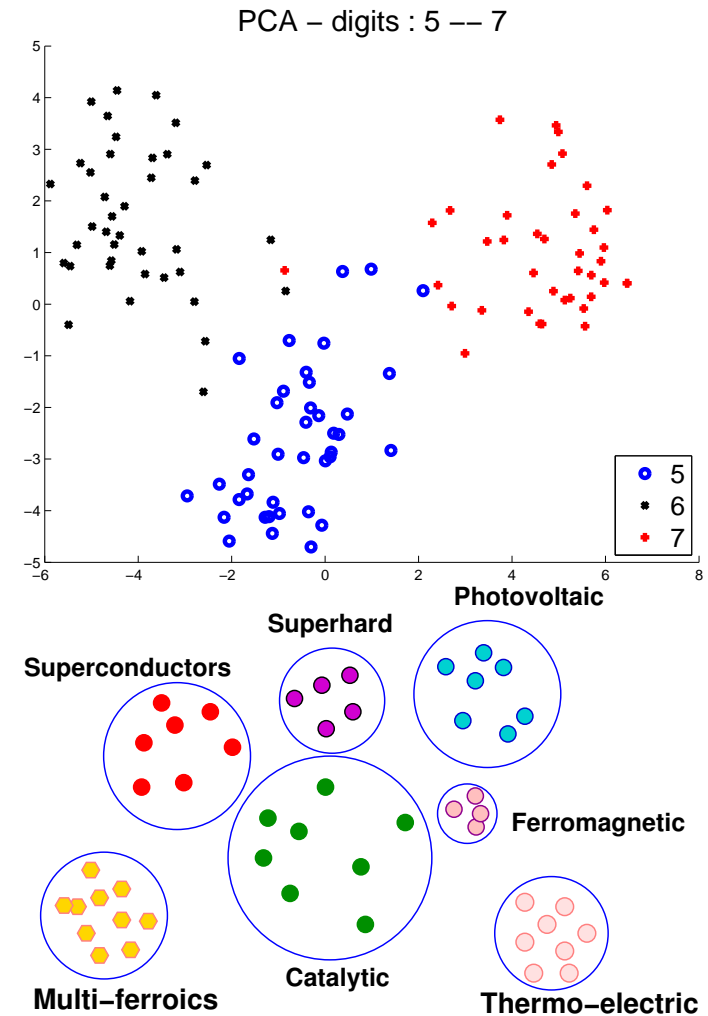
$$\sum_i \|x_i - WW^\top x_i\|^2 = \sum_i \|x_i - Wy_i\|^2$$

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



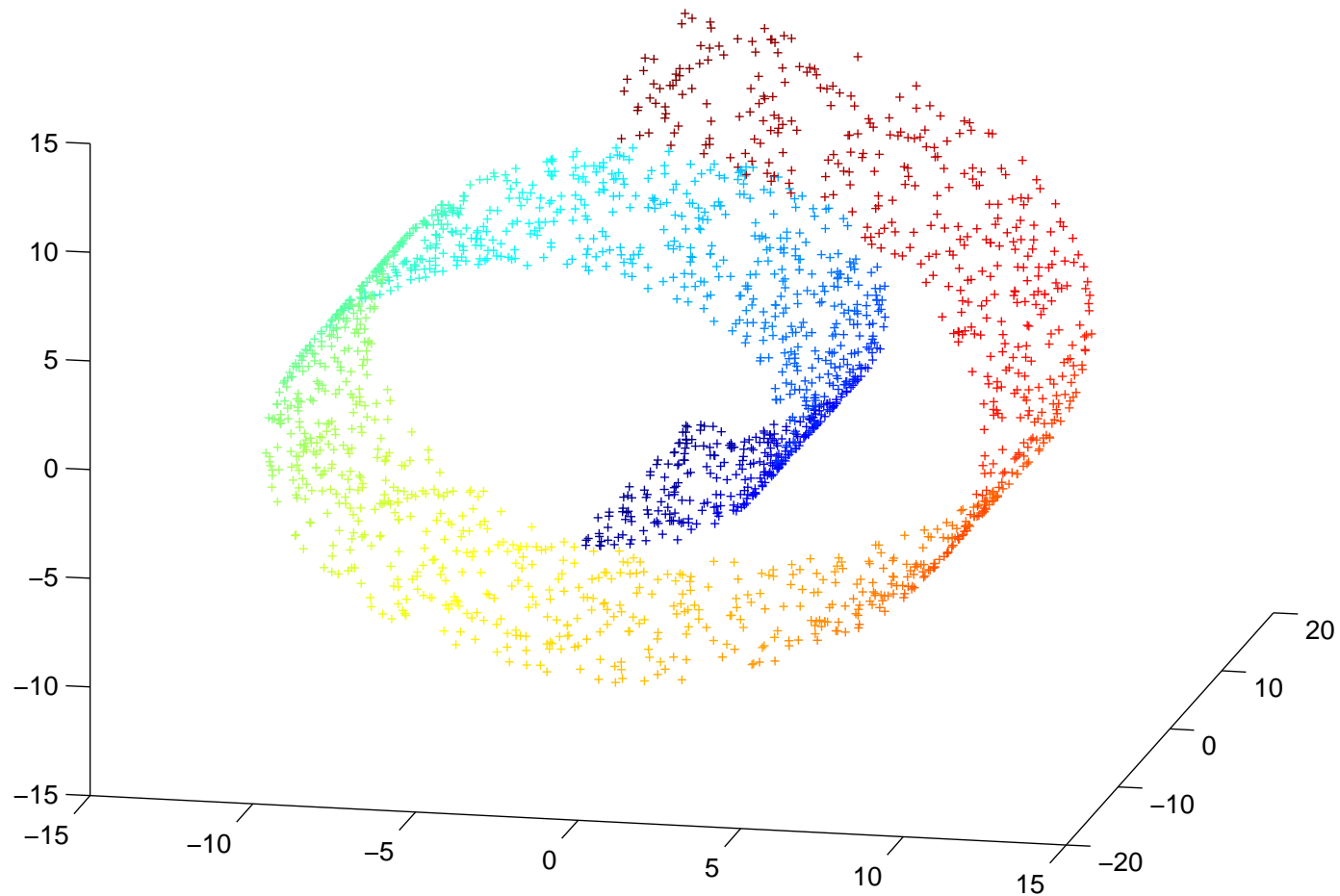
# Unsupervised learning

- “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 clustering technique: K-means

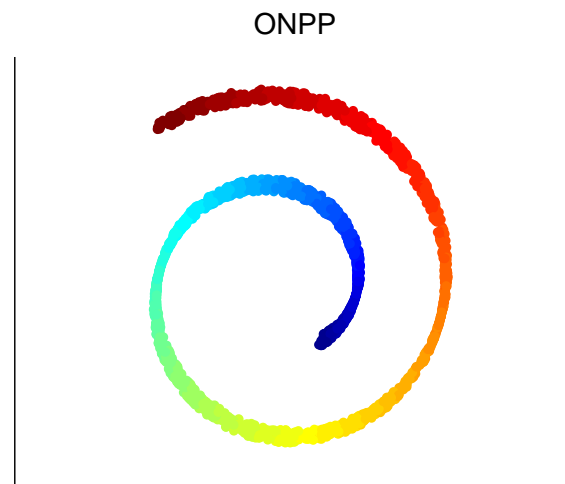
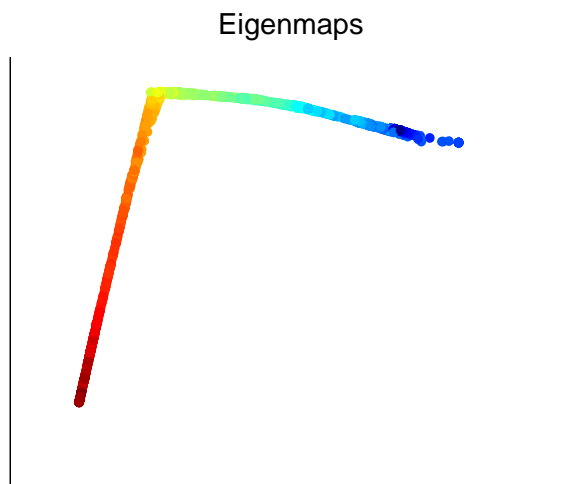
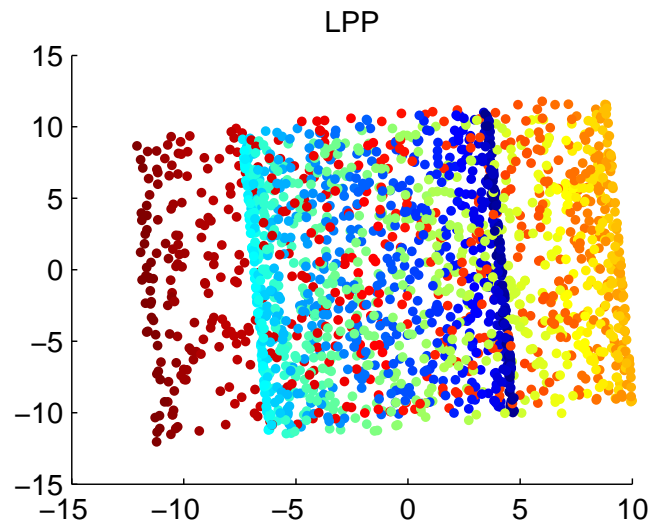
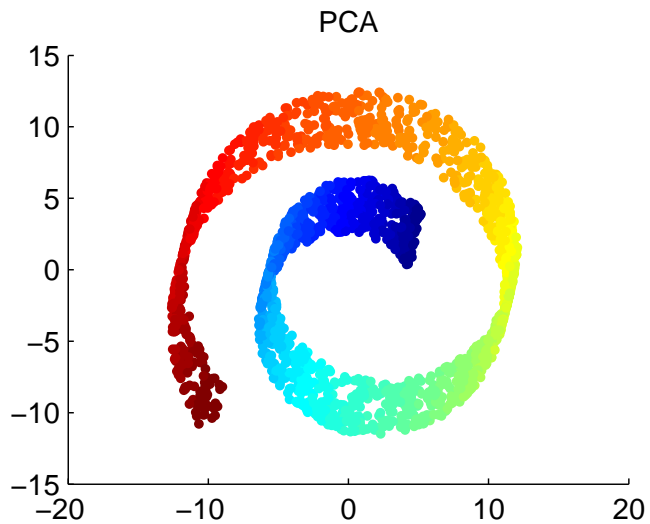


# Example 1: The 'Swirl-Roll' (2000 points in 3-D)

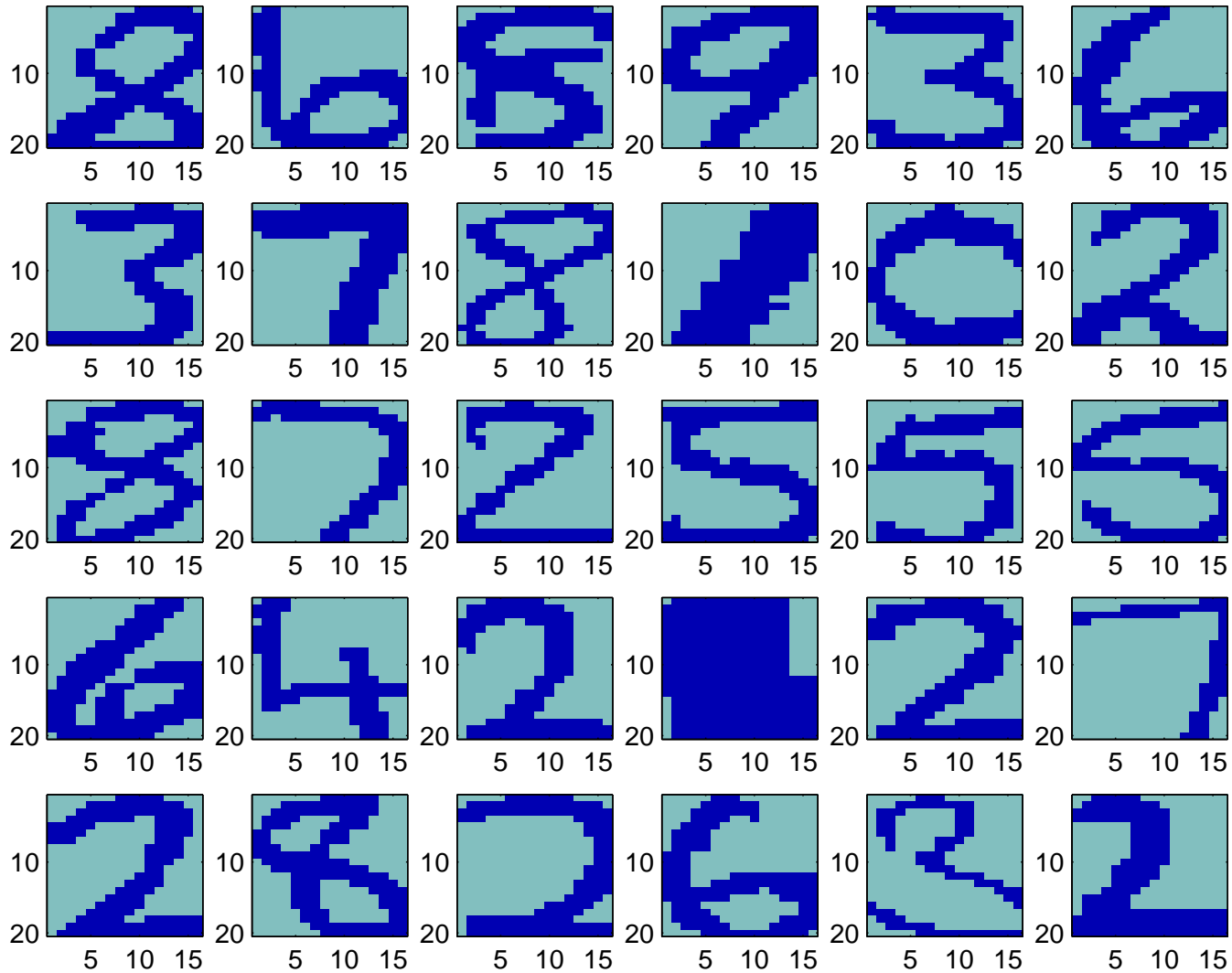
Original Data in 3-D



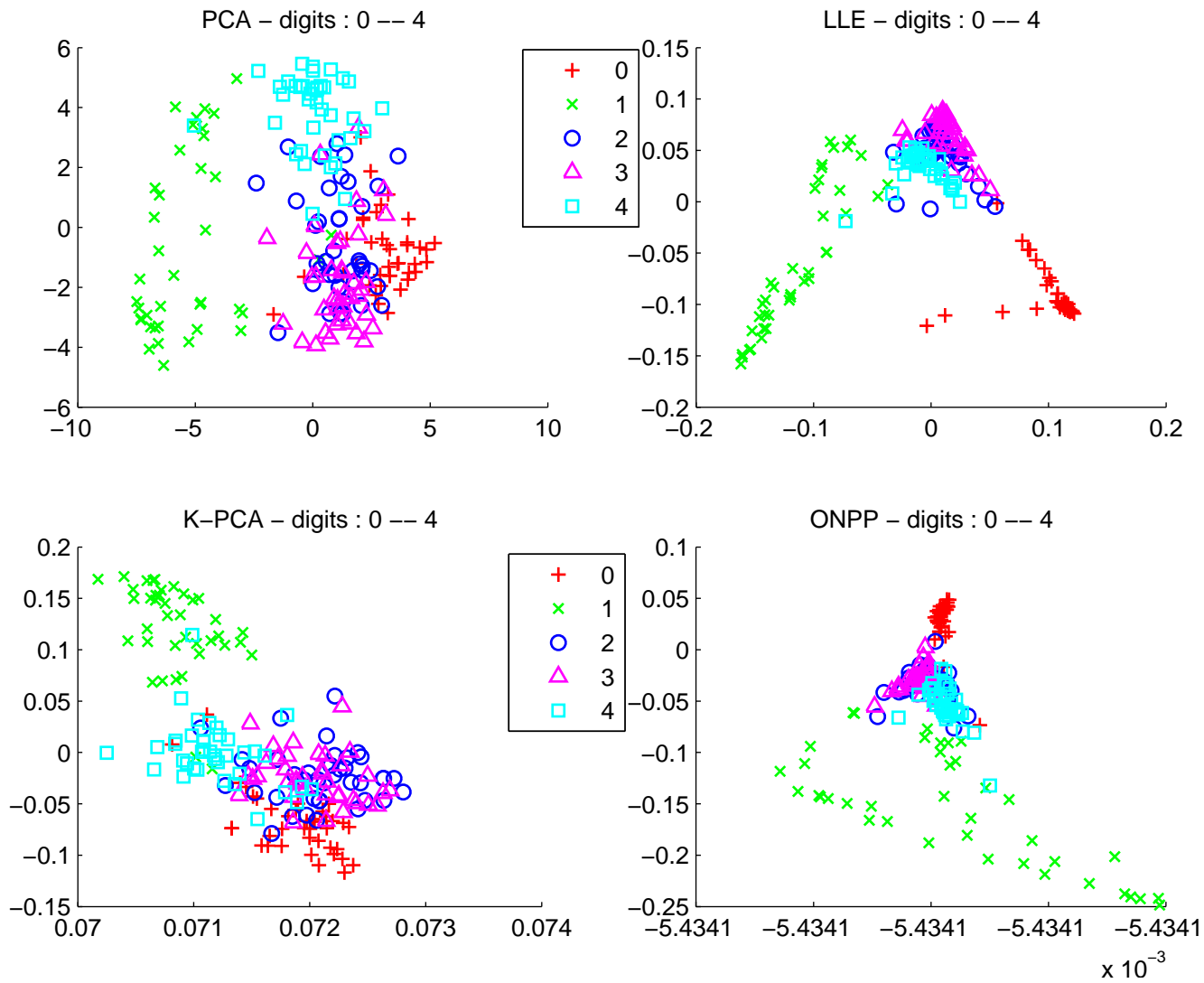
# 2-D 'reductions':



## Example 2: Digit images (a random sample of 30)

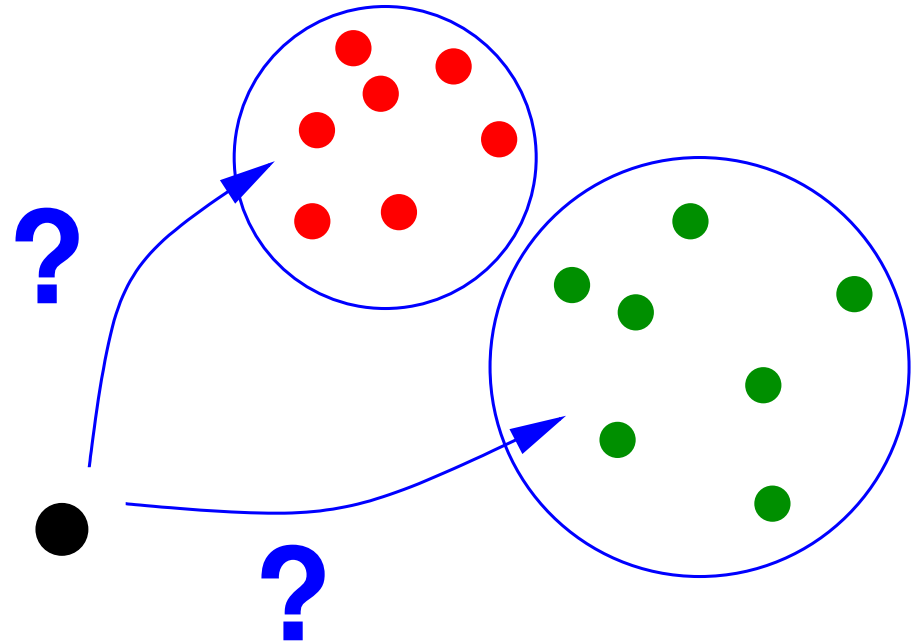


# 2-D 'reductions':



## *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.



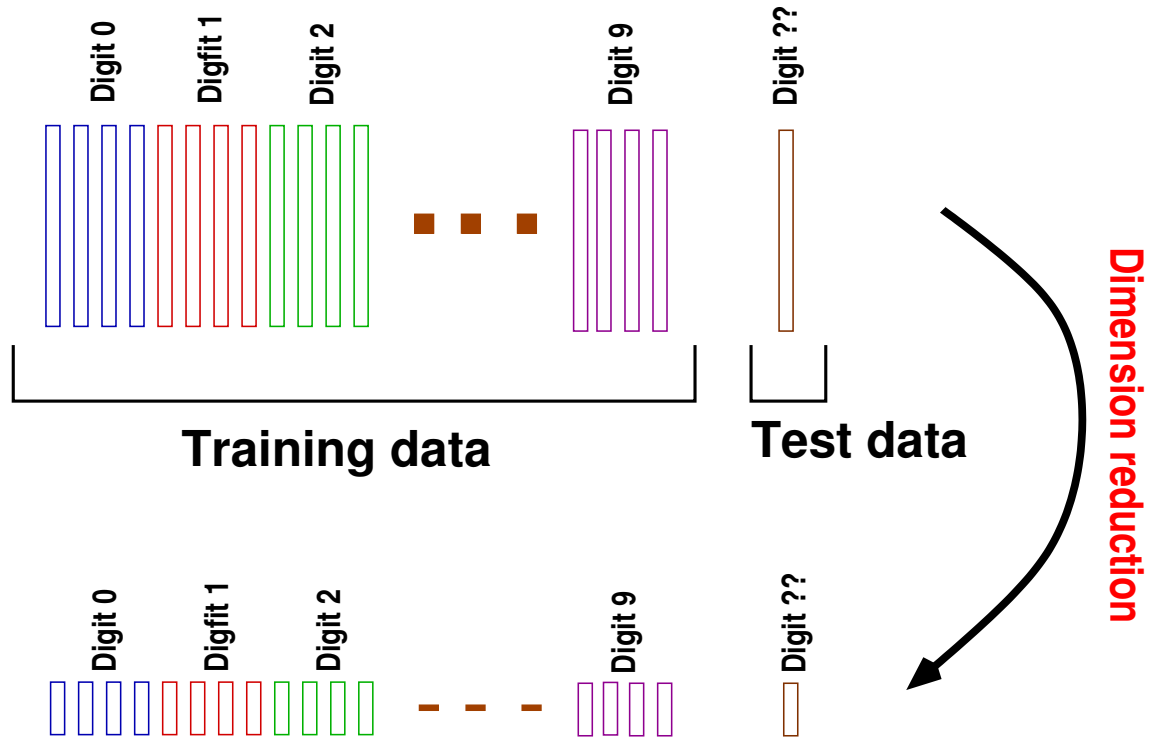
- Many applications.
- Example: distinguish SPAM and non-SPAM messages
- Can be extended to more than 2 classes.

# Supervised learning: classification

- Best illustration: written digits recognition example

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

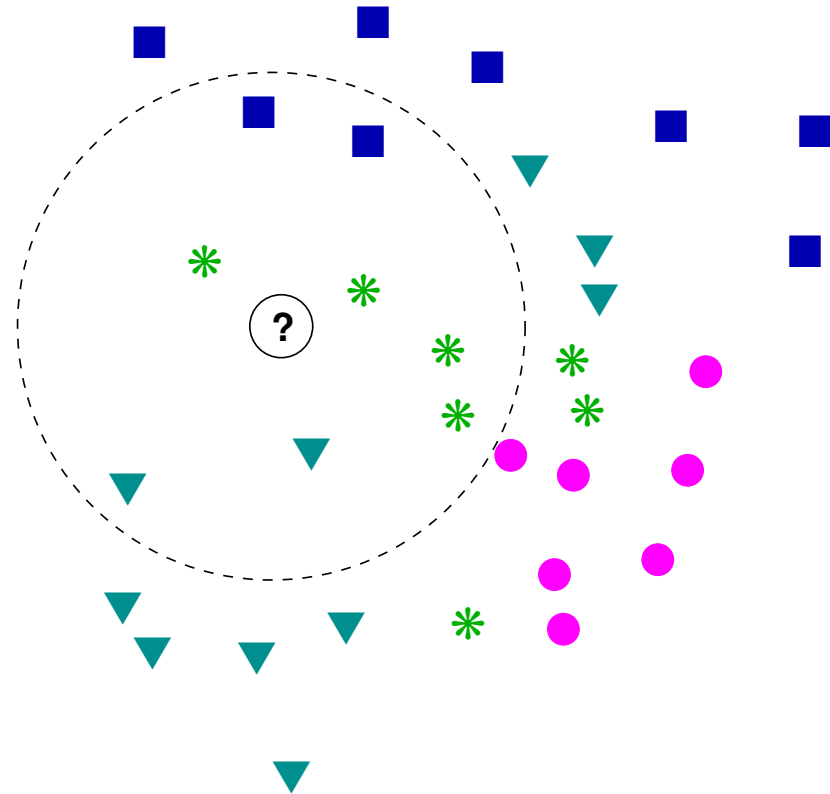
**Problem:** find label of test image



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

## Basic method: *K*-nearest neighbors (*KNN*) classification

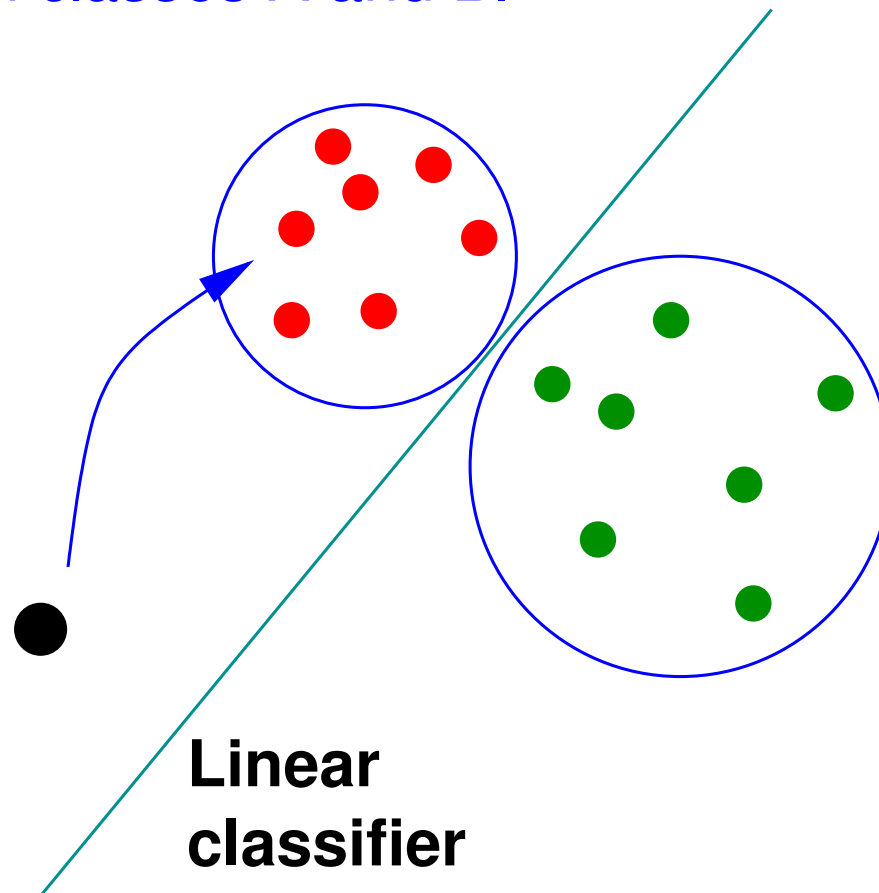
- Idea of a voting system: get distances between test sample and training samples
- Get the  $k$  nearest neighbors (here  $k = 8$ )
- Predominant class among these  $k$  items is assigned to the test sample (“\*” here)





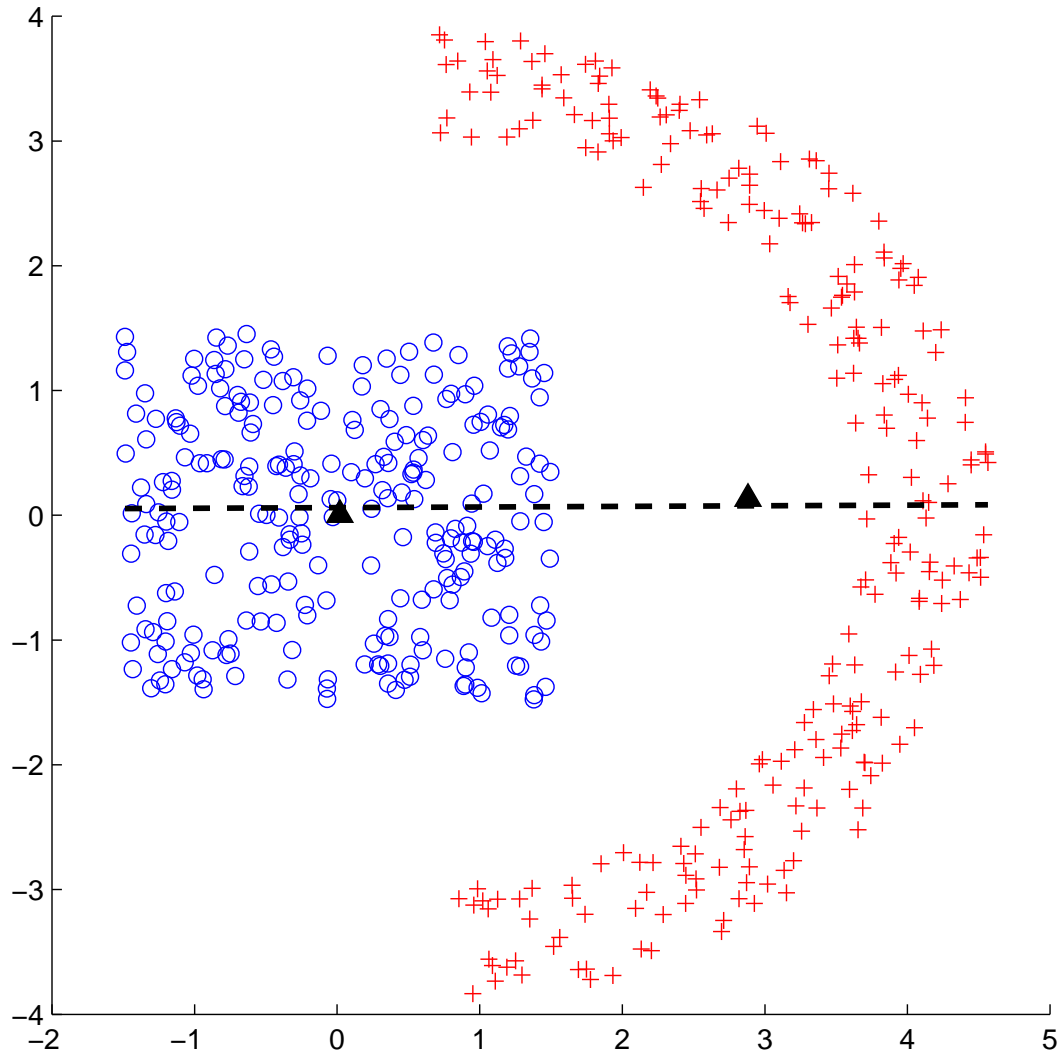
## *Linear classifiers and Fisher's LDA*

- Idea for two classes: Find a hyperplane which best separates the data in classes A and B.



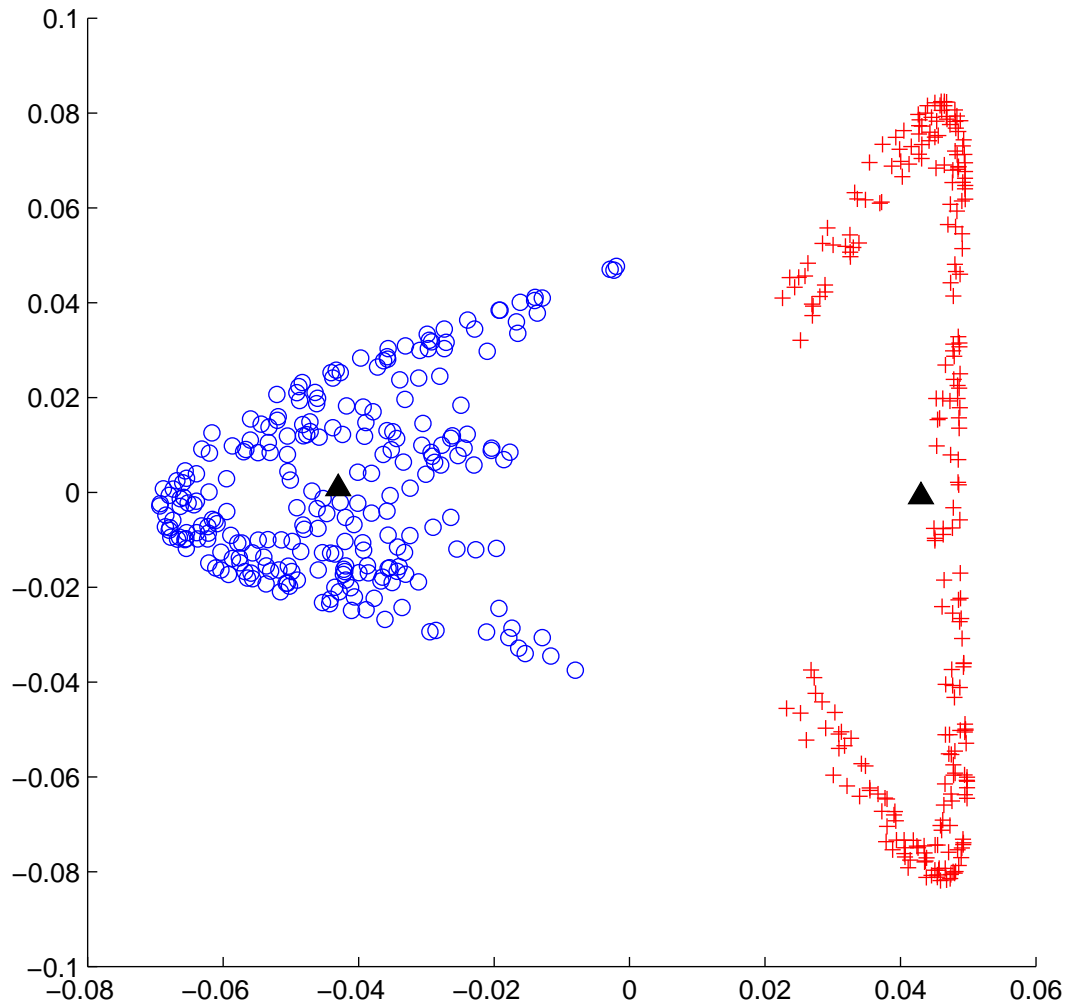
**A harder case:**

Spectral Bisection (PDDP)



➤ Use kernels to transform

Projection with Kernels --  $\sigma^2 = 2.7463$



Transformed data with a Gaussian Kernel

## *Fisher's Linear Discriminant Analysis (LDA)*

**Goal:** Use label information to define a good projector, i.e., one that can 'discriminate' well between given classes

- Define “**between scatter**”: a measure of how well separated two distinct classes are.
- Define “**within scatter**”: a measure of how well clustered items of the same class are.
- Objective: make “between scatter” measure large **and** “within scatter” small.

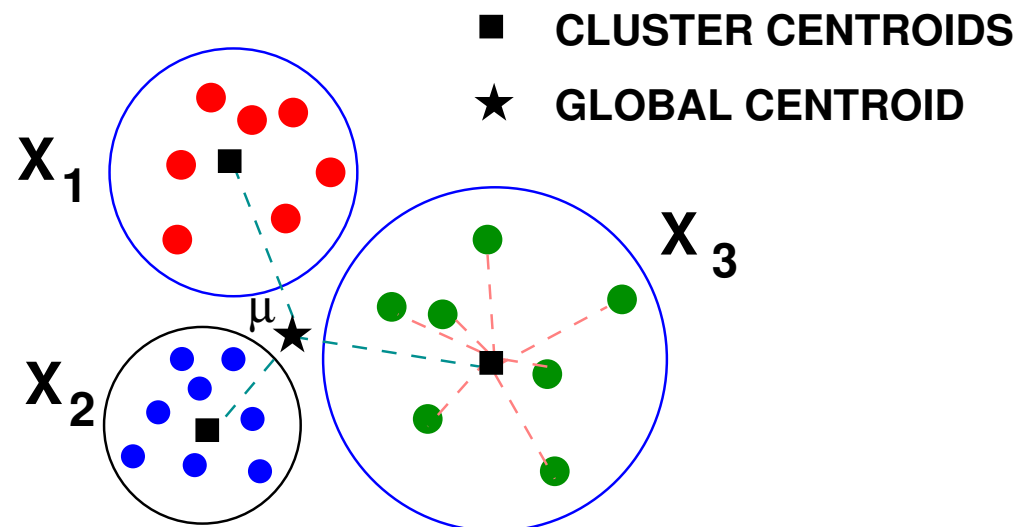
**Idea:** Find projector that maximizes the ratio of the “between scatter” measure over “within scatter” measure

Define:

Where:

$$S_B = \sum_{k=1}^c n_k (\mu^{(k)} - \mu) (\mu^{(k)} - \mu)^T,$$
$$S_W = \sum_{k=1}^c \sum_{x_i \in X_k} (x_i - \mu^{(k)}) (x_i - \mu^{(k)})^T$$

- $\mu$  = mean ( $X$ )
- $\mu^{(k)}$  = mean ( $X_k$ )
- $X_k$  =  $k$ -th class
- $n_k = |X_k|$



- Consider 2nd moments for a vector  $a$ :

$$a^T S_B a = \sum_{i=1}^c n_k |a^T (\mu^{(k)} - \mu)|^2,$$

$$a^T S_W a = \sum_{k=1}^c \sum_{x_i \in X_k} |a^T (x_i - \mu^{(k)})|^2$$

- $a^T S_B a \equiv$  weighted variance of projected  $\mu_j$ 's
- $a^T S_W a \equiv$  w. sum of variances of projected classes  $X_j$ 's

- LDA projects the data so as to maximize the ratio of these two numbers:

$$\max_a \frac{a^T S_B a}{a^T S_W a}$$

- Optimal  $a =$  eigenvector associated with the largest eigenvalue of:  $S_B u_i = \lambda_i S_W u_i .$

## LDA – Extension to arbitrary dimensions

- Criterion: maximize the ratio of two traces:

$$\frac{\text{Tr} [U^T S_B U]}{\text{Tr} [U^T S_W U]}$$

- Constraint:  $U^T U = I$  (orthogonal projector).
- Reduced dimension data:  $Y = U^T X$ .

*Common viewpoint:* hard to maximize, therefore ...

- ... alternative: Solve instead the ('easier') problem:

$$\max_{U^T S_W U = I} \text{Tr} [U^T S_B U]$$

- Solution: largest eigenvectors of  $S_B u_i = \lambda_i S_W u_i$ .

## LDA – Extension to arbitrary dimensions (cont.)

- Consider the original problem:

$$\max_{U \in \mathbb{R}^{n \times p}, U^T U = I} \frac{\text{Tr}[U^T A U]}{\text{Tr}[U^T B U]}$$

Let  $A, B$  be symmetric & assume that  $B$  is semi-positive definite with  $\text{rank}(B) > n - p$ . Then  $\text{Tr}[U^T A U] / \text{Tr}[U^T B U]$  has a finite maximum value  $\rho_*$ . The maximum is reached for a certain  $U_*$  that is unique up to unitary transforms of columns.

- Consider the function:

$$f(\rho) = \max_{V^T V = I} \text{Tr}[V^T (A - \rho B) V]$$

- Call  $V(\rho)$  the maximizer.
- Note:  $V(\rho)$  = Set of eigenvectors - not unique



- Define  $G(\rho) \equiv A - \rho B$  and its  $n$  eigenvalues:

$$\mu_1(\rho) \geq \mu_2(\rho) \geq \cdots \geq \mu_n(\rho) .$$

- Clearly:

$$f(\rho) = \mu_1(\rho) + \mu_2(\rho) + \cdots + \mu_p(\rho) .$$

- Can express this differently. Define eigenprojector:

$$P(\rho) = V(\rho)V(\rho)^T$$

- Then:

$$\begin{aligned} f(\rho) &= \text{Tr} [V(\rho)^T G(\rho) V(\rho)] \\ &= \text{Tr} [G(\rho) V(\rho) V(\rho)^T] \\ &= \text{Tr} [G(\rho) P(\rho)] . \end{aligned}$$

➤ Recall [e.g. Kato '65] that:

$$P(\rho) = \frac{-1}{2\pi i} \int_{\Gamma} (G(\rho) - zI)^{-1} dz$$

$\Gamma$  is a smooth curve containing the  $p$  eigenvalues of interest

➤ Hence:  $f(\rho) = \frac{-1}{2\pi i} \text{Tr} \int_{\Gamma} G(\rho)(G(\rho) - zI)^{-1} dz = \dots$

$$= \frac{-1}{2\pi i} \text{Tr} \int_{\Gamma} z(G(\rho) - zI)^{-1} dz$$

➤ With this, can prove :

1.  $f$  is a non-increasing function of  $\rho$ ;
2.  $f(\rho) = 0$  iff  $\rho = \rho_*$ ;
3.  $f'(\rho) = -\text{Tr} [V(\rho)^T B V(\rho)]$

*Can now use Newton's method.*

$$\rho_{new} = \rho - \frac{\text{Tr}[V(\rho)^T(A - \rho B)V(\rho)]}{-\text{Tr}[V(\rho)^T B V(\rho)]} = \frac{\text{Tr}[V(\rho)^T A V(\rho)]}{\text{Tr}[V(\rho)^T B V(\rho)]}$$

➤ Newton's method to find the zero of  $f \equiv$  a fixed point

iteration with

$$g(\rho) = \frac{\text{Tr}[V^T(\rho) A V(\rho)]}{\text{Tr}[V^T(\rho) B V(\rho)]},$$

➤ Idea: Compute  $V(\rho)$  by a Lanczos-type procedure

➤ Note: Standard problem - [not generalized]  $\rightarrow$  inexpensive!

➤ See T. Ngo, M. Bellalij, and Y.S. 2010 for details

➤ Recent papers advocated similar or related techniques

[1] C. Shen, H. Li, and M. J. Brooks, A convex programming approach to the trace quotient problem. In *ACCV (2) – 2007*.

[2] H. Wang, S.C. Yan, D.Xu, X.O. Tang, and T. Huang. Trace ratio vs. ratio trace for dimensionality reduction. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007*

[3] S. Yan and X. O. Tang, “Trace ratio revisited” Proceedings of the European Conference on Computer Vision, 2006.

...

## Background: The Lanczos procedure

### ALGORITHM : 1 . Lanczos

---

1. Choose an initial vector  $v_1$  of norm unity.  
Set  $\beta_1 \equiv 0, v_0 \equiv 0$
2. For  $j = 1, 2, \dots, m$  Do:
3.  $\alpha_j := (w_j, v_j)$
4.  $w_j := Av_j - \alpha_j v_j - \beta_j v_{j-1}$
5.  $\beta_{j+1} := \|w_j\|_2$ . If  $\beta_{j+1} = 0$  then Stop
6.  $v_{j+1} := w_j / \beta_{j+1}$
7. EndDo

➤ In first few steps of Newton: rough approximation needed.

## **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

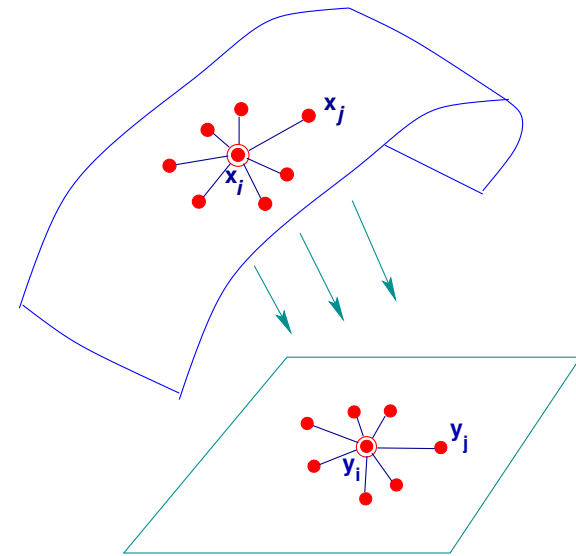
- Define a **graph Laplacean:**

$$L = D - W$$

$$\text{e.g.,: } w_{ij} = \begin{cases} 1 & \text{if } j \in N_i \\ 0 & \text{else} \end{cases}$$

$$D = \text{diag} \left[ d_{ii} = \sum_{j \neq i} w_{ij} \right]$$

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



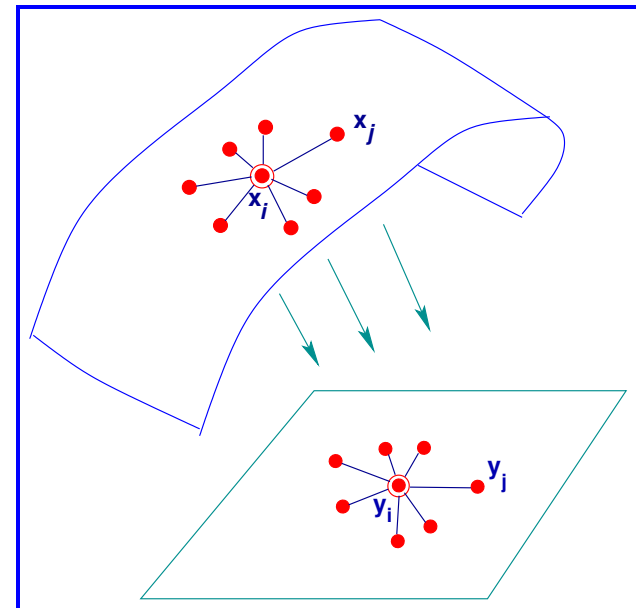
## Example: The Laplacean eigenmaps approach

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

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

**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]





## Locally Linear Embedding (Roweis-Saul-00)

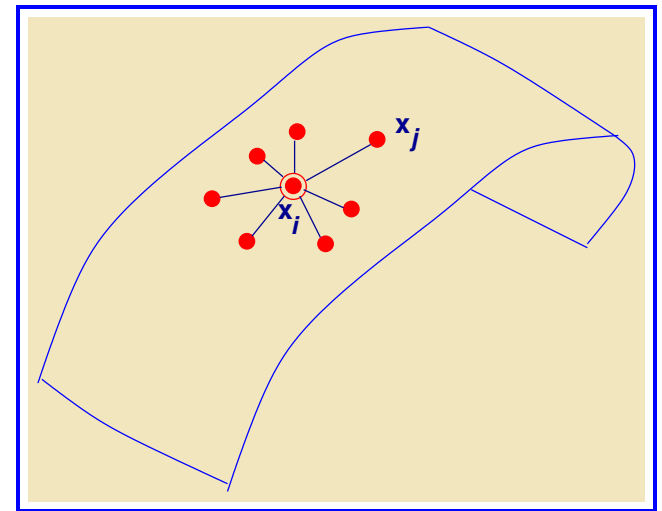
- Very similar to Eigenmaps.
- Graph Laplacean matrix is replaced by an 'affinity' graph

**Graph:** Each  $x_i$  written as a convex combination of its  $k$  nearest neighbors:

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

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

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



- Mapped data ( $Y$ ) computed by minimizing

$$\sum \|y_i - \sum w_{ij} y_j\|^2$$

## *Implicit vs explicit mappings*

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

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

- In Eigenmaps and LLE we only know

$$\mathbf{y}_i = \phi(\mathbf{x}_i), i = 1, \dots, n$$

- Not easy to get  $\phi(\mathbf{x})$  for an arbitrary  $\mathbf{x}$  not in the sample.
- Methods are powerful but inconvenient for classification
- “The out-of-sample extension” problem

## *ONPP (Kokopoulou and YS '05)*

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

$$\begin{aligned} \min_V \text{Tr} \left[ V^T X (I - W^T) (I - W) X^T V \right] \\ \text{s.t. } V^T V = I \end{aligned}$$

- In LLE replace  $V^T X$  by  $Y$

## *Face Recognition – background*

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



## Face Recognition – background

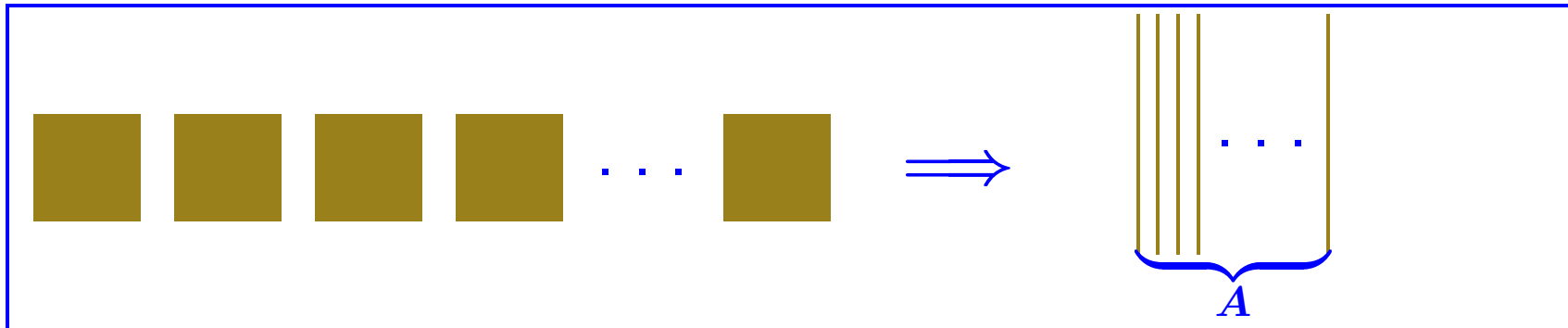
**Problem:** We are given a database of images: [arrays of pixel values]. And a test (new) image.



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

## 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
- Similar to LSI in spirit – but data is not sparse.

## 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:  $\text{rank}(W) = n - c$ .

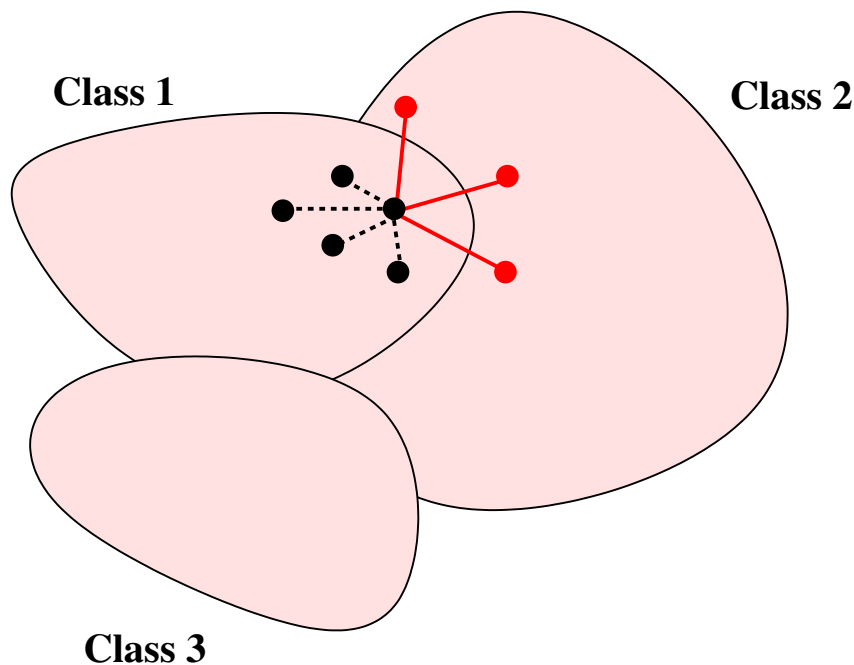
➤ As before, graph Laplacean:

$$L_c = D - W$$

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

➤ Can be used for ONPP and other graph based methods

➤ Improvement: add **repulsion Laplacean** [Kokiopoulou, YS 09]



Leads to eigenvalue problem with matrix:

$$L_C - \rho L_R$$

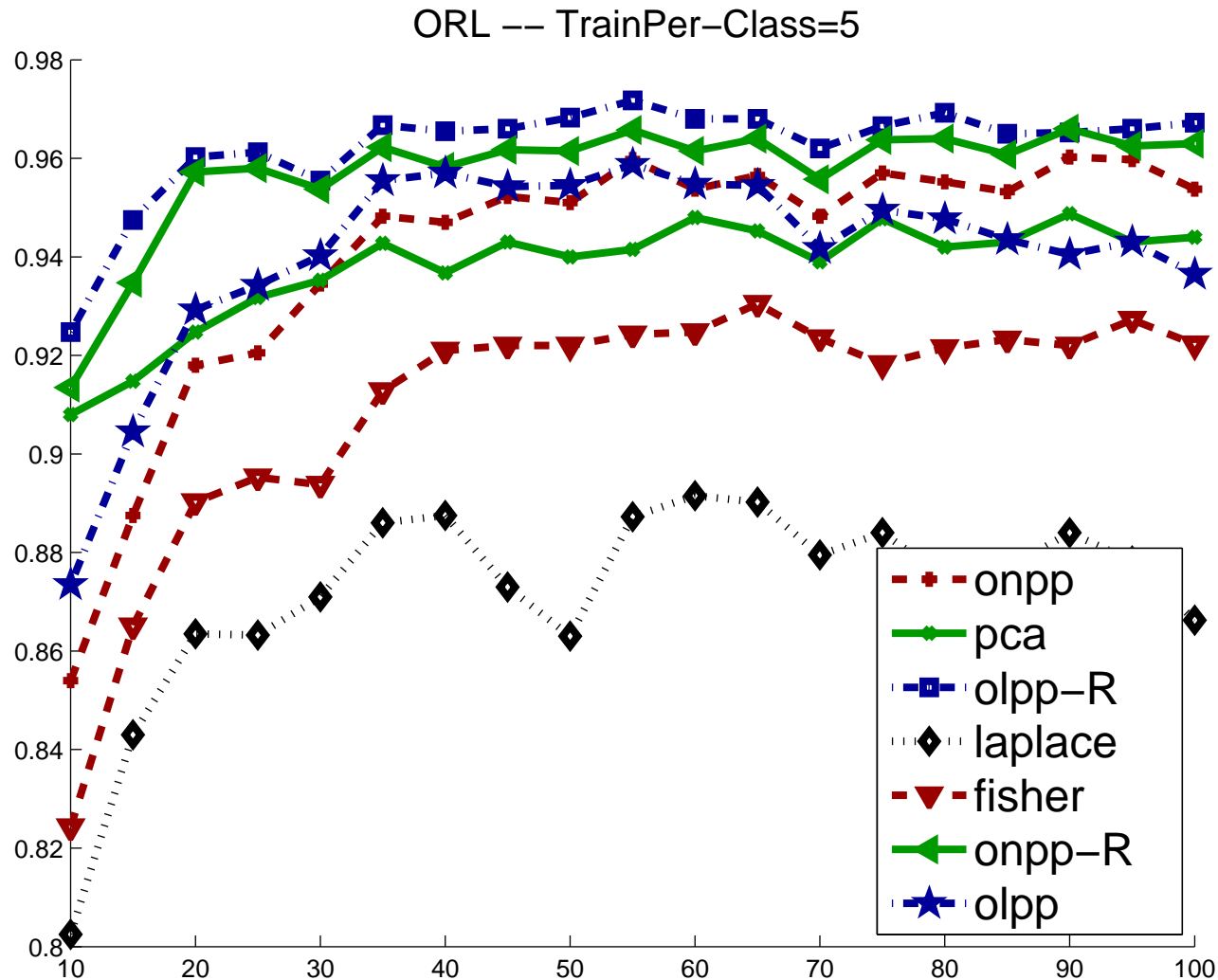
- $L_C$  = class-Laplacian,
- $L_R$  = repulsion Laplacian,
- $\rho$  = parameter

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



# of pixels :  $112 \times 92$ ;      TOT. # images : 400





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

## **LINEAR ALGEBRA METHODS: EXAMPLES**

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

- Lanczos is good at catching large (and small) eigenvalues: can compute singular vectors with Lanczos, & use them in LSI
  - Can do better: Use the Lanczos vectors directly for the projection..
  - See: K. Blom and A. Ruhe [SIMAX, vol. 26, 2005]
  - Major difference with what we do: A Krylov subspace for each query [expensive].
- Proposed: One Lanczos run- random initial vector. Then use Lanczos vectors in place of singular vectors.

# Tests: IR

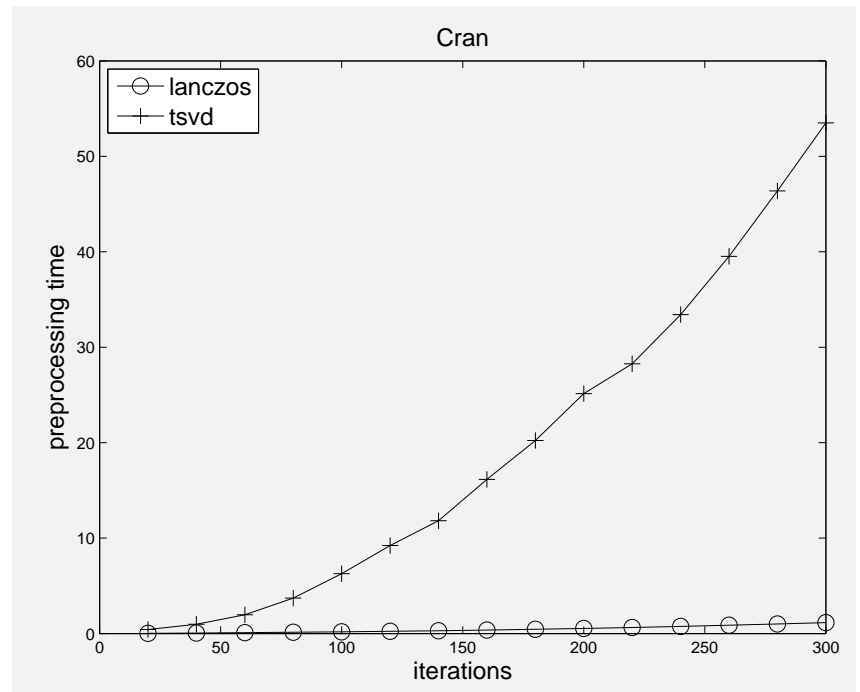
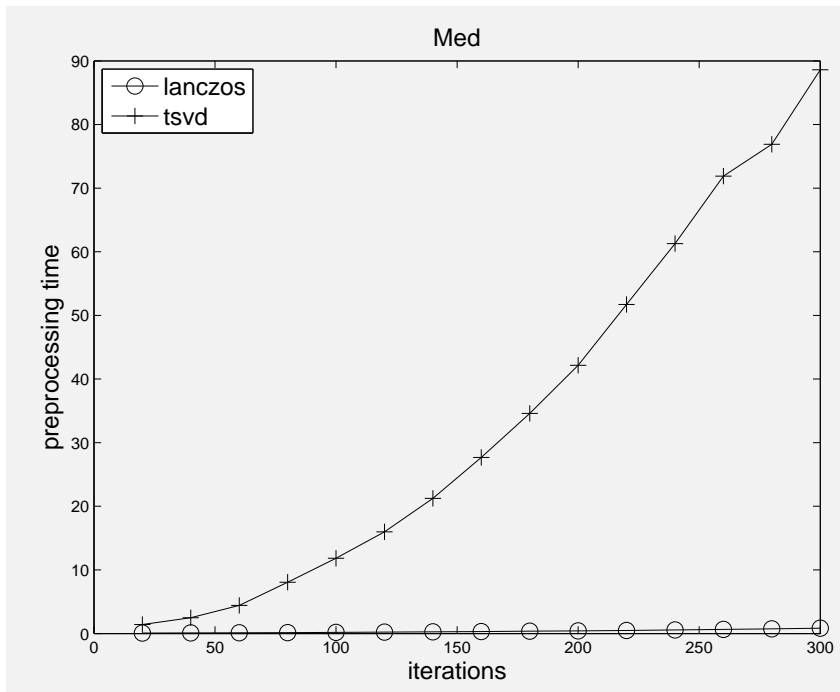
Information  
retrieval  
datasets

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

Med dataset.

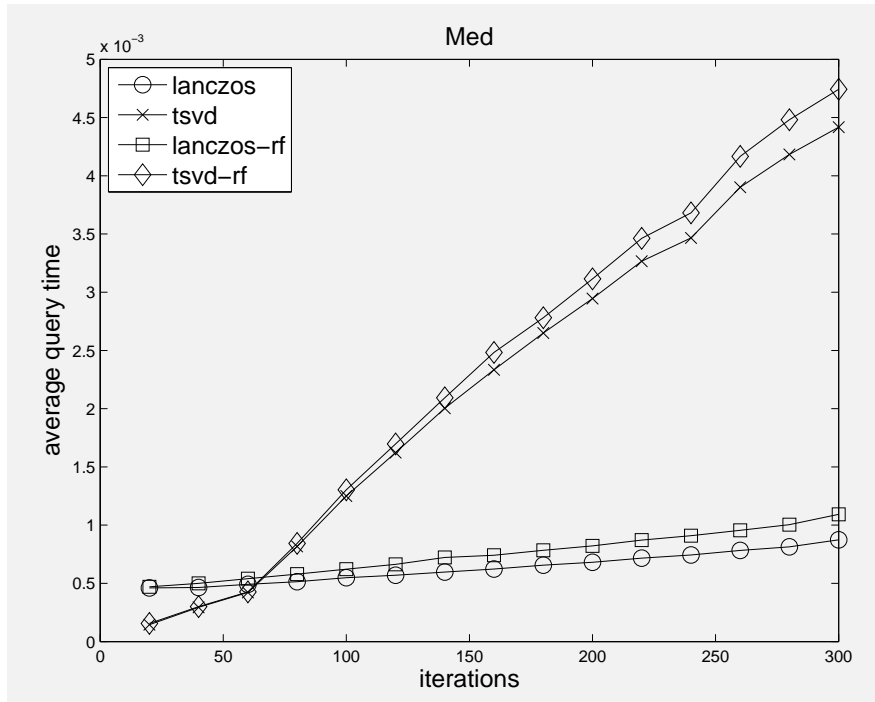
Cran dataset.

Preprocessing times

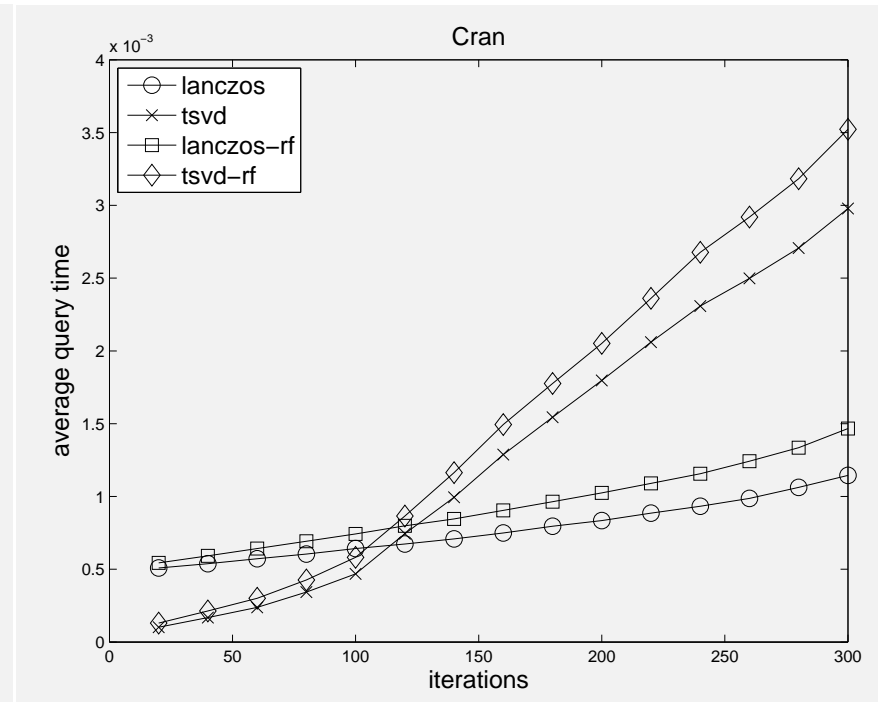


# Average query times

## Med dataset

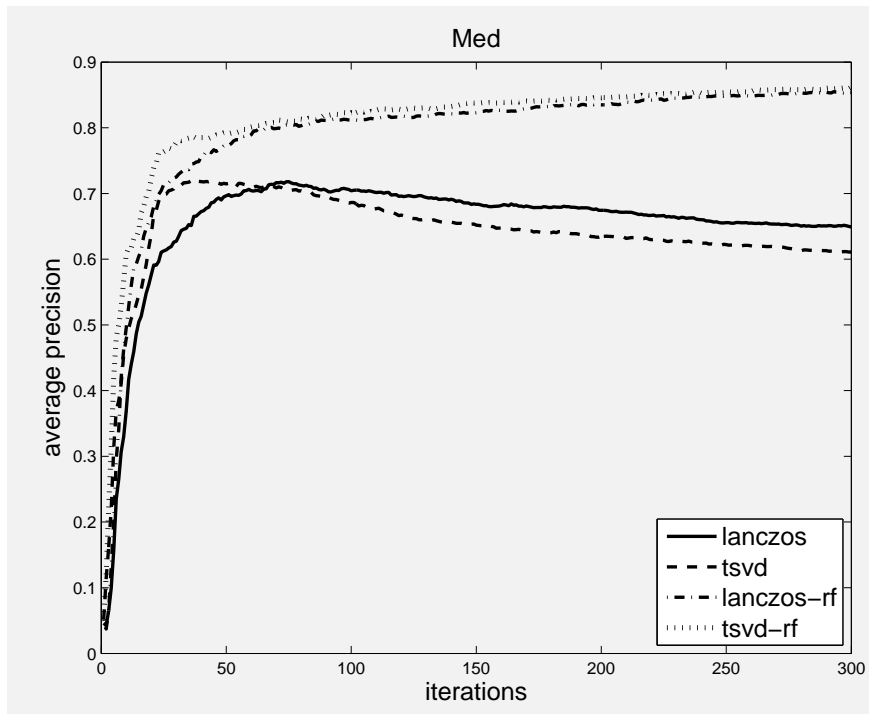


## Cran dataset.

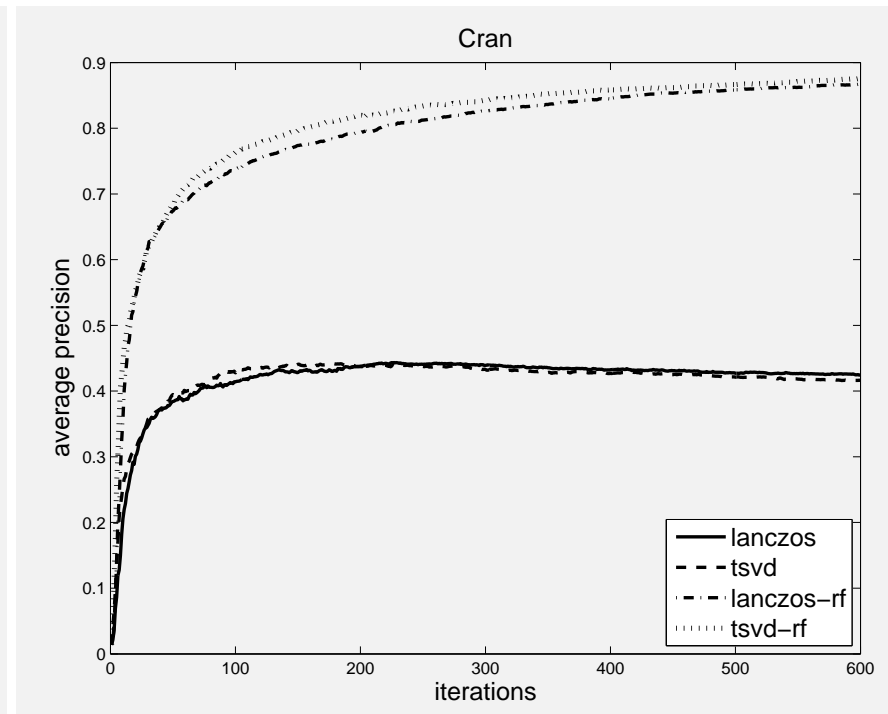


# Average retrieval precision

## Med dataset



## Cran dataset



## Retrieval precision comparisons

### *In summary:*

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

### *Thanks:*

- Helpful tools and datasets widely available. We used TMG [developed at the U. of Patras (D. Zeimpekis, E. Gallopoulos)]

## Updating the SVD (E. Vecharynski and YS'13)

- In applications, data matrix  $X$  often updated
- Example: Information Retrieval (IR), can add documents, add terms, change weights, ..

### Problem

Given the partial SVD of  $X$ , how to get a partial SVD of  $X_{new}$

- Will illustrate only with update of the form  $X_{new} = [X, D]$  (documents added in IR)



## Updating the SVD: Zha-Simon algorithm

- Assume  $A \approx U_k \Sigma_k V_k^T$  and  $A_D = [A, D]$ ,  $D \in \mathbb{R}^{m \times p}$
- Compute  $D_k = (I - U_k U_k^T) D$  and its QR factorization:

$$[\hat{U}_p, R] = qr(D_k, 0), \quad R \in \mathbb{R}^{p \times p}, \quad \hat{U}_p \in \mathbb{R}^{m \times p}$$

Note:  $A_D \approx [U_k, \hat{U}_p] H_D \begin{bmatrix} V_k & 0 \\ 0 & I_p \end{bmatrix}^T$ ;  $H_D \equiv \begin{bmatrix} \Sigma_k & U_k^T D \\ 0 & R \end{bmatrix}$

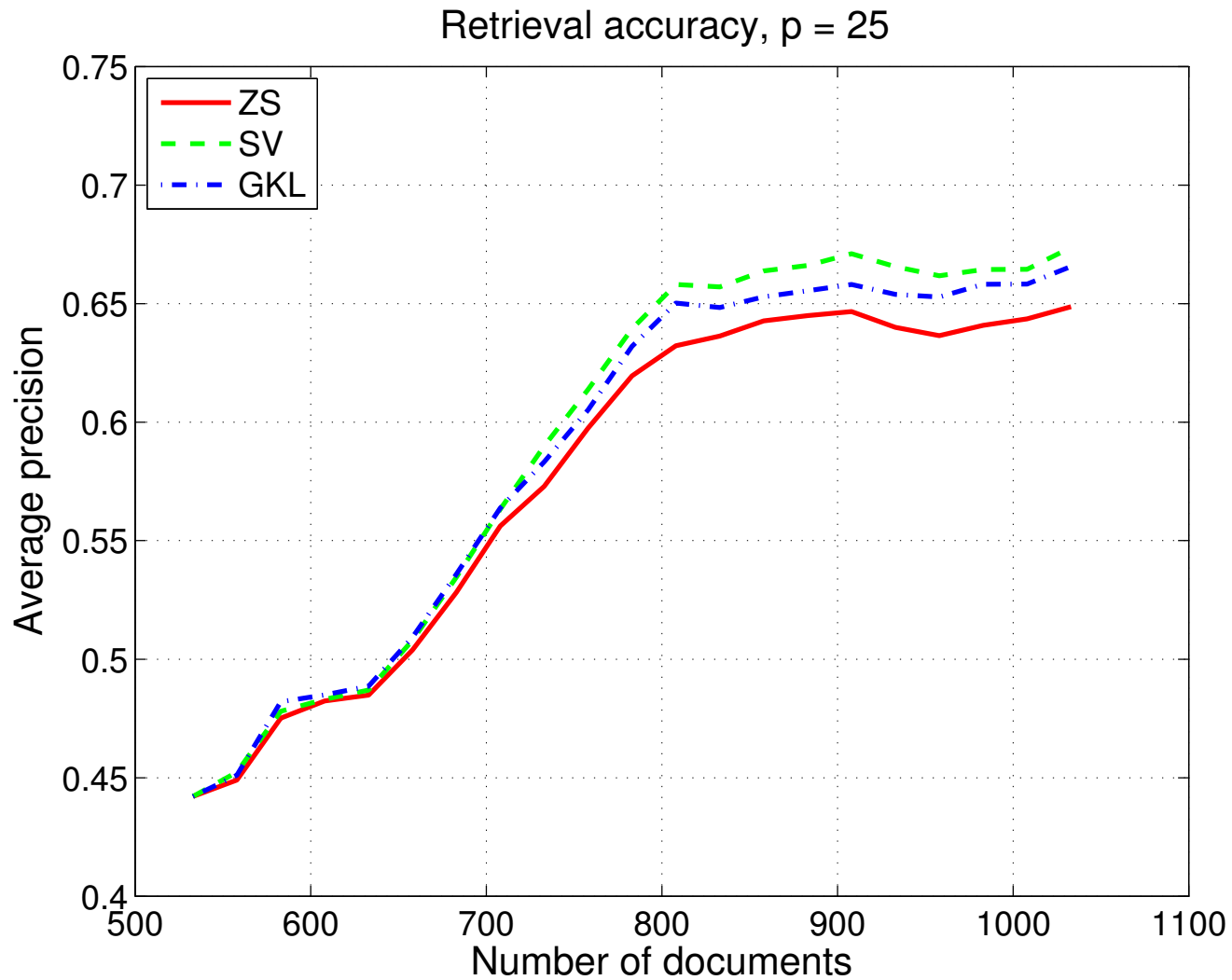
- Zha–Simon ('99): Compute the SVD of  $H_D$  & get approximate SVD from above equation
- It turns out this is a Rayleigh-Ritz projection method for the SVD [E. Vecharynski & YS 2013]
- Can show optimality properties as a result

## Updating the SVD

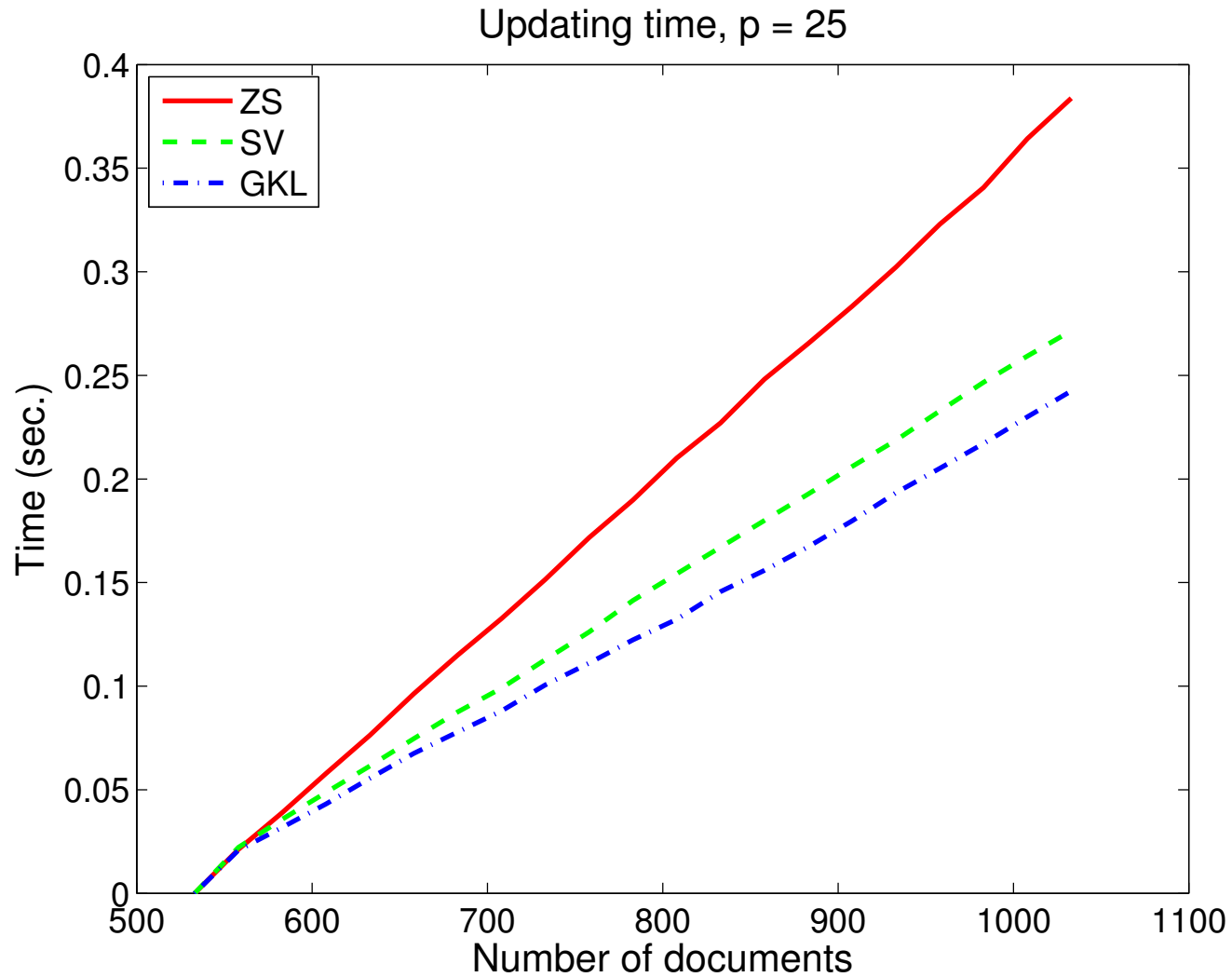
- When the number of updates is large this becomes costly.
- Idea: Replace  $\hat{U}_p$  by a low dimensional approximation:
- Use  $\bar{U}$  of the form  $\bar{U} = [U_k, Z_l]$  instead of  $\bar{U} = [U_k, \hat{U}_p]$
- $Z_l$  must capture the range of  $D_k = (I - U_k U_k^T) D$
- Simplest idea : best rank- $l$  approximation using the SVD.
- Can also use Lanczos vectors from the Golub-Kahan-Lanczos algorithm.

## *An example*

- LSI - with MEDLINE collection:  $m = 7,014$  (terms),  $n = 1,033$  (docs),  $k = 75$  (dimension),  $t = 533$  (initial # docs),  $n_q = 30$  (queries)
- Adding blocks of 25 docs at a time
- The number of singular triplets of  $(I - U_k U_k^T) D$  using SVD projection (“SV”) is 2.
- For GKL approach (“GKL”) 3 GKL vectors are used
- These two methods are compared to Zha-Simon (“ZS”).
- We show average precision then time



➤ Experiments show: gain in accuracy is rather consistent



➤ Times can be significantly better for large sets

## **APPLICATION TO MATERIALS**

## *Data mining for materials: Materials Informatics*

➤ Huge potential in exploiting two trends:

**1** Improvements in efficiency and capabilities in computational methods for materials

**2** Recent progress in data mining techniques

➤ Current practice: “One student, one alloy, one PhD” [see special MRS issue on materials informatics] → Slow ..

➤ Data Mining: can help speed-up process, e.g., by exploring in smarter ways

**Issue 1:** Who will do the work? Few researchers are familiar with both worlds

*Issue 2:* databases, and more generally sharing, not too common in materials

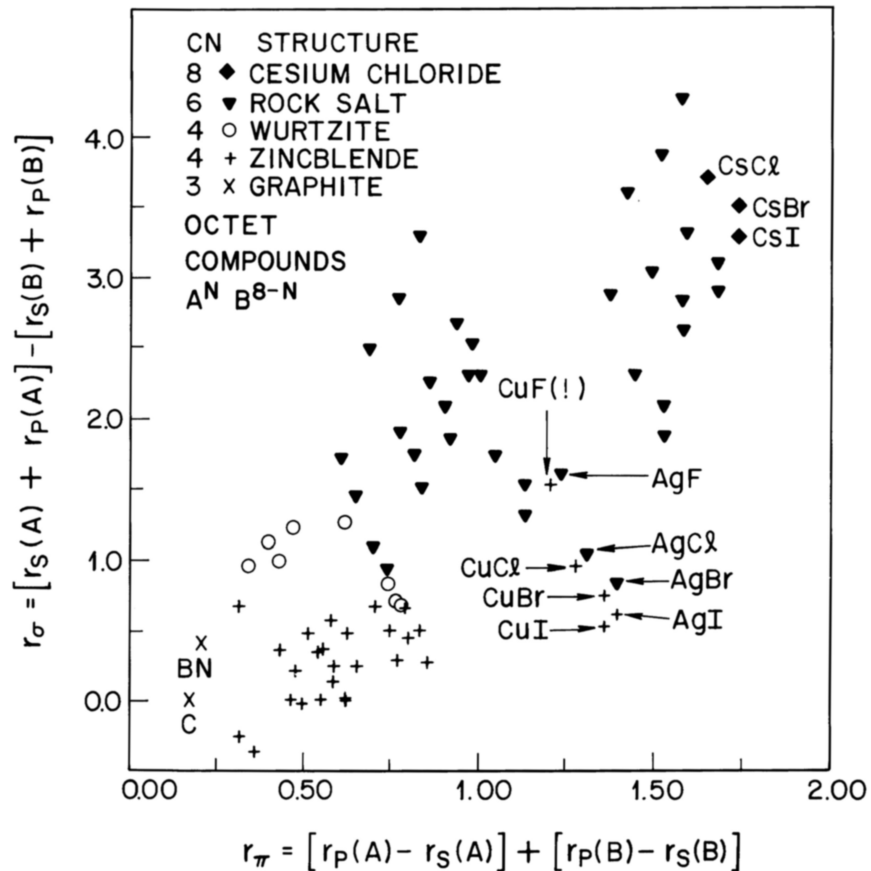
*The inherently fragmented and multidisciplinary nature of the materials community poses barriers to establishing the required networks for sharing results and information. One of the largest challenges will be encouraging scientists to think of themselves not as individual researchers but as part of a powerful network collectively analyzing and using data generated by the larger community. These barriers must be overcome.*

NSTC report to the White House, June 2011.

➤ Materials genome initiative [NSF]



# Unsupervised learning



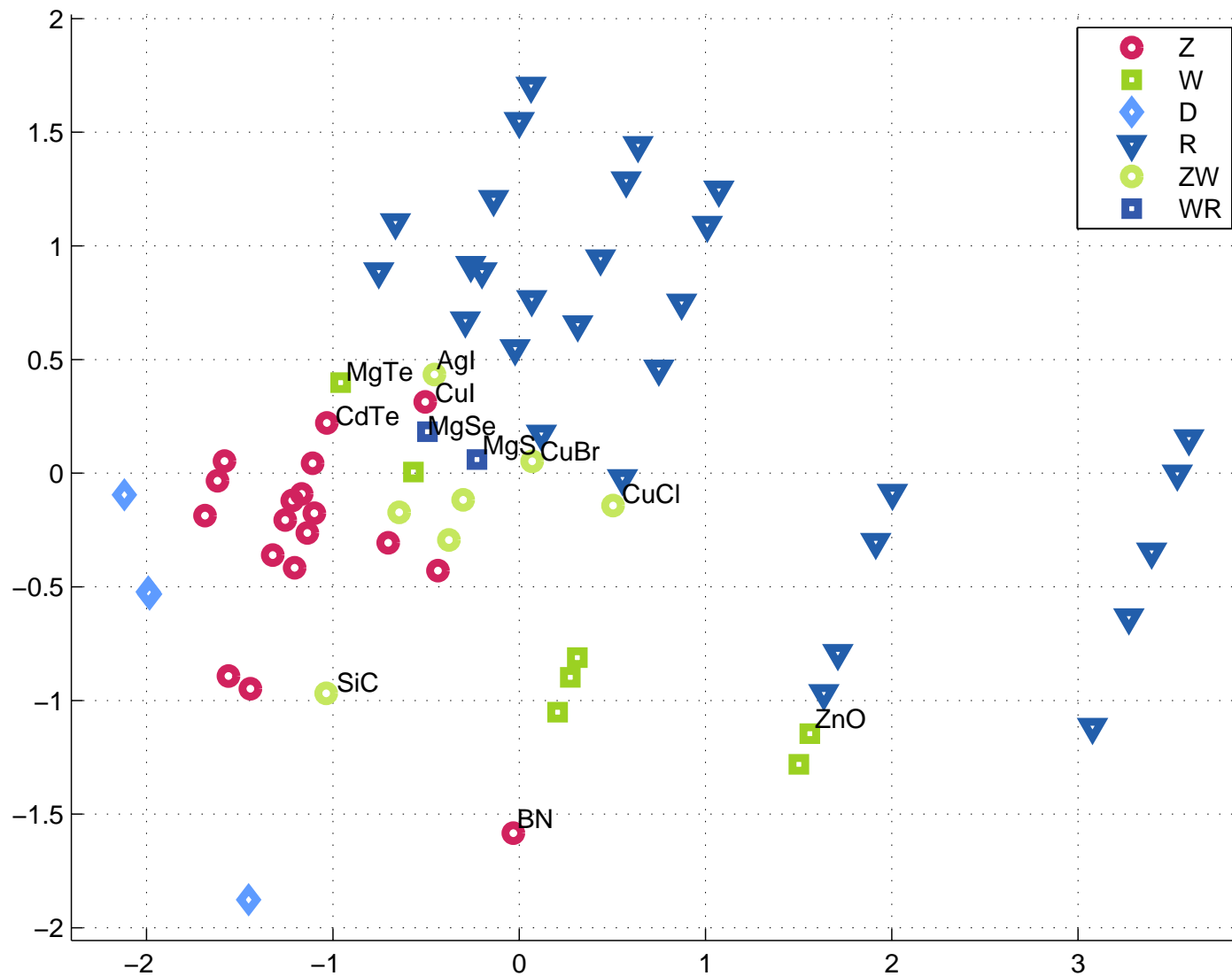
- 1970s: Data Mining “by hand”: Find coordinates that will cluster materials according to structure
- 2-D projection from physical knowledge
- ‘Anomaly Detection’: helped find that compound Cu F does not exist

see: J. R. Chelikowsky, J. C. Phillips, Phys Rev. B 19 (1978).

*Question:* Can modern data mining achieve a similar diagrammatic separation of structures?

- Should use only information from the two constituent atoms
- Experiment: 67 binary 'octets'.
- Use PCA – exploit only data from 2 constituent atoms:
  1. Number of valence electrons;
  2. Ionization energies of the s-states of the ion core;
  3. Ionization energies of the p-states of the ion core;
  4. Radii for the s-states as determined from model potentials;
  5. Radii for the p-states as determined from model potentials.

➤ Result:



## *Supervised learning: classification*

- Problem: classify an unknown binary compound into its crystal structure class
- 55 compounds, 6 crystal structure classes
- “leave-one-out” experiment

**Case 1:** Use features 1:5 for atom A and 2:5 for atom B. No scaling is applied.

**Case 2:** Features 2:5 from each atom + scale features 2 to 4 by square root of # valence electrons (feature 1)

**Case 3:** Features 1:5 for atom A and 2:5 for atom B. Scale features 2 and 3 by square root of # valence electrons.

## *Three methods tested*

1. PCA classification. Project and do identification in space of reduced dimension (Euclidean distance in low-dim space).
2. KNN K-nearest neighbor classification –
3. Orthogonal Neighborhood Preserving Projection (ONPP) - a graph based method - [see Kokiopoulou, YS, 2005]

Recognition rates for 3 different methods using different features


| Case   | KNN   | ONPP  | PCA   |
|--------|-------|-------|-------|
| Case 1 | 0.909 | 0.945 | 0.945 |
| Case 2 | 0.945 | 0.945 | 1.000 |
| Case 3 | 0.945 | 0.945 | 0.982 |

# Recent work

## ➤ Some data is becoming available

Materials Project :: Home https://materialsproject.org

---

 [Home](#) [Apps](#) [Resources](#) [About](#) [References](#) [Dashboard](#) [Logout](#)

# MATERIALS PROJECT


A Materials Genome Approach

Accelerating materials discovery through advanced scientific computing and innovative design tools.


Search powered by **MOOGL**

### Database Statistics


|                                    |                                   |
|------------------------------------|-----------------------------------|
| <b>38151</b> materials             | <b>14618</b> bandstructures       |
| <b>610</b> intercalation batteries | <b>16277</b> conversion batteries |




**Materials Explorer**  
Search for material's information by chemistry, composition, or property.




**Lithium Battery Explorer**  
Find candidate materials for lithium batteries. Get voltage profiles and cogeneration data.




**Crystal Toolkit**  
Convert between CIF and VASP input files. Generate new crystals by substituting or removing species.



**Phase Diagram App**  
Computational phase diagrams for closed and open systems. Find stable phases and study reaction pathways.



**Reaction Calculator**  
Calculate the enthalpy of tens of thousands of reactions and compare with experimental values.

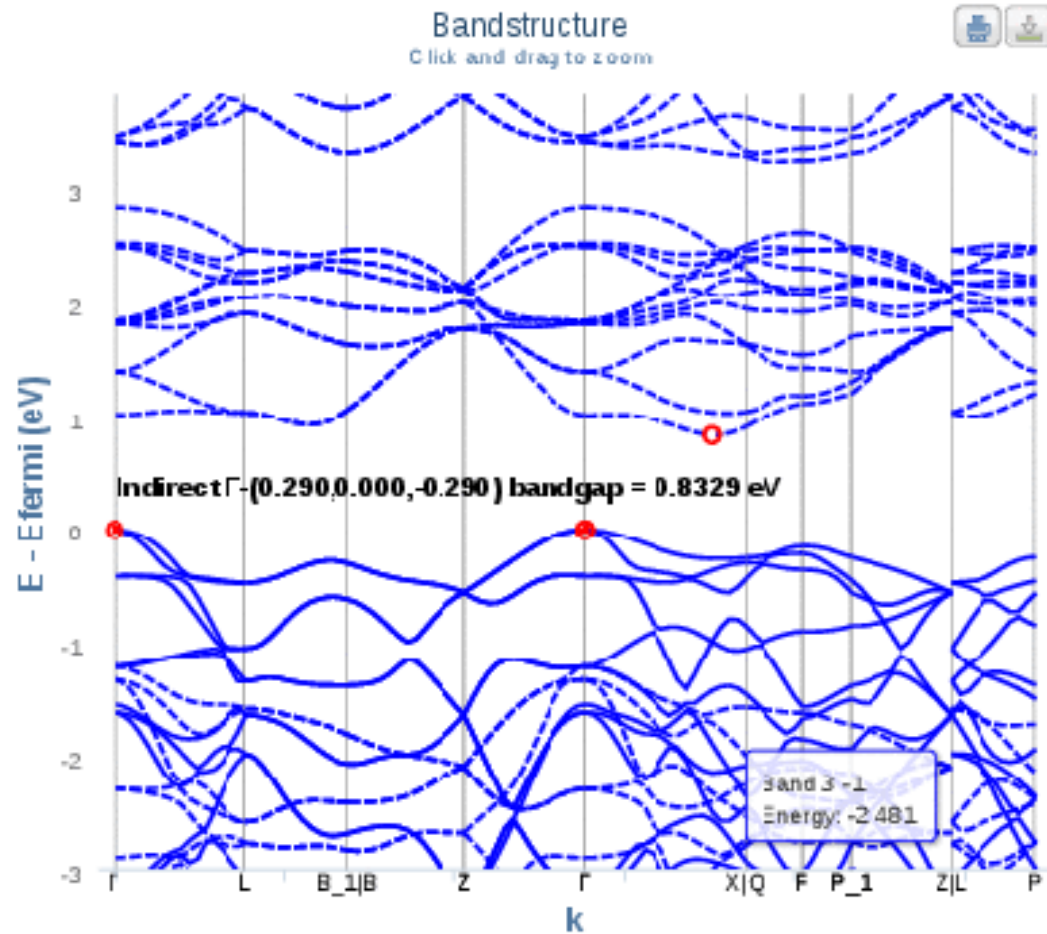


**Pourbaix Diagrams**  
Generate Pourbaix Diagrams from experimental ion data.

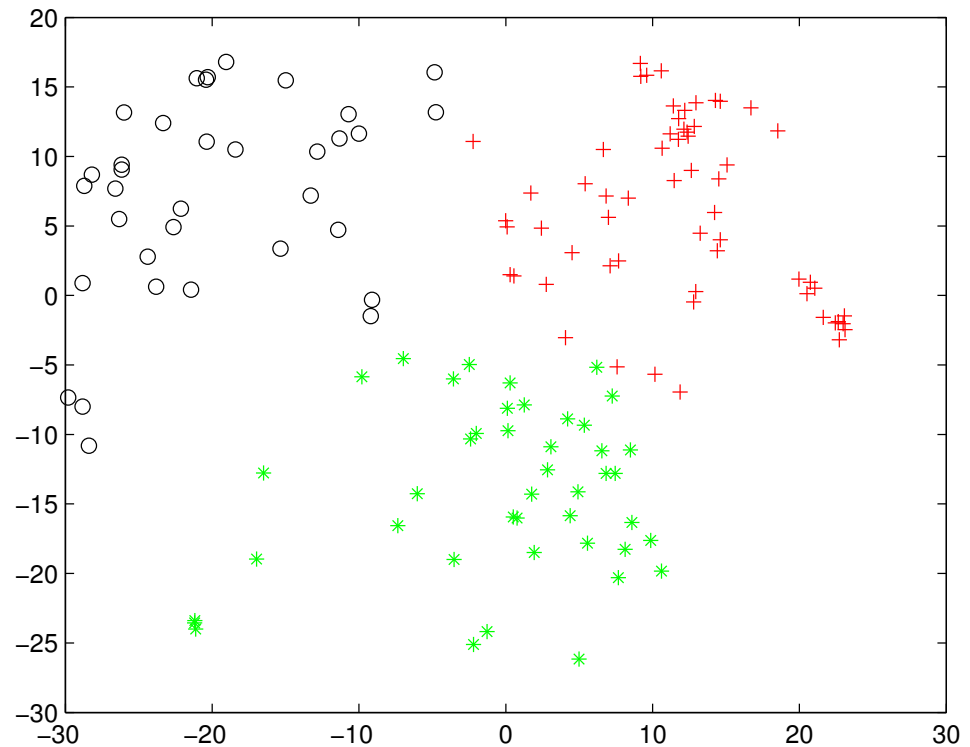
Find out more about our open [Materials API](#) and [pymatgen](#) library for querying large amounts of data.

## Recent work

- Exploit Bandstructures - in the same way we use images..
- For now we do clustering.



- Work in progress
- 3-way clustering obtained with dim. reduction + k-means →
- Working on unraveling the info & exploring classification with the data





## *Conclusion*

- Many, 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
- On the + side: **materials genome** project is starting to energize the field

- 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 !”*

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

*“The quicker you let go of old cheese, the sooner you find new cheese.”*

**Thank you !**