Numerical Linear algebra methods for data mining

Yousef Saad

Department of Computer Science and Engineering

University of Minnesota

William & Mary

Oct. 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 (2.3×10^{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 – emphasis on linear algebra viewpoint

- + our initial work on materials.

- Focus on "Dimension reduction methods"

Major tool of Data Mining: Dimension reduction

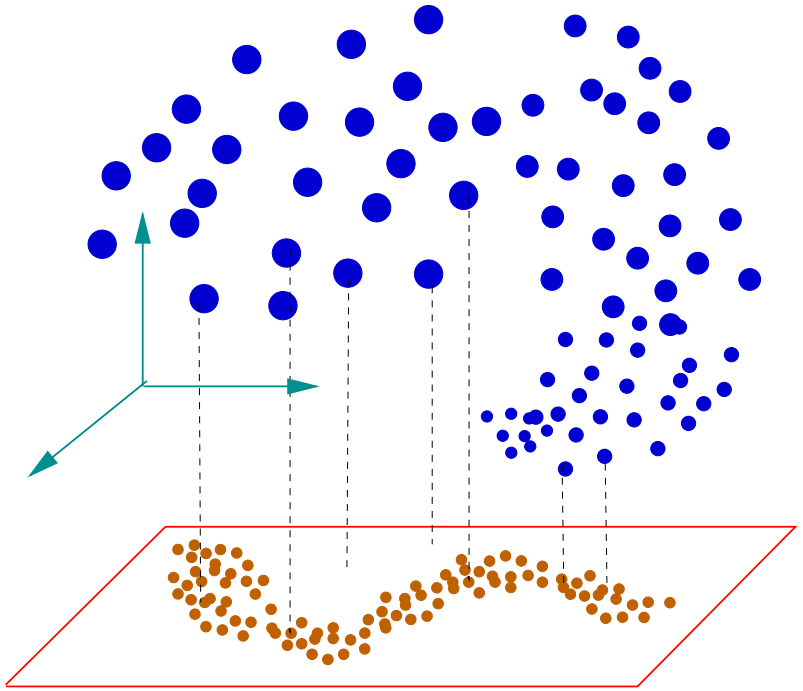
➤ Goal is not as much to reduce size (& cost) but to:

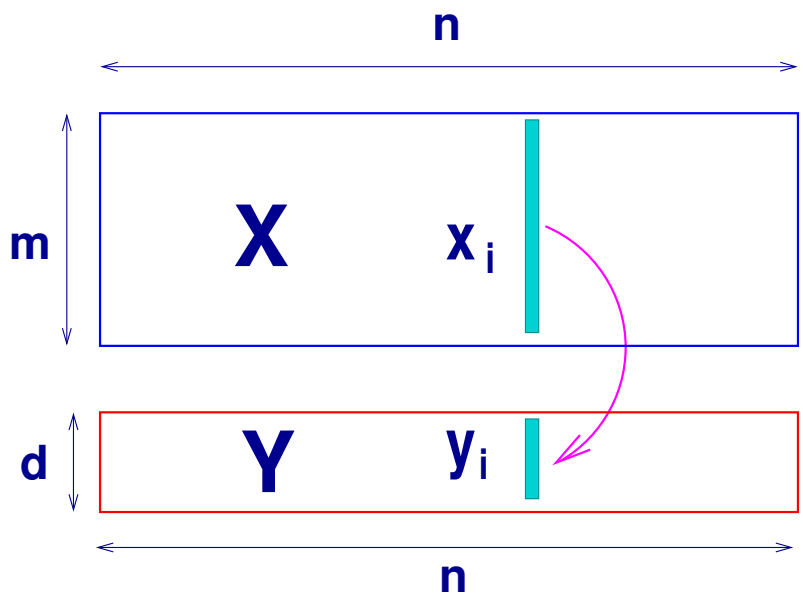
- Reduce noise and redundancy in data before performing a task [e.g., classification as in digit/face recognition]
- Discover important 'features' or 'paramaters'

The problem: Given: $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$, find a low-dimens. representation $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n] \in \mathbb{R}^{d \times n}$ of $\boldsymbol{X}$

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

$$\phi(\boldsymbol{x}_i) = \boldsymbol{y}_i, \quad \boldsymbol{i} = 1, \cdots, \boldsymbol{n}$$





- 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:

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

- Leads to maximizing

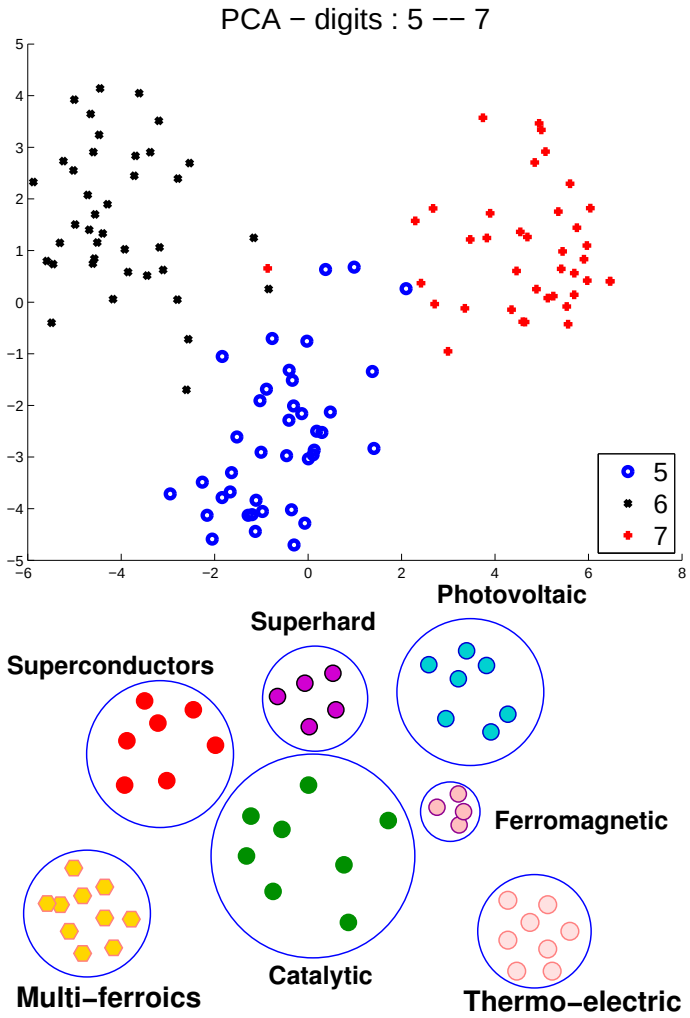
$$\text{Tr}\left[\boldsymbol{W}^\top(\boldsymbol{X}-\boldsymbol{\mu}\boldsymbol{e}^\top)(\boldsymbol{X}-\boldsymbol{\mu}\boldsymbol{e}^\top)^\top\boldsymbol{W}\right],\quad \boldsymbol{\mu}=\tfrac{1}{n}\Sigma_{i=1}^{n}\boldsymbol{x}_i$$

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

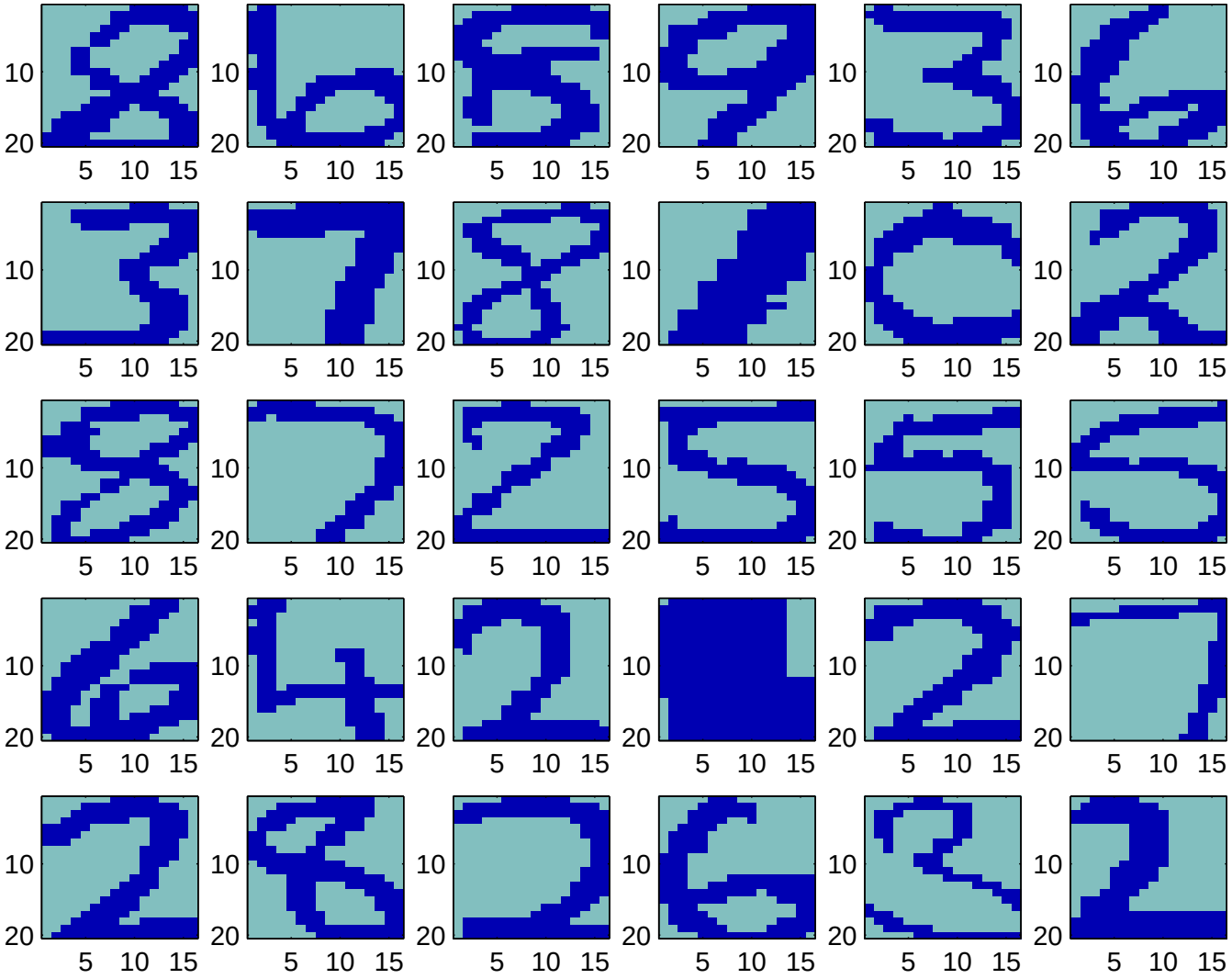
Unsupervised learning

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

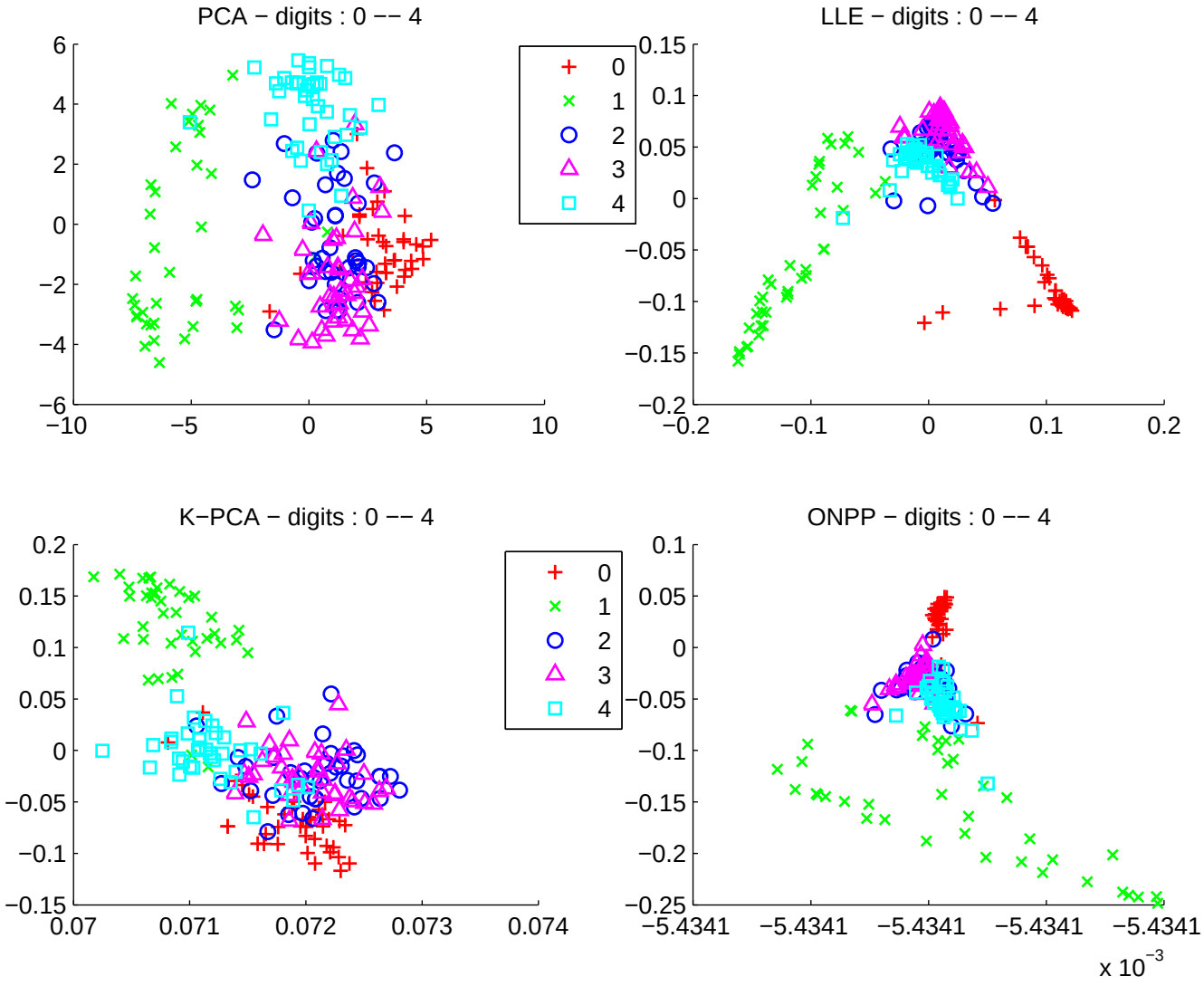
- Example of digits: perform a 2-D projection
- Images of same digit tend to cluster (more or less)
- Such 2-D representations are popular for visualization
- Can also try to find natural clusters in data, e.g., in materials
- Basic clusterning technique: K-means



Example: Digit images (a random sample of 30)

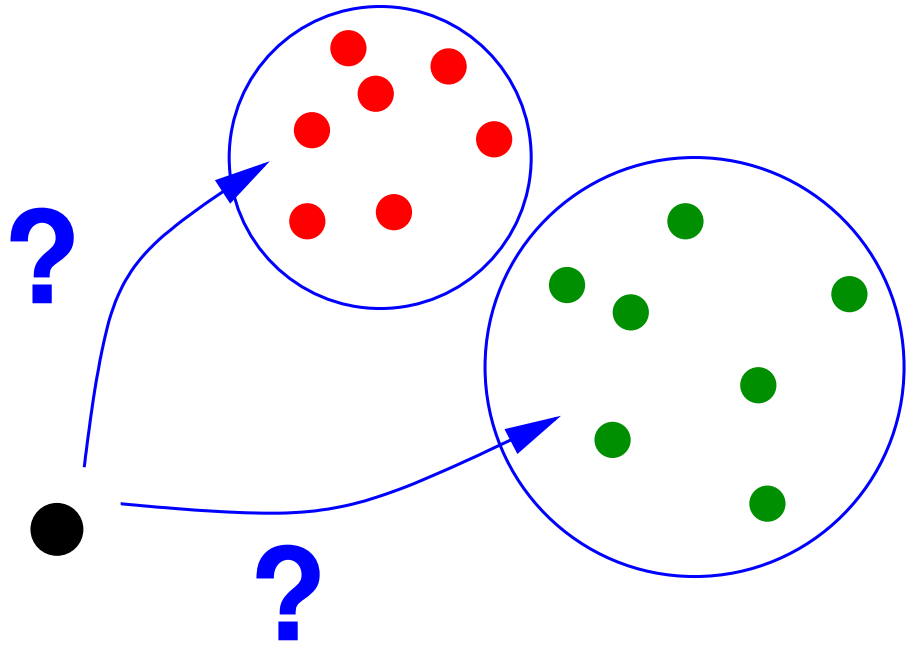


2-D 'reductions':



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.



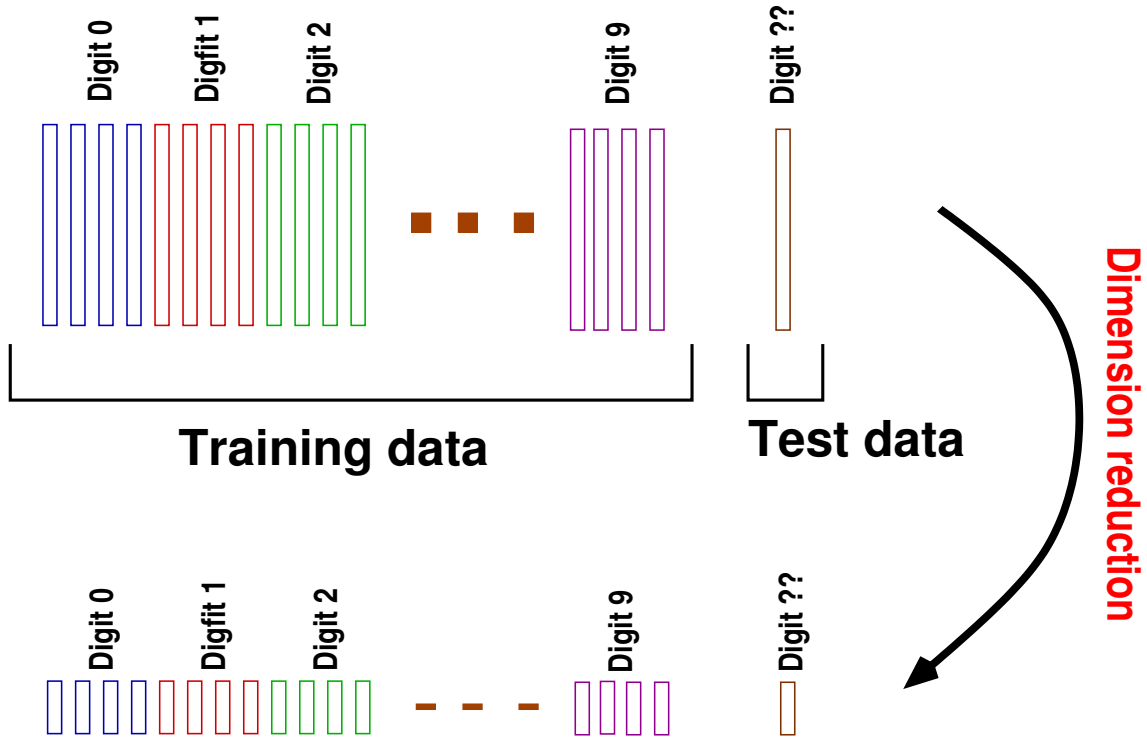
- Many applications.
- Example: distinguish SPAM and non-SPAM messages
- Can be extended to more than 2 classes.

Supervised learning: classification

- Best illustration: written digits recognition example

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

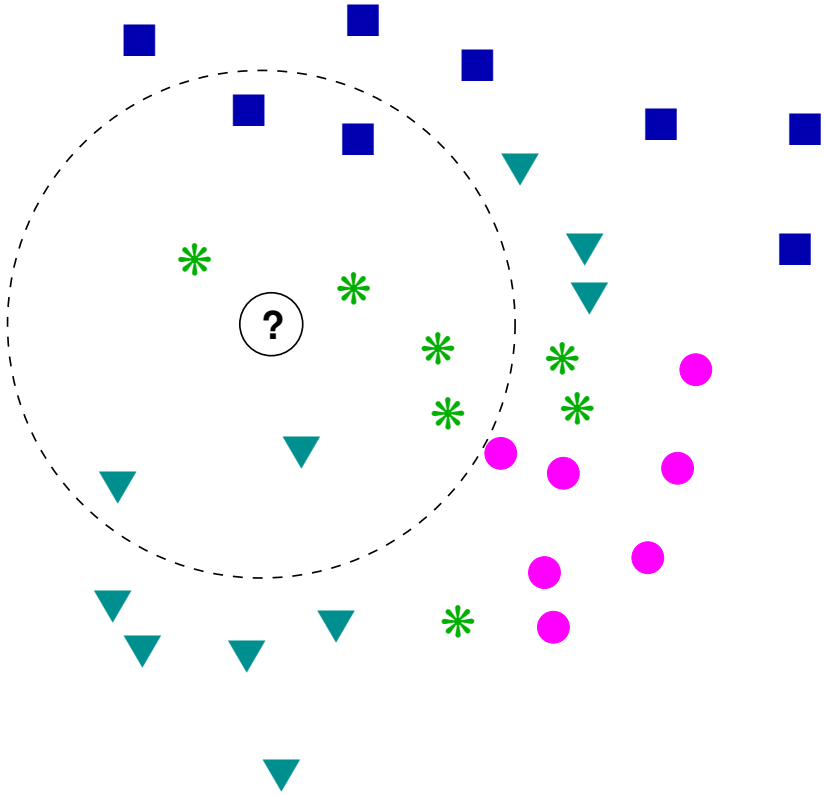
Problem: find label of test image



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

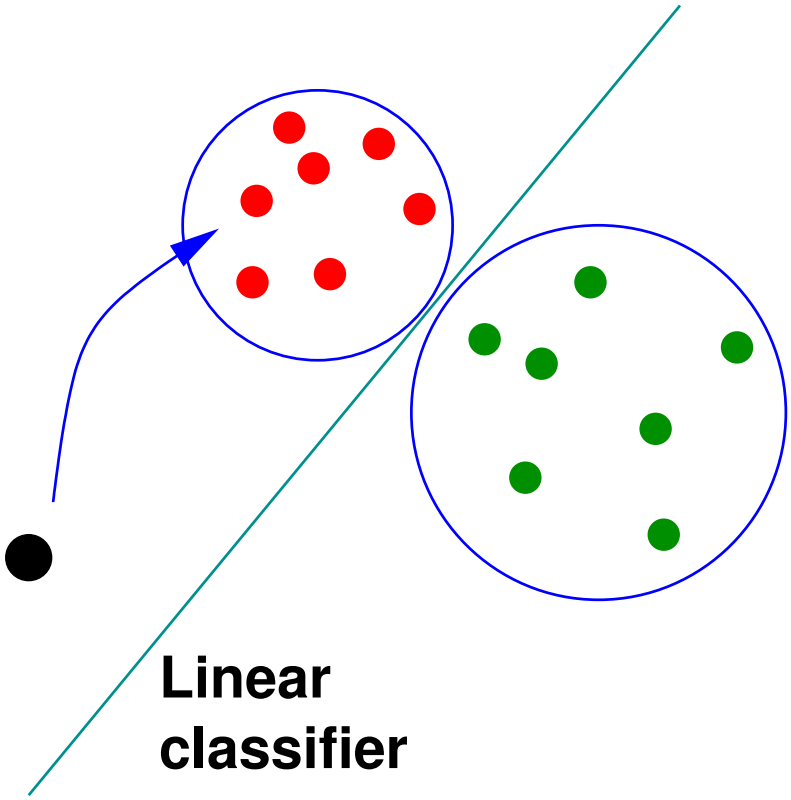
Basic method: K-nearest neighbors (KNN) classification

- 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)



Supervised learning: Linear classification

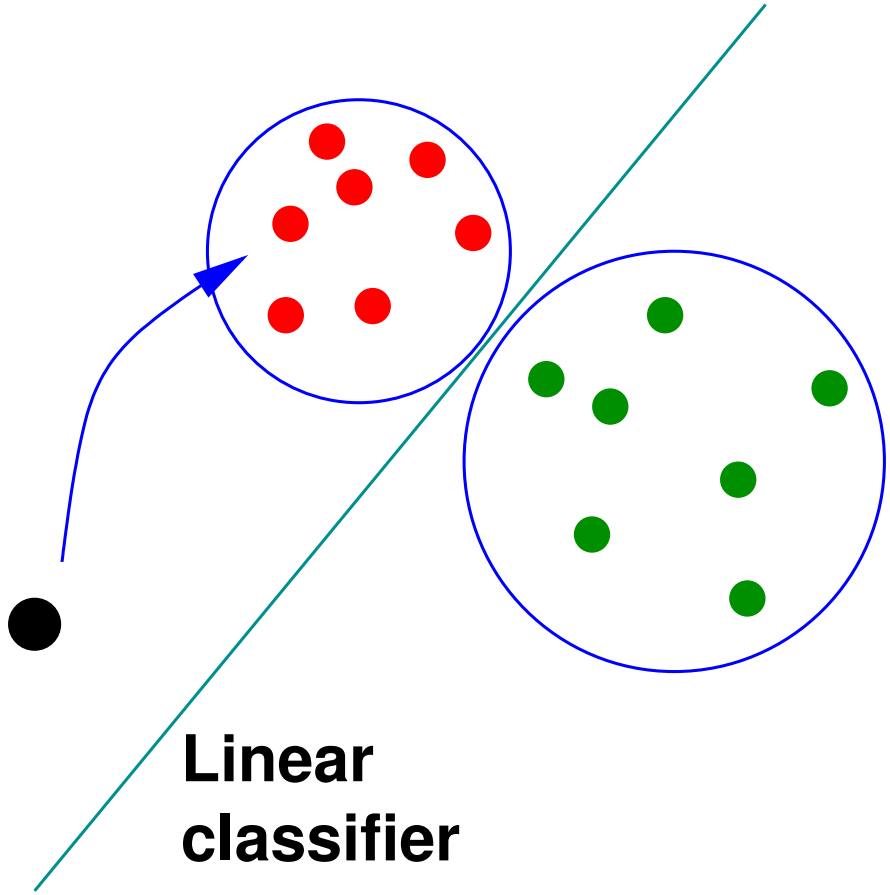
Linear classifiers: Find a hyperplane which best separates the data in classes A and B.



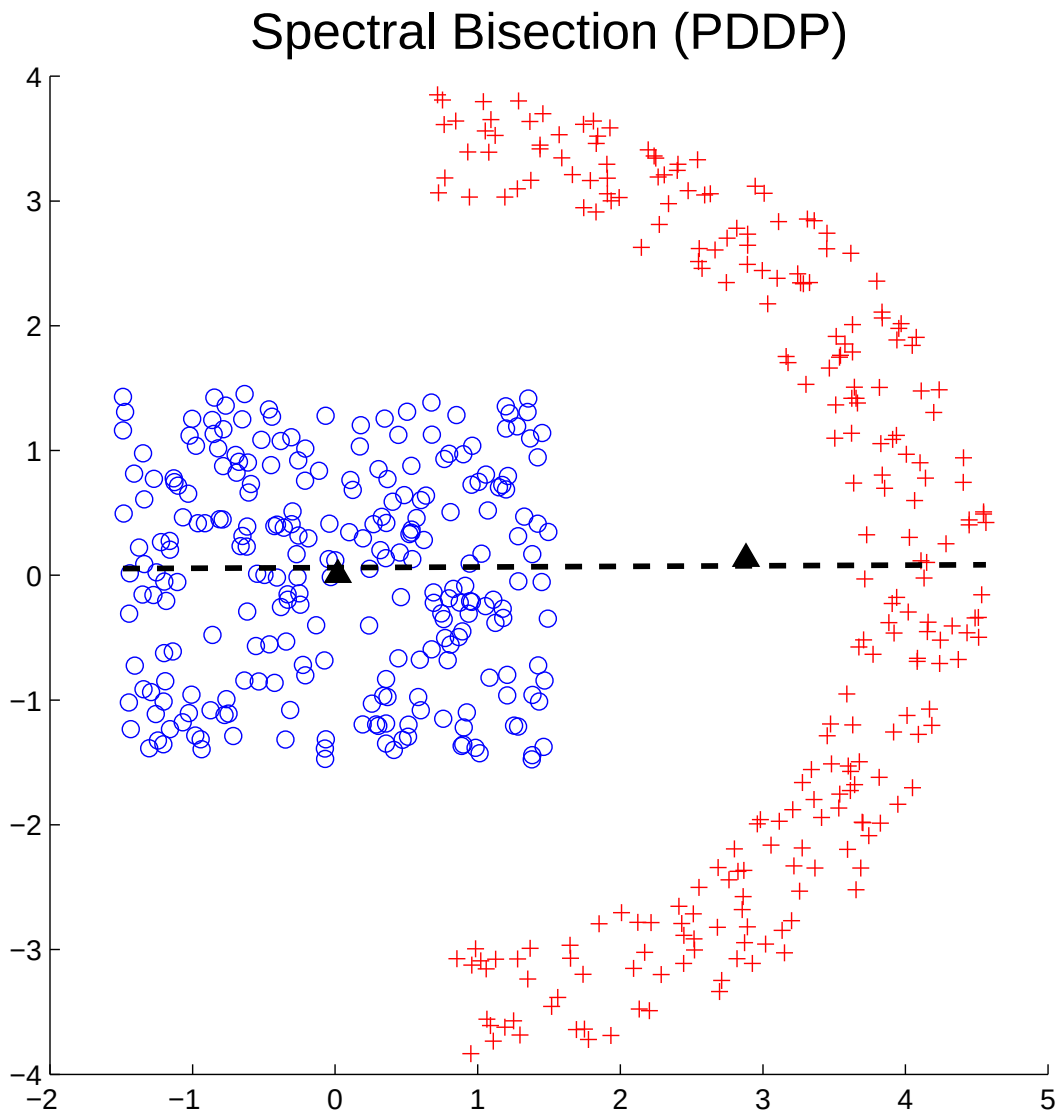
➤ Note: The world in non-linear. Often this is combined with Kernels – amounts to changing the inner product

Linear classifiers and Fisher's LDA

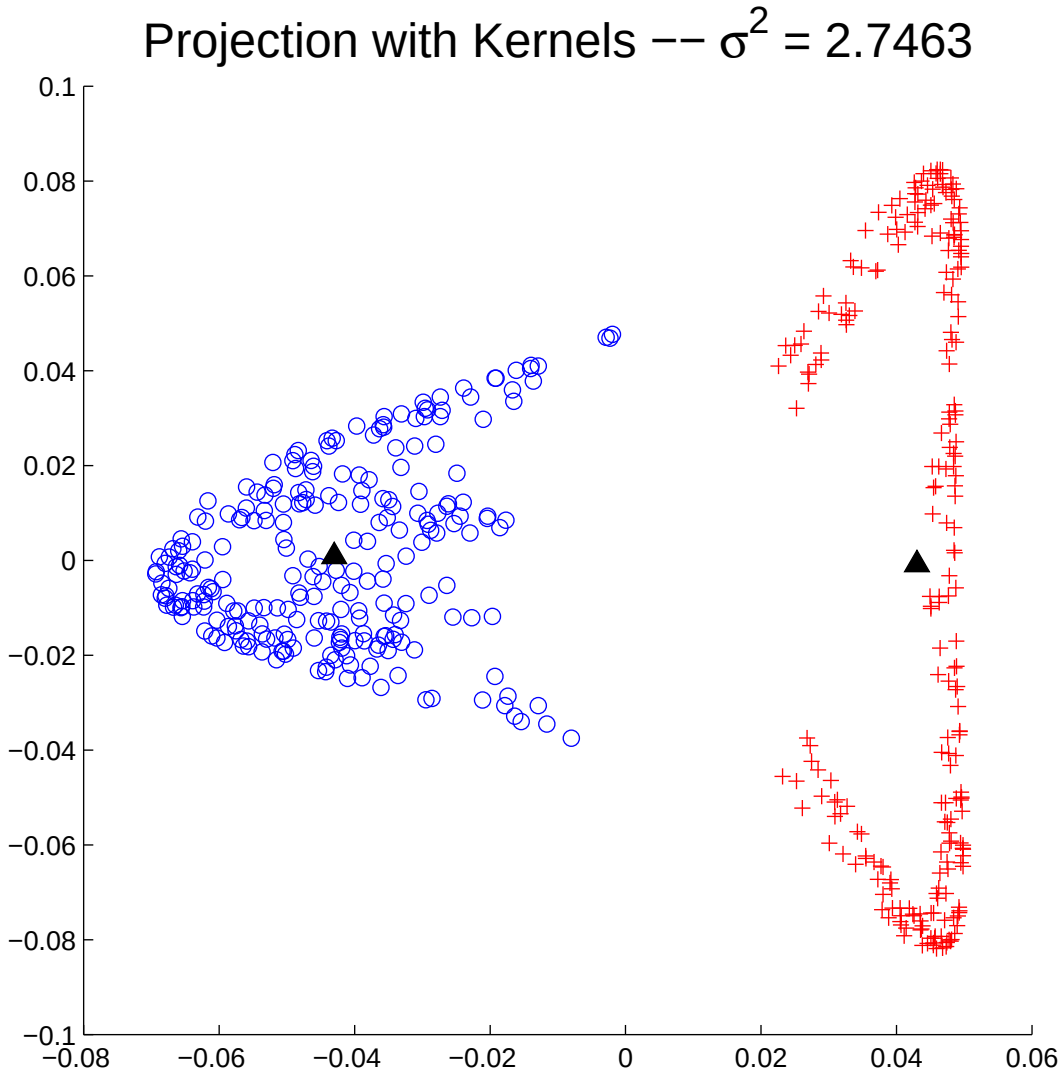
➤ Idea for two classes: Find a hyperplane which best separates the data in classes A and B.



A harder case:



- Use kernels to transform



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

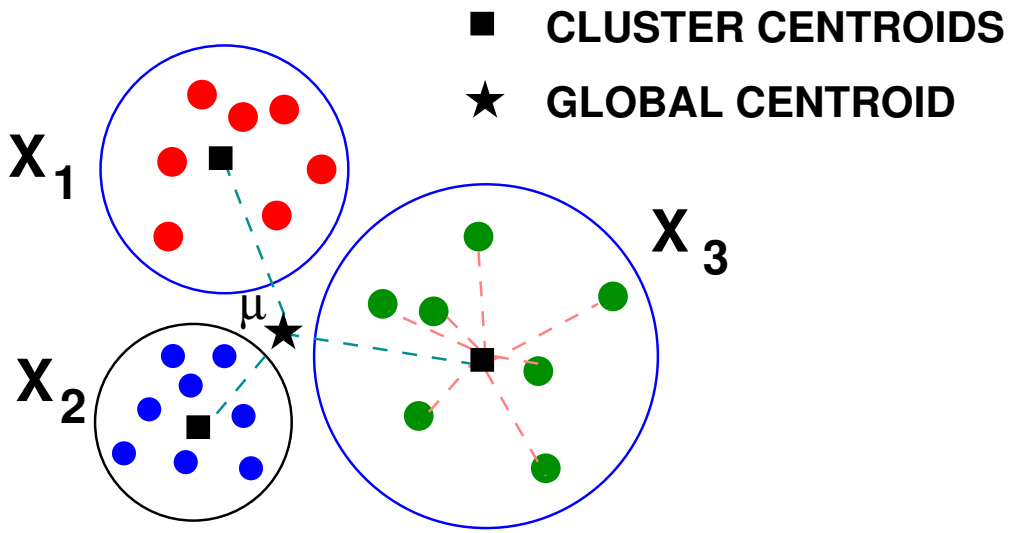
Define:

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

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

Where:

- μ = 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 μ_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:

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

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

➤ Consider the function:

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

➤ Call $V(\rho)$ the maximizer for an arbitrary given ρ.

➤ 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:

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

Γ is a smooth curve containing the p eivenvalues of interest

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

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

➤ With this, can prove :

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

Can now use Newton's method.

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

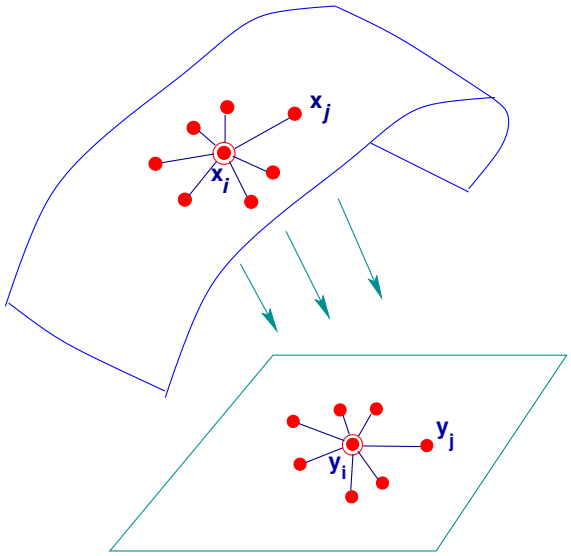
- Newton's method to find the zero of $f \equiv$ a fixed point iteration with $g(\rho) = \dfrac{\text{Tr}\,[V^T(\rho) A V(\rho)]}{\text{Tr}\,[V^T(\rho) B V(\rho)]},$
- Idea: Compute $V(\rho)$ by a Lanczos-type procedure
- Note: Standard problem - [not generalized] $\rightarrow$ inexpensive!
- See T. Ngo, M. Bellalij, and Y.S. 2010 for details

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

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

Example: The Laplacean eigenmaps approach

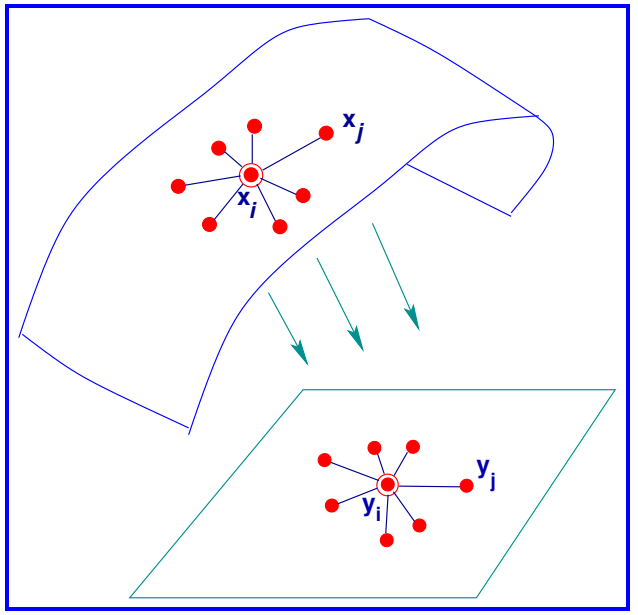
Laplacean Eigenmaps [Belkin-Niyogi '01] *minimizes*

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

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]



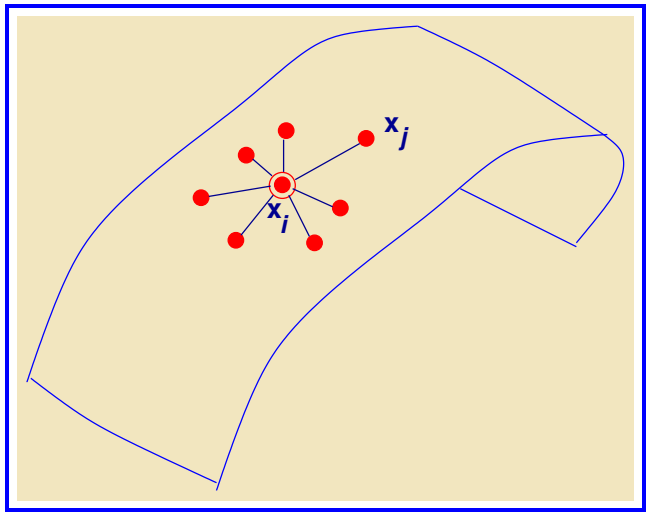
Locally Linear Embedding (Roweis-Saul-00)

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

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

$x_i \approx \Sigma w_{ij} x_j, \quad \sum_{j \in Adj(i)} w_{ij} = 1$

- Optimal weights computed ('local calculation') by minimizing $\|x_i - \Sigma w_{ij} x_j\|$ for $i = 1, \cdots, n$



- Mapped data (Y) computed by minimizing

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

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:

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

- In Eigenmaps and LLE we only know

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

- 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

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:

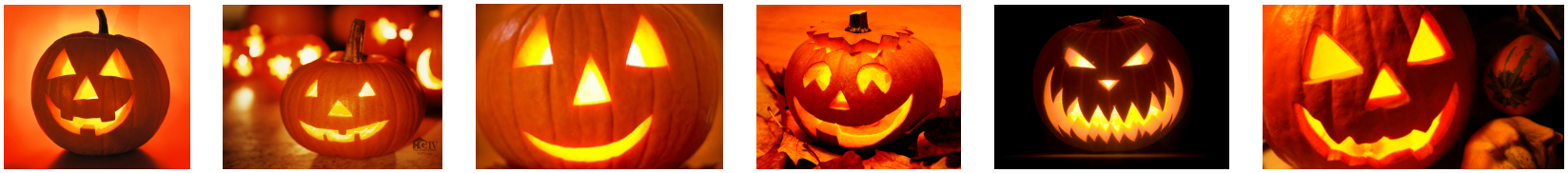
$$\min_{\boldsymbol{V}} \text{Tr} \left[\boldsymbol{V}^\top \boldsymbol{X} (\boldsymbol{I} - \boldsymbol{W}^\top)(\boldsymbol{I} - \boldsymbol{W}) \boldsymbol{X}^\top \boldsymbol{V} \right]$$

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

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

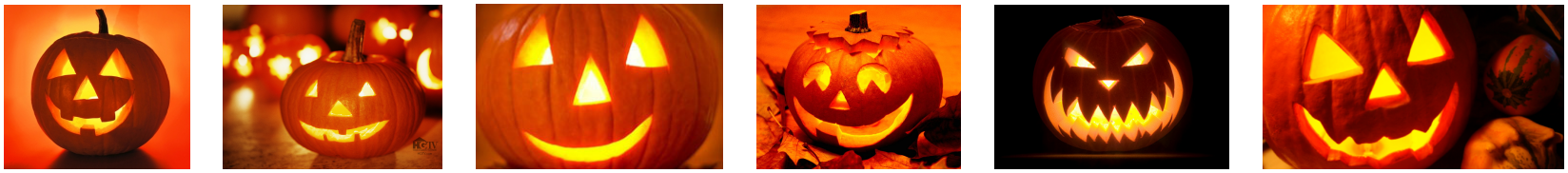
Face Recognition – background

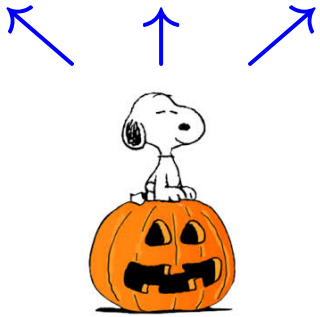
Problem: We are given a database of images: [arrays of pixel values]. And a test (new) image.



Face Recognition – background

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





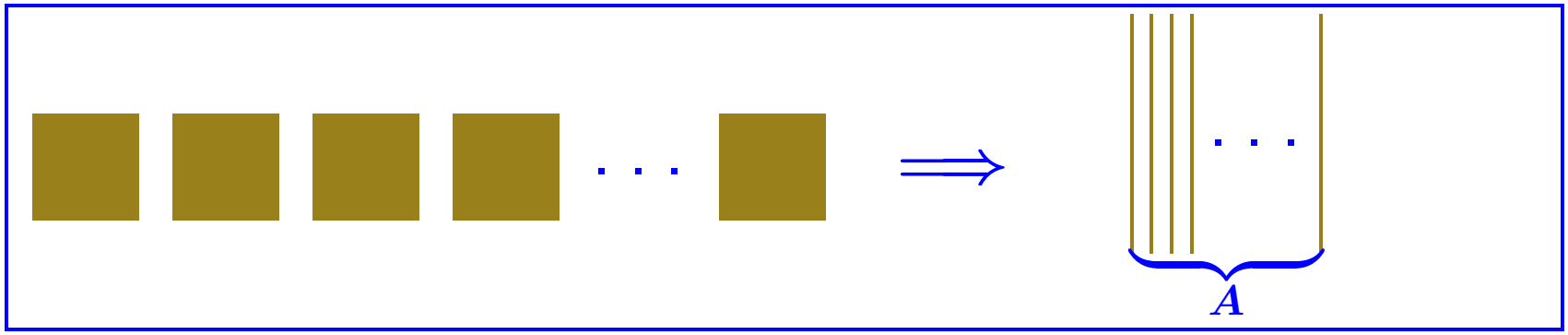
Question: Does this new image correspond to one of those in the database?

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

Graph-based methods in a supervised setting

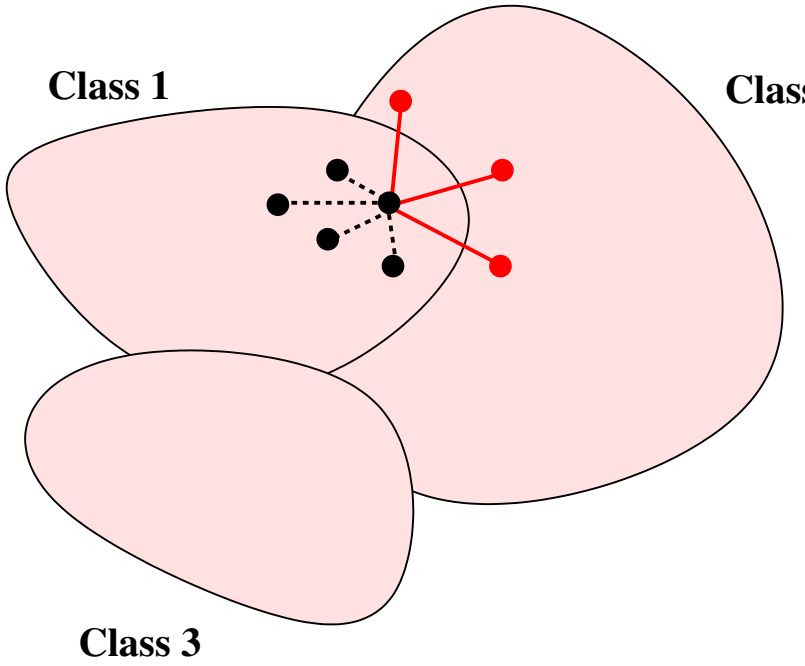
Graph-based methods can be adapted to supervised mode. Idea: Build G so that nodes in the same class are neighbors. If c = # classes, G consists of c cliques.

- Weight matrix W =block-diagonal
- Note: $\mathbf{rank}(W) = n - c$.
- 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]



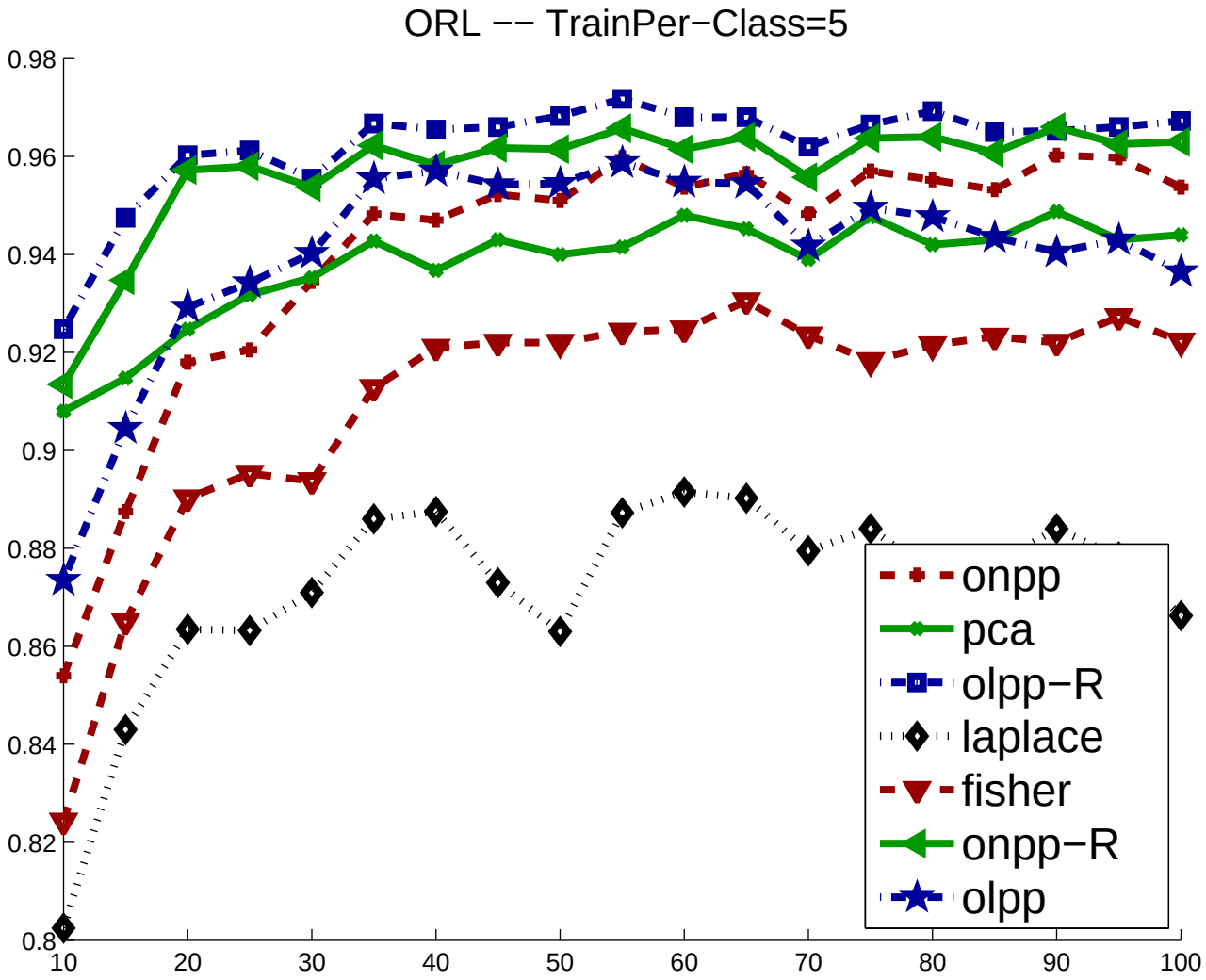
Leads to eigenvalue problem with matrix:

$$L_c - \rho L_R$$

- 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

LINEAR ALGEBRA METHODS: EXAMPLES

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.

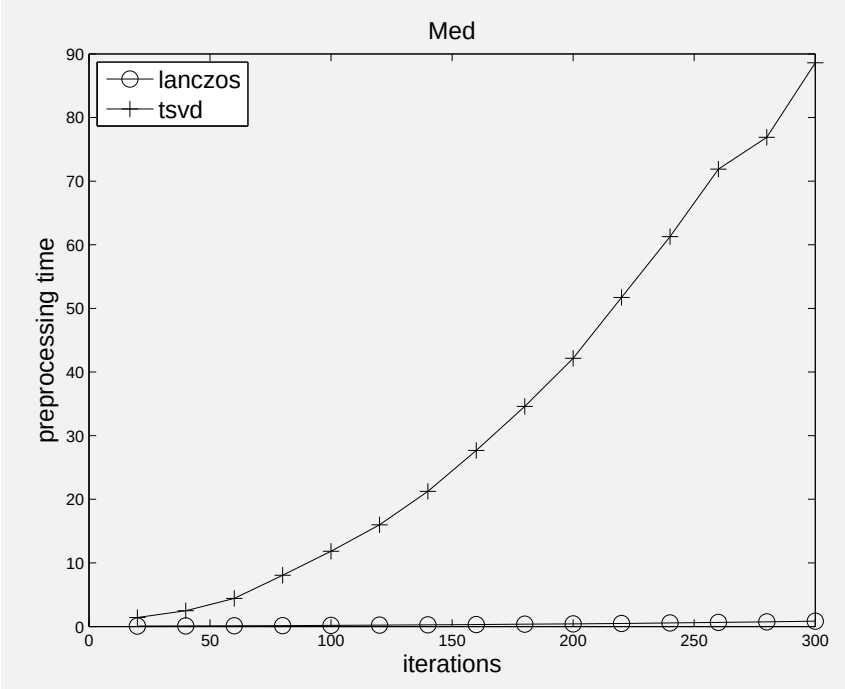
Tests: IR

Information retrieval datasets

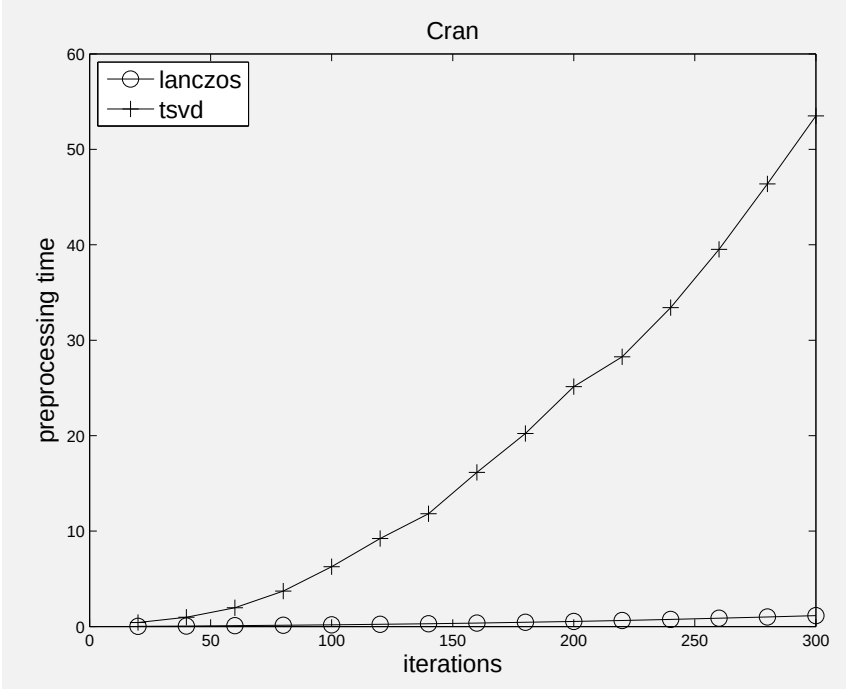
	# Terms	# Docs	# queries	sparsity
MED	7,014	1,033	30	0.735
CRAN	3,763	1,398	225	1.412

Preprocessing times

Med dataset.

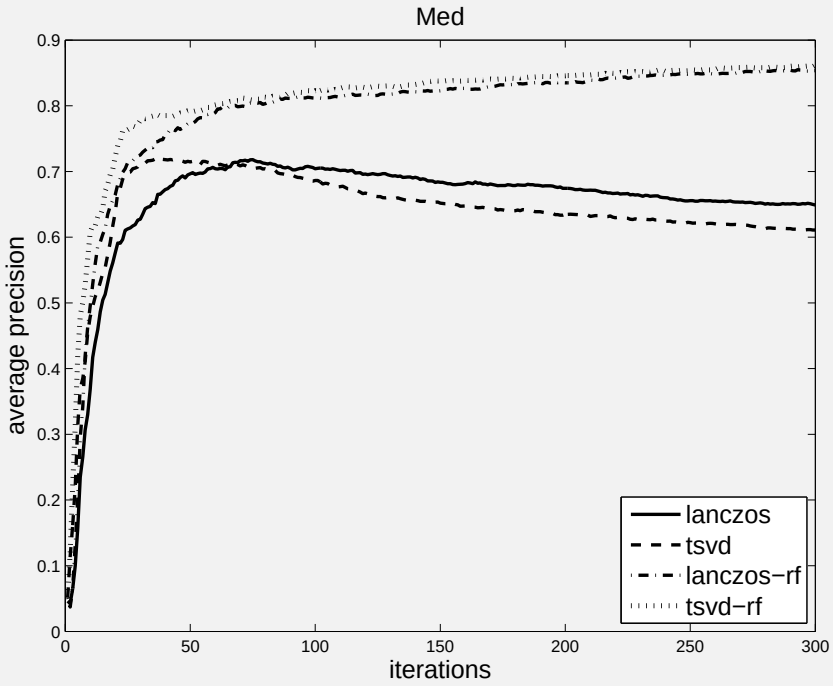


Cran dataset.

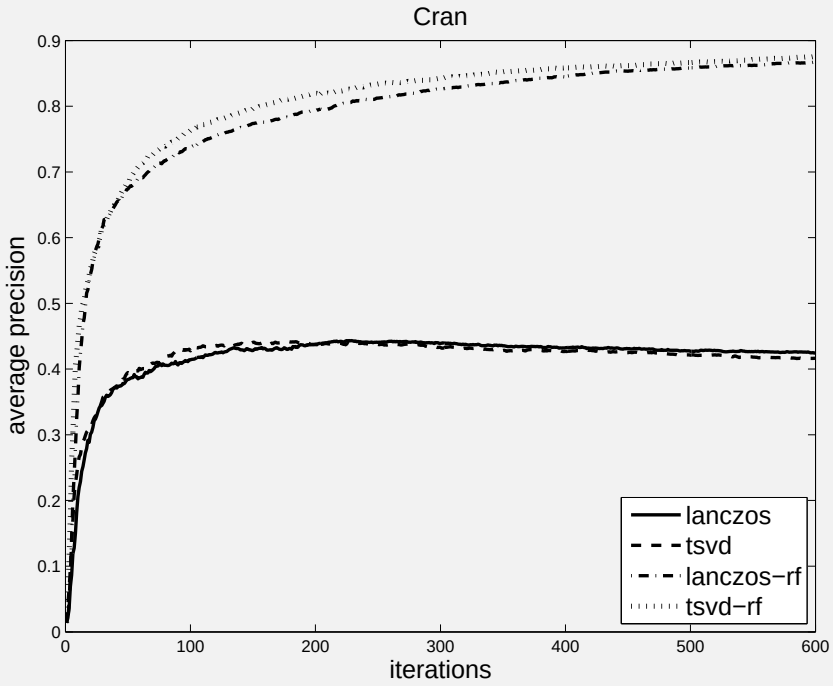


Average retrieval precision

Med dataset



Cran dataset



Retrieval precision comparisons

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

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

Problem

Given the partial SVD of $\boldsymbol{X}$, how to get a partial SVD of $\boldsymbol{X}_{new}$

- Will illustrate only with update of the form $\boldsymbol{X}_{new} = [\boldsymbol{X}, \boldsymbol{D}]$ (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:

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

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

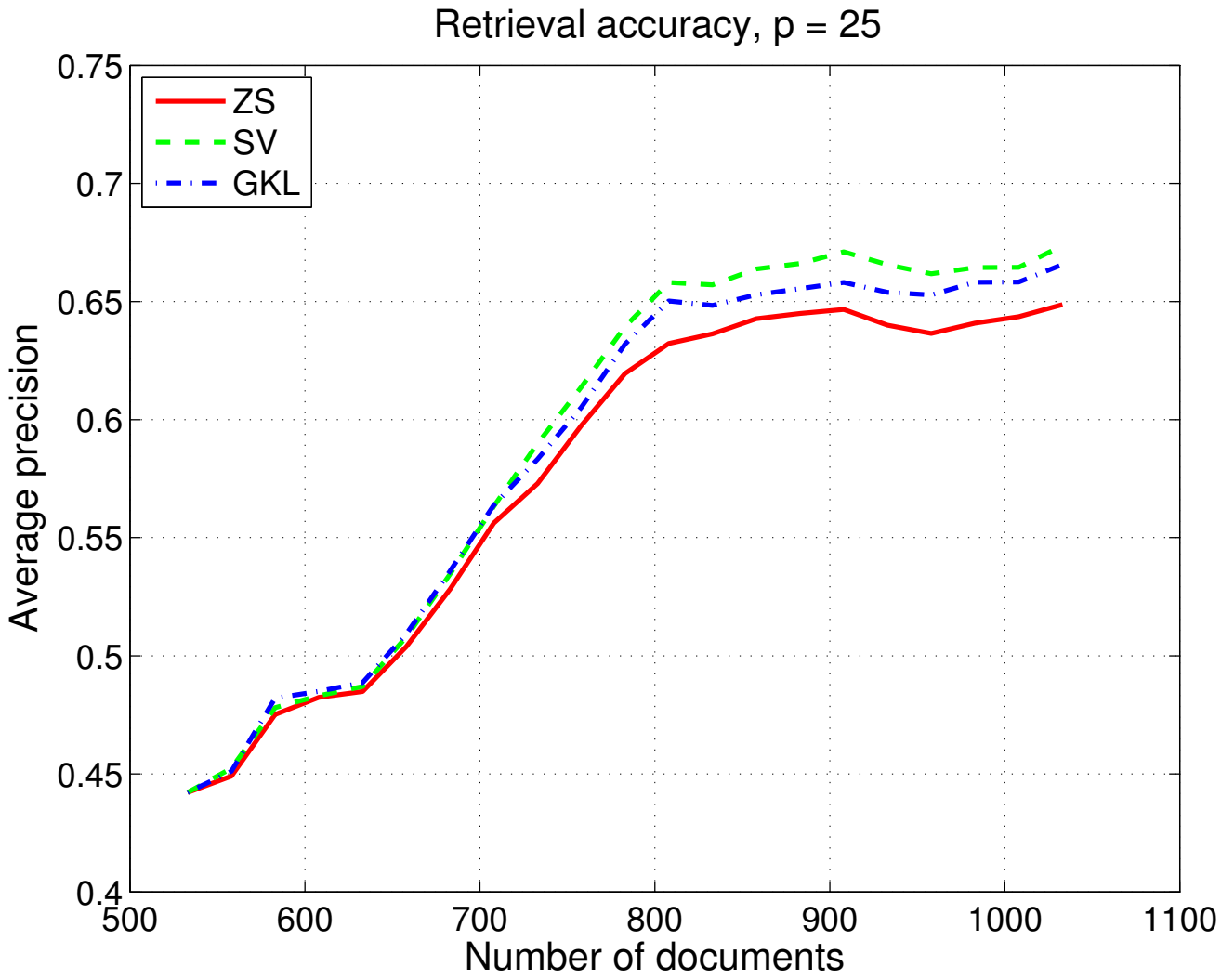
- Zha–Simon ('99): Compute the SVD of H_D & get approximate SVD from above equation
- It turns out this is a Rayleigh-Ritz projection method for the SVD [E. Vecharynski & YS 2013]
- Can show optimality properties as a result

Updating the SVD

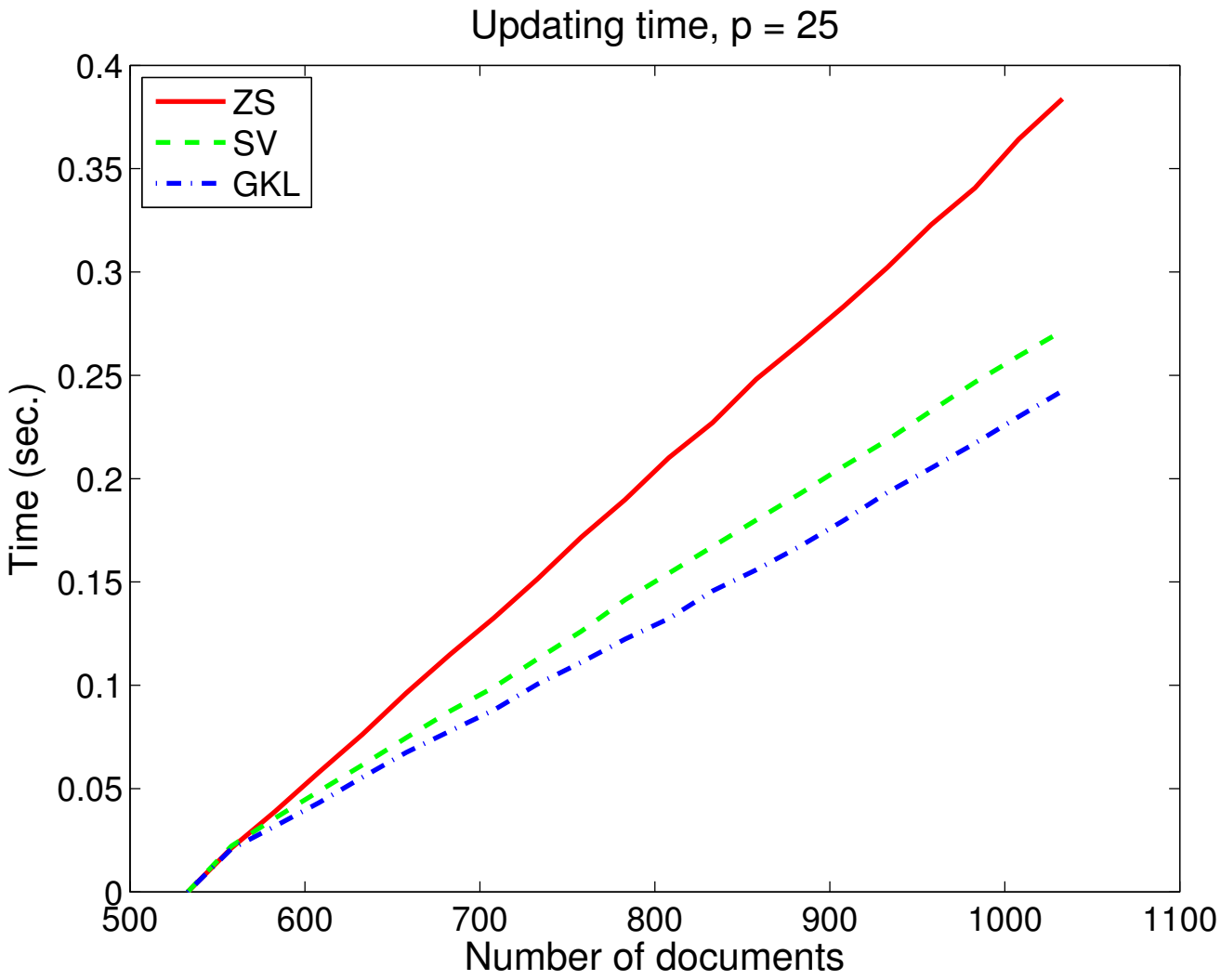
- When the number of updates is large this becomes costly.
- Idea: Replace $\hat{U}_p$ by a low dimensional approximation:
- Use $\bar{U}$ of the form $\bar{U} = [U_k, Z_l]$ instead of $\bar{U} = [U_k, \hat{U}_p]$
- Z_l must capture the range of $D_k = (I - U_k U_k^T) D$
- Simplest idea : best rank–l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

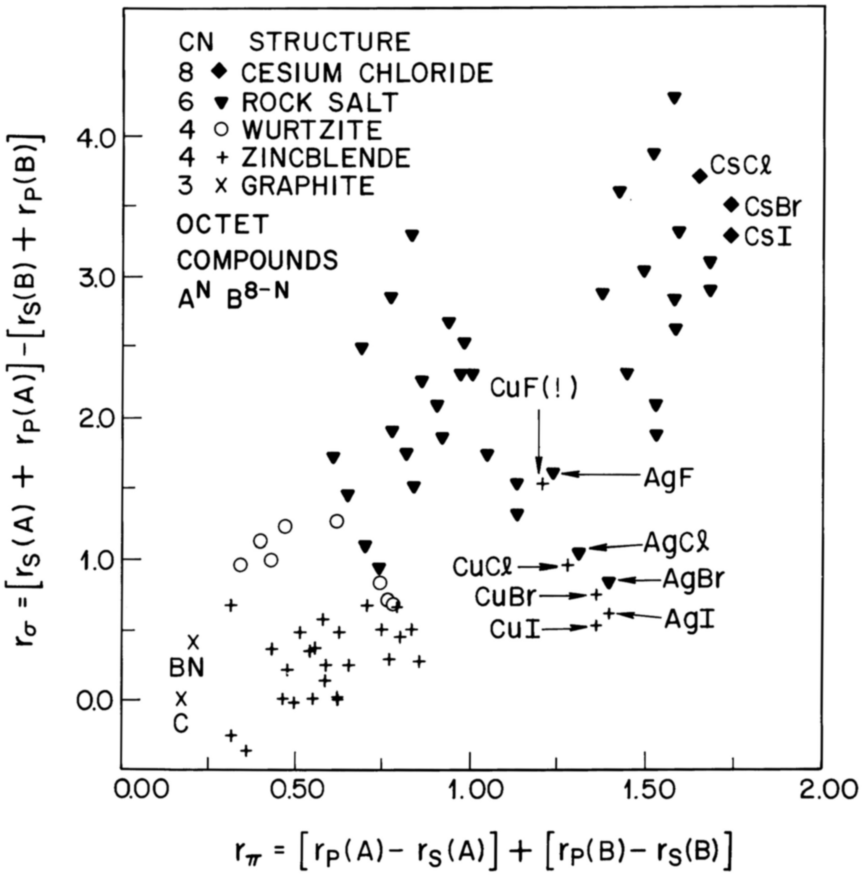
Issue 2: databases, and more generally sharing, not too common in materials

> *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: Unsupervised learning "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

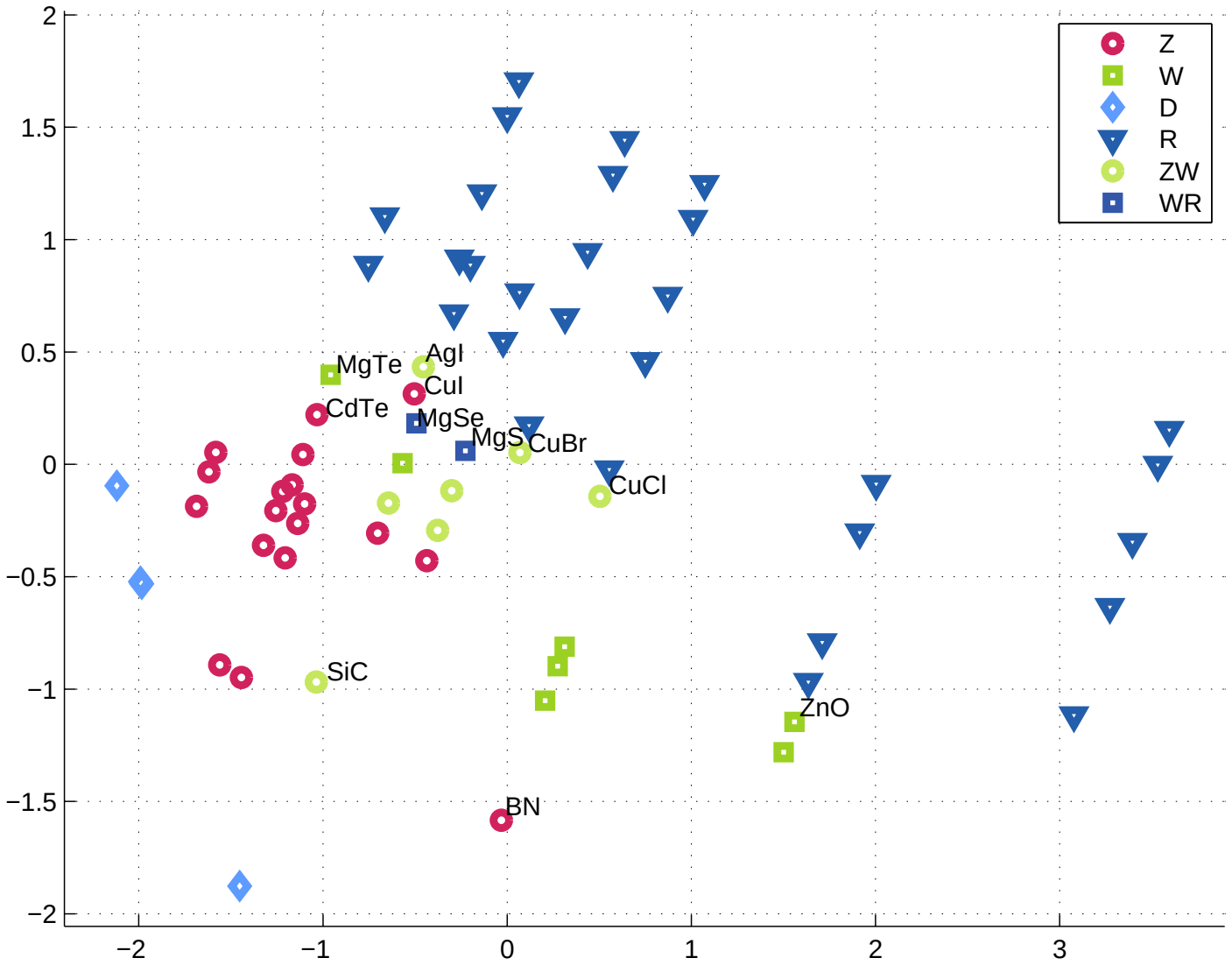
see: J. R. Chelikowsky, J. C. Phillips, Phys Rev. B 19 (1978).

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:


Z
W
D
R
ZW
WR
MgTe
AgI
CuI
CdTe
MgSe
MgS
CuBr
CuCl
SiC
ZnO
BN
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1
0
1
2
3

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 –

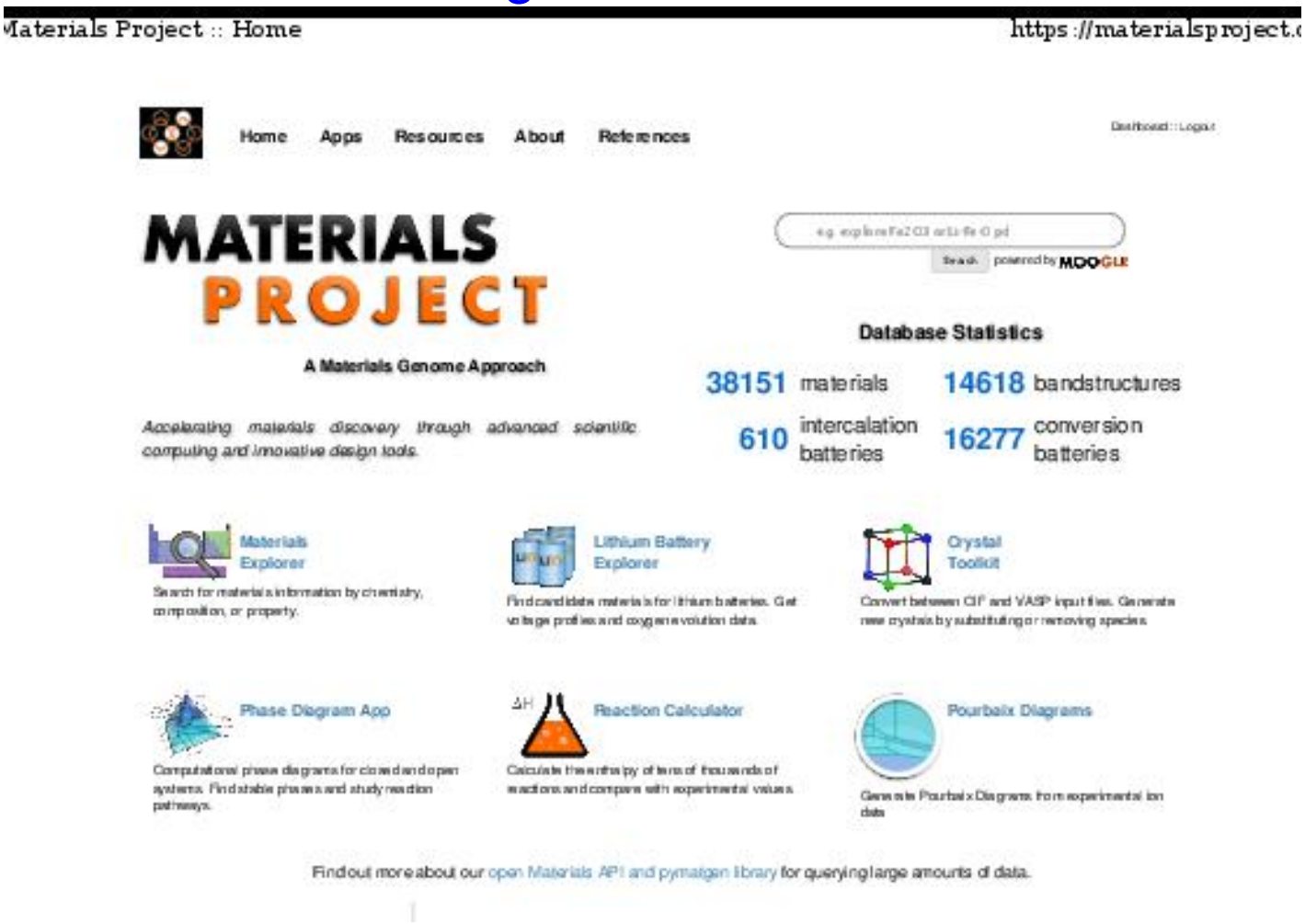
3. Orthogonal Neighborhood Preserving Projection (ONPP) - a graph based method - [see Kokiopoulou, YS, 2005]

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

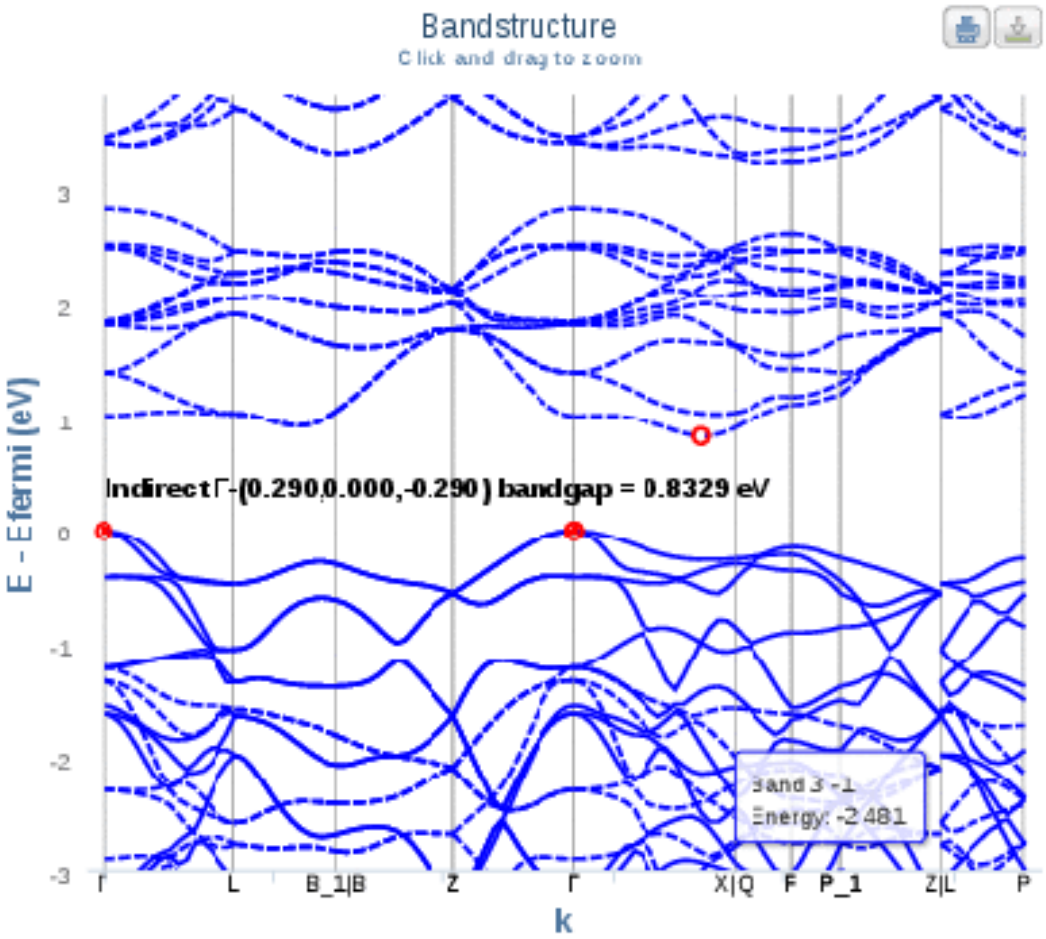
Recent work

- Some data is becoming available

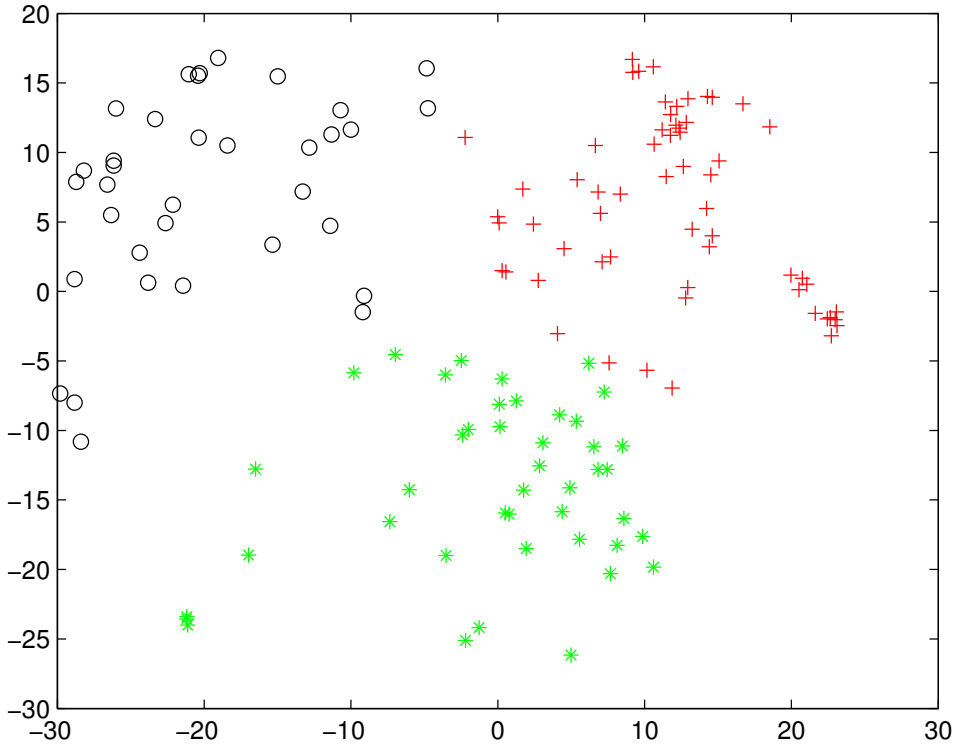


Recent work

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



- Work in progress
- 3-way clustering obtained with dim. reduction + k-means $\longrightarrow$
- 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
- 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 !