UNIVERSITY OF MINNESOTA TWIN CITIES

From data-mining to nanotechnology and back: The new problems of numerical linear algebra

Yousef Saad

Department of Computer Science and Engineering

University of Minnesota

Computer Science Colloquium

Nov. 9th, 2009

Introduction

Numerical linear algebra has always been a "universal" tool in science and engineering. Its focus has changed over the years to tackle "new challenges"

1940s–1950s: Major issue: the flutter problem in aerospace engineering. Focus: eigenvalue problem.

➤ Then came the discoveries of the LR and QR algorithms, and the package Eispack followed a little later

1960s: Problems related to the power grid promoted what we know today as general sparse matrix techniques.

Late 1980s – 1990s: Focus on parallel matrix computations.

Late 1990s: Big spur of interest in "financial computing" [Focus: Stochastic PDEs]

- Then the stock market tanked .. and numerical analysts returned to more mundaine basics

Recent/Current: Google page rank, data mining, problems related to internet technology, knowledge discovery, bio-informatics, nano-technology, ...

- Major new factor: Synergy between disciplines.

Example: Discoveries in materials (e.g. semi conductors replacing vacuum tubes in 1950s) lead to faster computers, which in turn lead to better physical simulations..

- What about data-mining and materials?
- Potential for a perfect union...

NUMERICAL LINEAR ALGEBRA IN ELECTRONIC STRUCTURE

Electronic structure

➤ Quantum mechanics: Single biggest scientific achievement of 20th century. See

Thirty years that shook physics, George Gamov, 1966

➤ "There is plenty of room at the bottom" [Feynman, 1959] – started the era of nanotechnology [making materials with specific properties.]

see *http://www.zyvex.com/nanotech/feynman.html*

➤ Examples of applications: Superhard materials, superconductors, drug design, efficient photovoltaic cells, ...

➤ All these need as a basic tool: computational methods to determine electronic structure.

Electronic structure and Schrödinger's equation

- Basic and essential tool for simulations at the nano-scale.
- Determining matter's electronic structure can be a major challenge:

  Number of particules is large [a macroscopic amount contains $\approx 10^{23}$ electrons and nuclei] and the physical problem is intrinsically complex.

- Solution via the many-body Schrödinger equation:

$$H\Psi = E\Psi$$

- In its original form the above equation is very complex

- Hamiltonian H is of the form :

$$H = -\sum_i \frac{\hbar^2 \nabla_i^2}{2M_i} - \sum_j \frac{\hbar^2 \nabla_j^2}{2m} + \frac{1}{2}\sum_{i,j} \frac{Z_i Z_j e^2}{|\vec{R}_i - \vec{R}_j|} - \sum_{i,j} \frac{Z_i e^2}{|\vec{R}_i - \vec{r}_j|} + \frac{1}{2}\sum_{i,j} \frac{e^2}{|\vec{r}_i - \vec{r}_j|}$$

- $\Psi = \Psi(r_1, r_2, \ldots, r_n, R_1, R_2, \ldots, R_N)$ depends on coordinates of all electrons/nuclei.
- Involves sums over all electrons / nuclei and their pairs
- Note: $\nabla_i^2 \Psi$ is Laplacean of Ψ w.r.t. variable r_i. Represents kinetic energy for i-th particle.

A hypothetical calculation: [with a "naive approach"]

- 10 Atoms each having 14 electrons [Silicon]
- ... a total of 15*10= 150 particles
- ... Assume each coordinate will need 100 points for discretization..
- ... you will get

$$\#\text{ Unknowns } = \underbrace{100}_{part.1} \times \underbrace{100}_{part.2} \times \cdots \times \underbrace{100}_{part.150} = 100^{150}$$

- Methods based on this basic formulation are limited to a few atoms – useless for real compounds.

The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to the explanation of the main features of complex atomic systems without too much computations. Dirac, 1929

- In 1929 quantum mechanics was basically understood
- Today the "desire" to have approximate practical methods is still alive

Several approximations/theories used

- Born-Oppenheimer approximation: Neglect motion of nuclei [Much heavier than electrons]
- Replace many electrons by one electron systems: each electron sees only average potentials from other particles
- Density Functional Theory [Hohenberg-Kohn '65]: Observables determined by ground state charge density
- Consequence: Equation of the form

$$\left[-\frac{h^2}{2m}\nabla^2 + v_0(r) + V_H + V_{xc}\right]\Psi = E\Psi$$

- v_0 = external potential, E_{xc} = exchange-correlation energy

Kohn-Sham equations → nonlinear eigenvalue Pb

$$\left[-\frac{1}{2}\nabla^2 + (V_{ion} + V_H + V_{xc})\right] \Psi_i = E_i \Psi_i, i = 1, ..., n_o$$

$$\rho(r) = \sum_i^{n_o} |\Psi_i(r)|^2$$

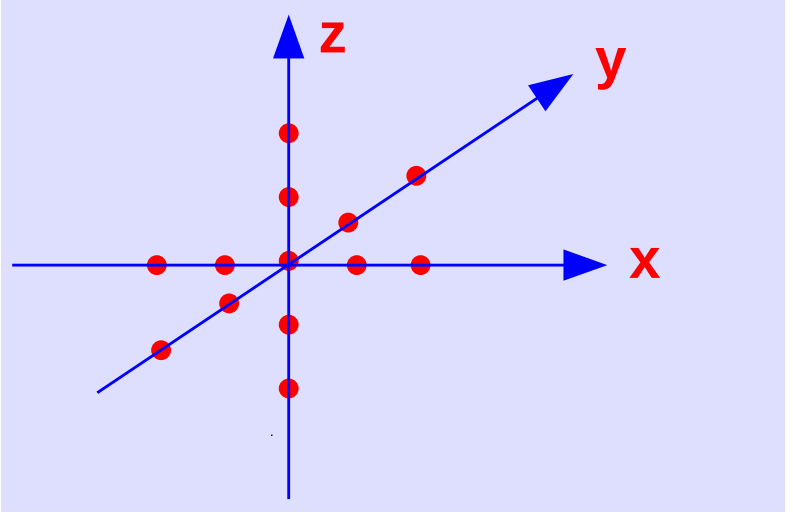
$$\nabla^2 V_H = -4\pi\rho(r)$$

- Both V_{xc} and V_H, depend on ρ.
- Potentials & charge densities must be self-consistent.
- Broyden-type quasi-Newton technique used
- Typically, a small number of iterations are required
- Most time-consuming part: diagonalization

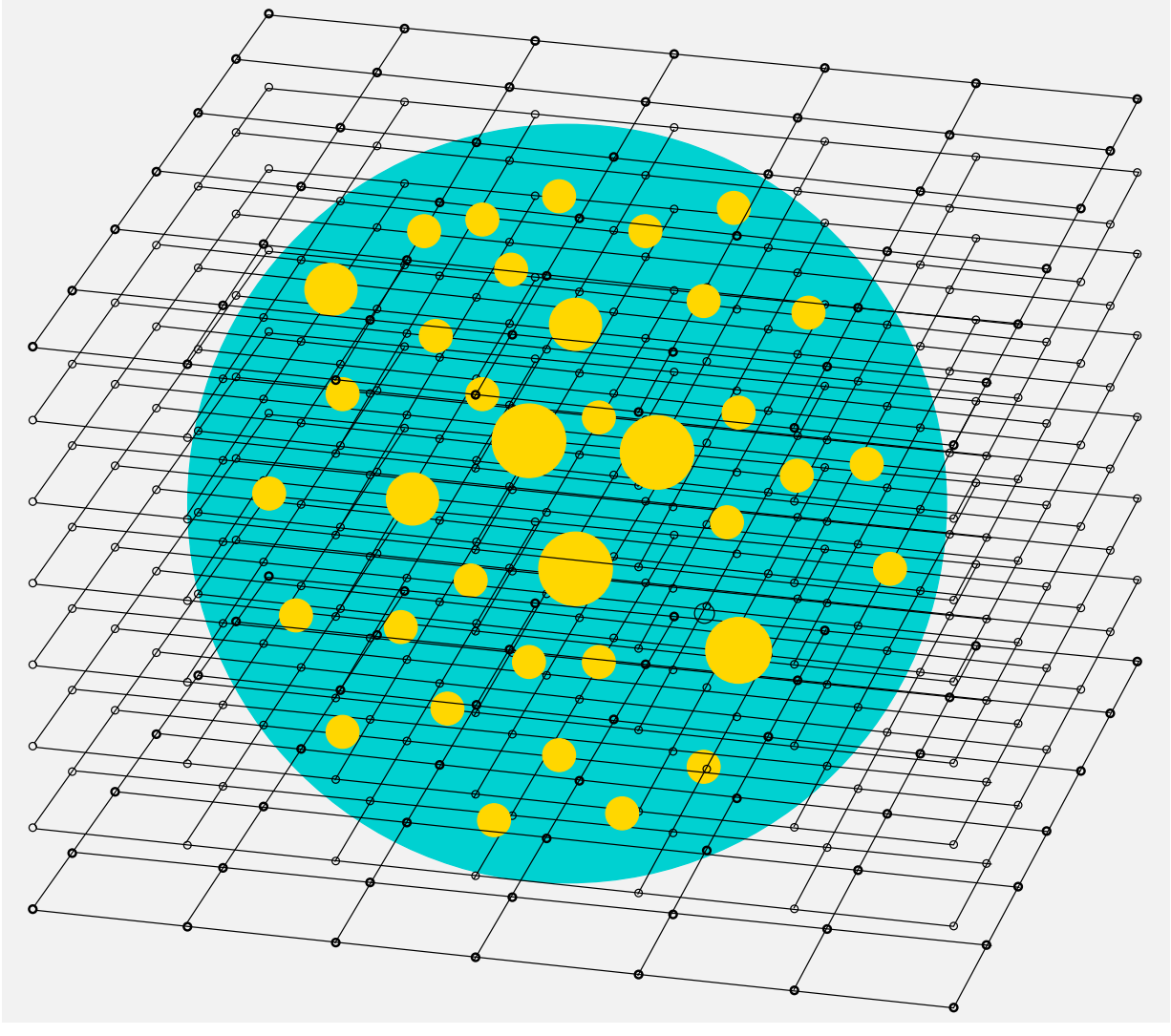
Real-space Finite Difference Methods

- Use High-Order Finite Difference Methods [Fornberg & Sloan '94]
- Typical Geometry = Cube – regular structure.
- Laplacean matrix need not even be stored.

Order 4 Finite Difference Approximation:



The physical domain



Computational code: PARSEC; Milestones

- *PARSEC* = Pseudopotential Algorithm for Real Space Electronic Calculations
- Sequential real-space code on Cray YMP [up to '93]
- Cluster of SGI workstations [up to '96]
- CM5 ['94-'96] Massive parallelism begins
- IBM SP2 [Using PVM]
- Cray T3D [PVM + MPI] $\sim$ '96; Cray T3E [MPI] – '97
- IBM SP with +256 nodes – '98+
- SGI Origin 3900 [128 processors] – '99+
- IBM SP + F90 - PARSEC name given, '02
- PARSEC released in $\sim$ 2005.

CHEBYSHEV FILTERING

Diagonalization via Chebyshev filtering

Given a basis $[v_1, \ldots, v_m]$, 'filter' each vector as

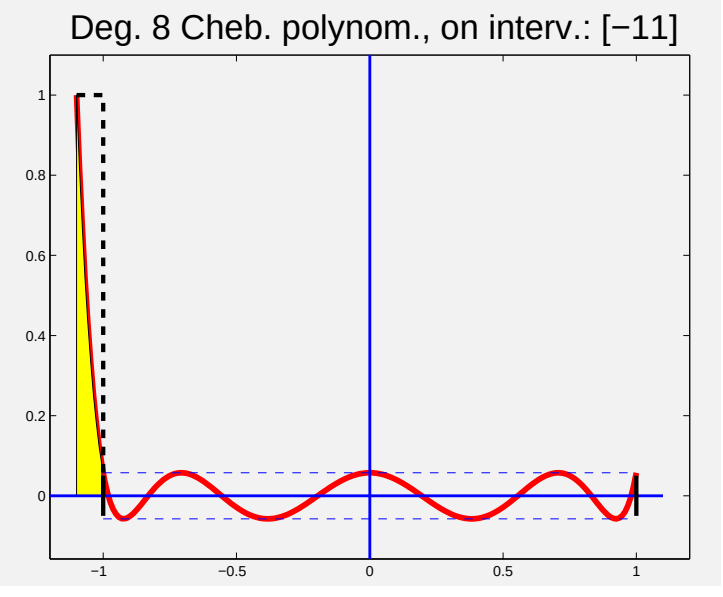
$$\hat{v}_i = p_k(A) v_i$$

➤ p_k = Low deg. polynomial. Dampens unwanted components

➤ Filtering step not used to compute eigenvectors accurately

➤ In effect: SCF & diagonalization loops merged

Convergence well preserved



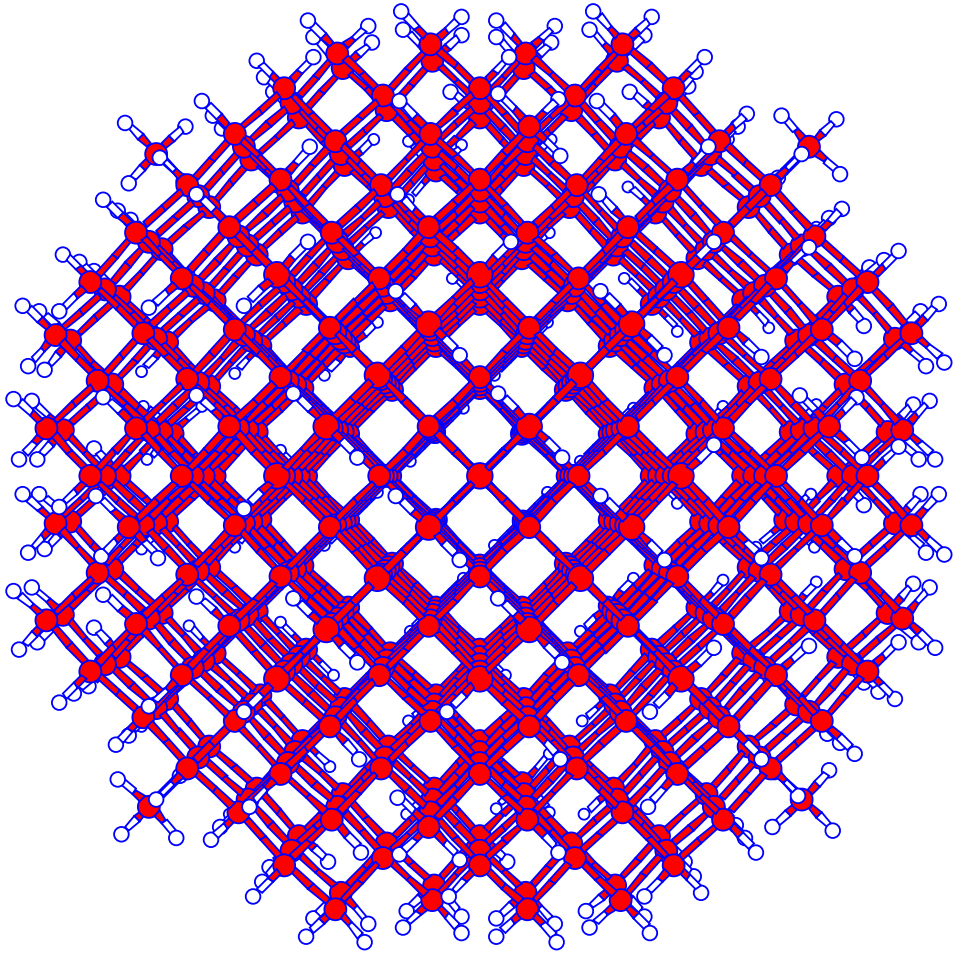
Yunkai Zhou, Y.S., Murilo L. Tiago, and James R. Chelikowsky, Phys. Rev. E, vol. 74, p. 066704 (2006).

Chebyshev Subspace iteration - experiments

- A large calculation: $Si_{9041}H_{1860}$, using 48 processors [SGI Altix, Itanium proc., 1.6 GHz]
- Hamiltonian size=$2,992,832$, Num. States= $19,015$.

$\#\ A*x$	# SCF	$total\_eV/atom$	1st CPU	total CPU
4804488	18	-92.00412	102.12 hrs.	294.36 hrs

- Calculation done in $\sim$ 2006.
- Largest one done in 1997: $Si_{525}H_{276}$
- Took a few days [48 h. cpu] on 64PE - Cray T3D.
- Now 2 hours on 1 PE (!)



NUMERICAL LINEAR ALGEBRA IN DATA MINING

Introduction: What is data mining?

➤ 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: brief overview with emphasis on dimension reduction techniques. interrelations between techniques, and graph theory tools.

Major tool of Data Mining: Dimension reduction

➤ Goal is not just to reduce computational cost but to:

- Reduce noise and redundancy in data
- Discover ‘features’ or ‘patterns’ (e.g., supervised learning)

➤ Techniques depend on application: Preserve angles? Preserve distances? Maximize variance? ..

The problem of Dimension Reduction

- Given $d \ll m$ find a mapping

$$\Phi : x \in \mathbb{R}^m \longrightarrow y \in \mathbb{R}^d$$

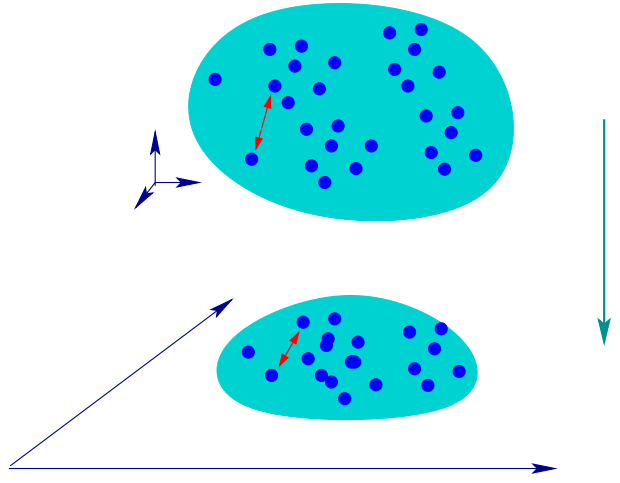
- Mapping may be explicit [typically linear], e.g.:

$$y = V^T x$$

- Or implicit (nonlinear)

Practically:

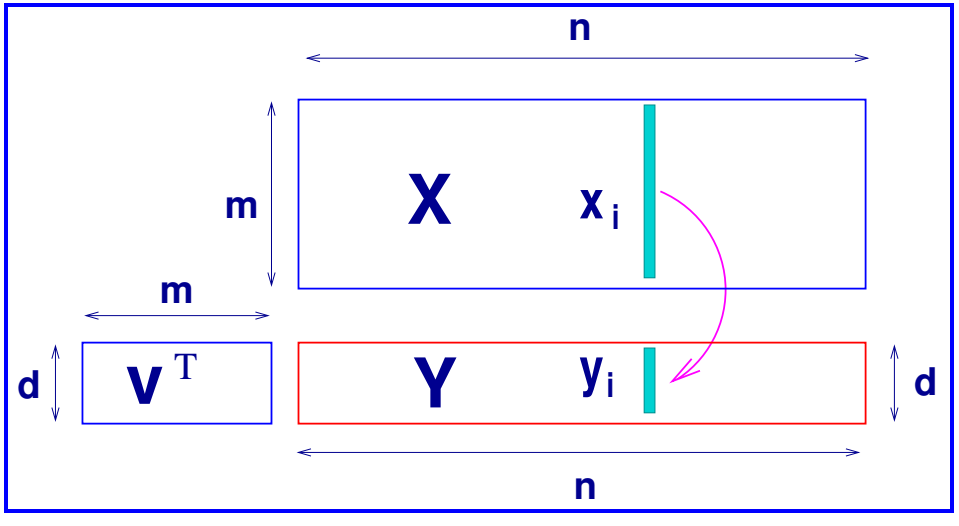
Given: $X \in \mathbb{R}^{m \times n}$.
Want: a low-dimensional representation $Y \in \mathbb{R}^{d \times n}$ of X



Linear Dimensionality Reduction

Given: a data set $X = [x_1, x_2, \ldots, x_n]$, and d the dimension of the desired reduced space $Y = [y_1, y_2, \ldots, y_n]$.

Want: A linear transformation from X to Y



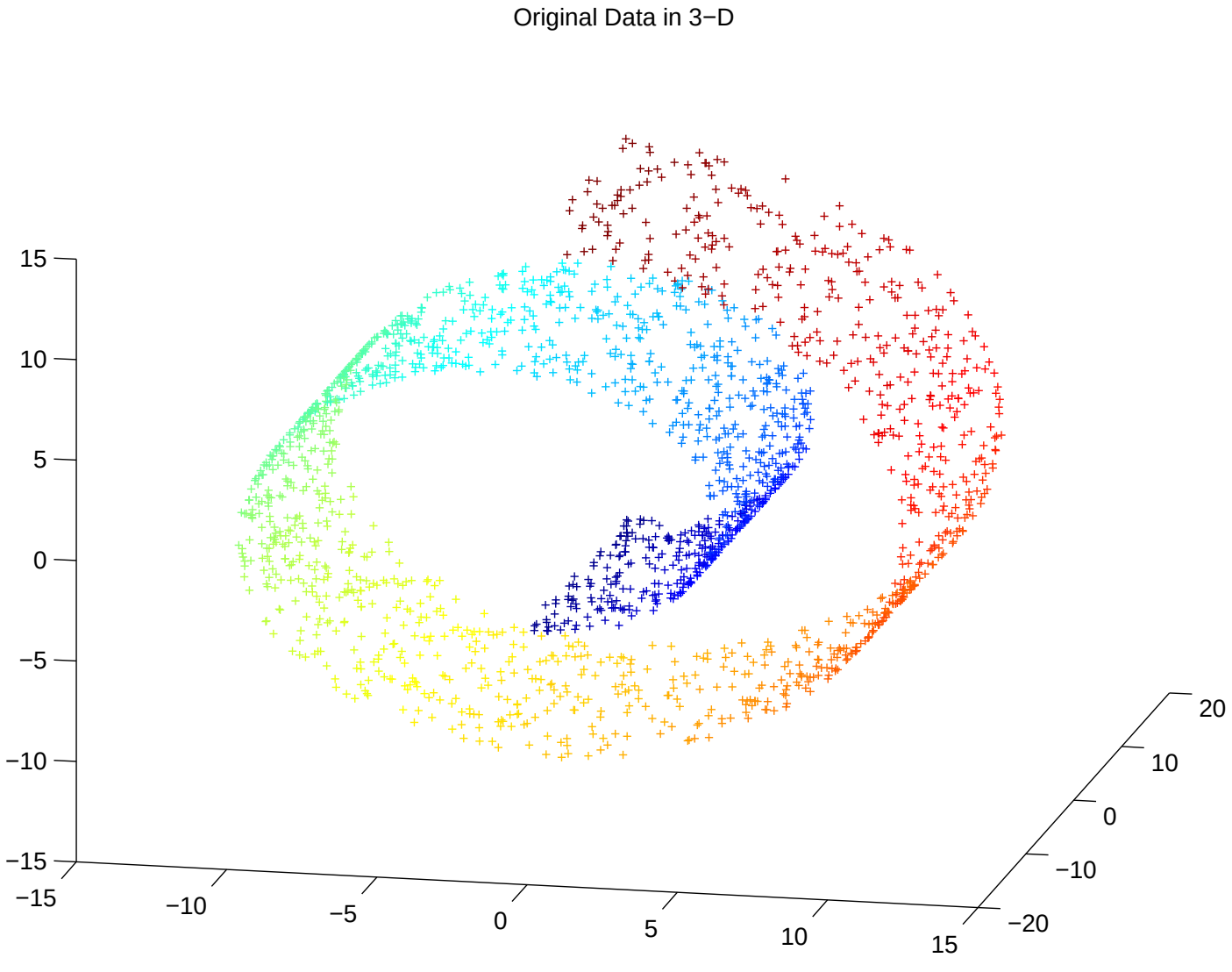
$$X \in \mathbb{R}^{m \times n}$$
$$V \in \mathbb{R}^{m \times d}$$
$$\boxed{Y = V^\top X}$$
$$\rightarrow \quad Y \quad \in \mathbb{R}^{d \times n}$$

- m-dimens. objects (x_i) 'flattened' to d-dimens. space (y_i)

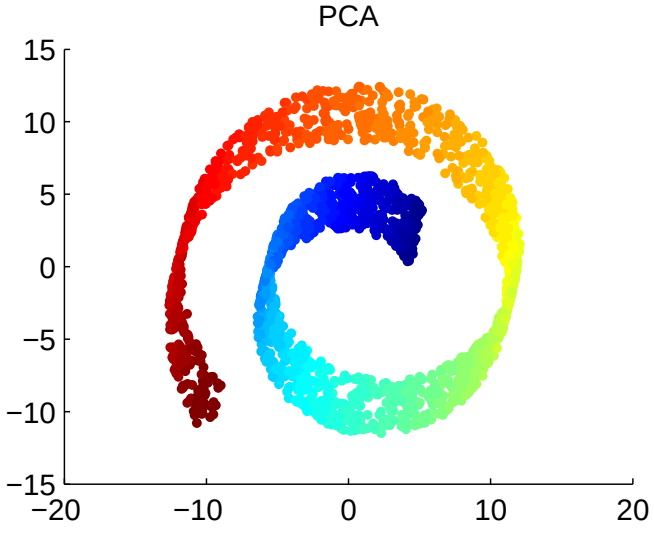
Constraint: The y_i's must satisfy certain properties

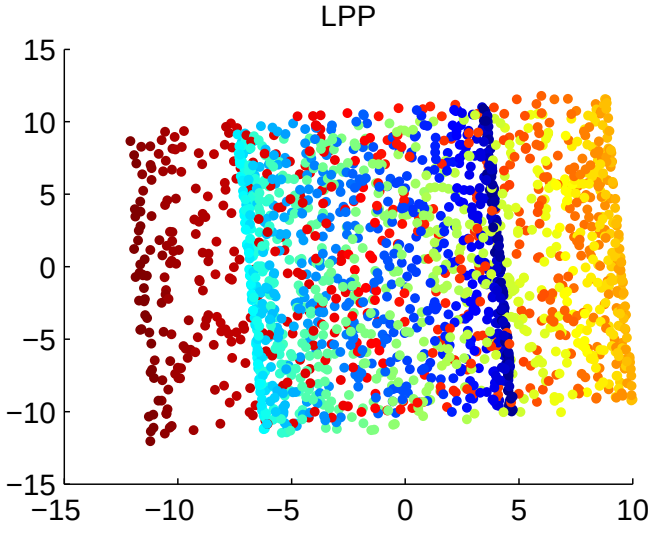
- Optimization problem

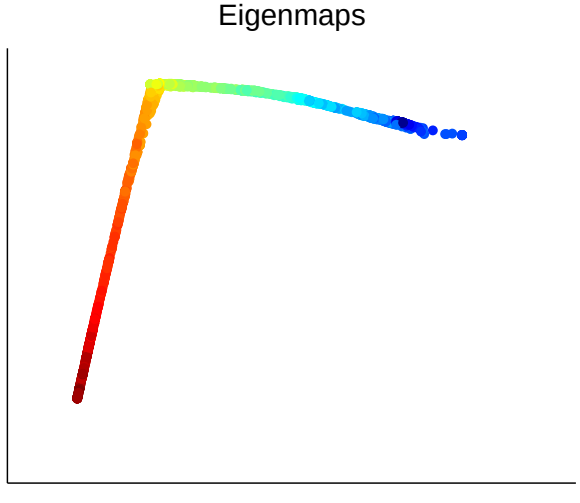
Example 1: The 'Swill-Roll' (2000 points in 3-D)

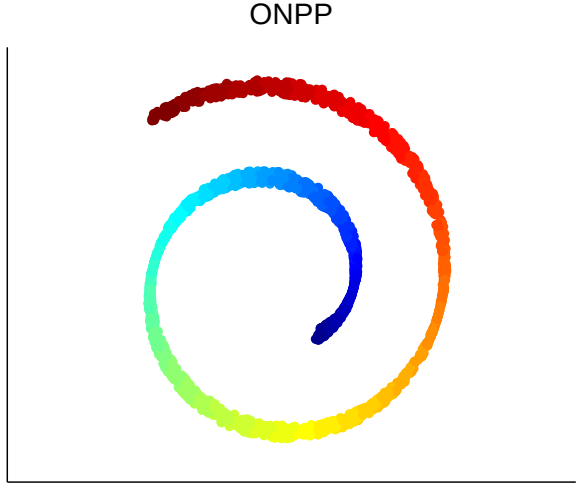


2-D ‘reductions’:

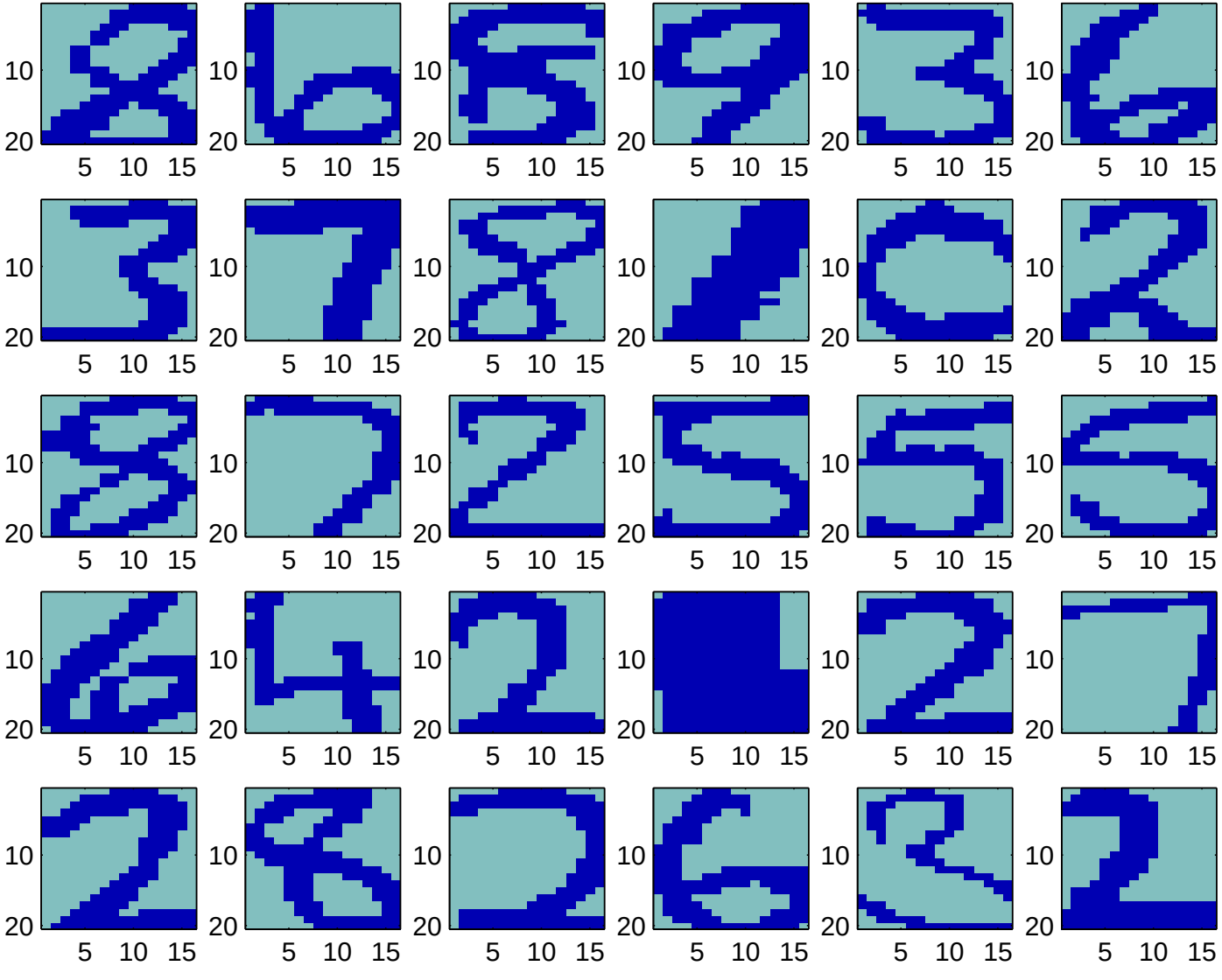




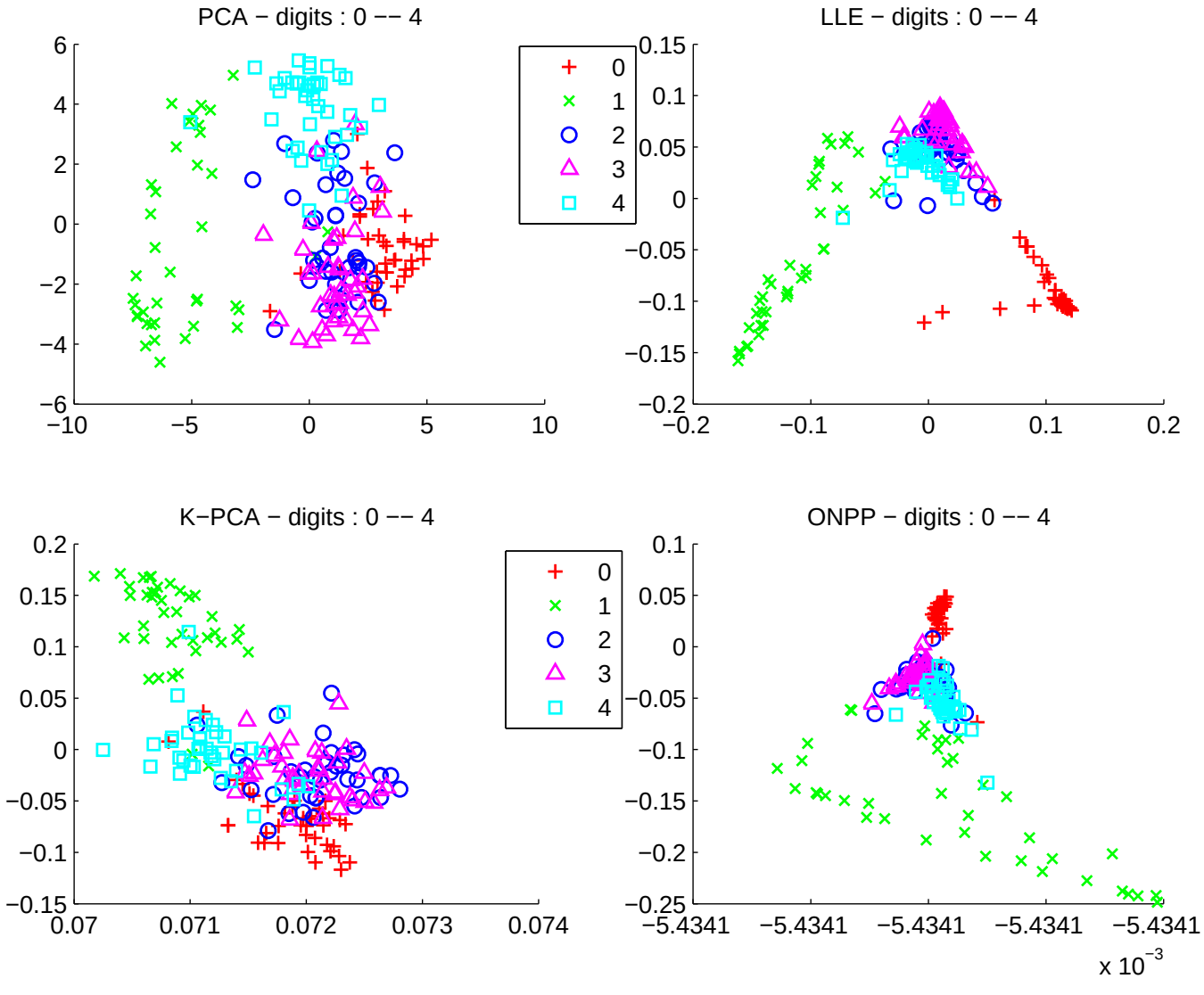




Example 2: Digit images (a random sample of 30)

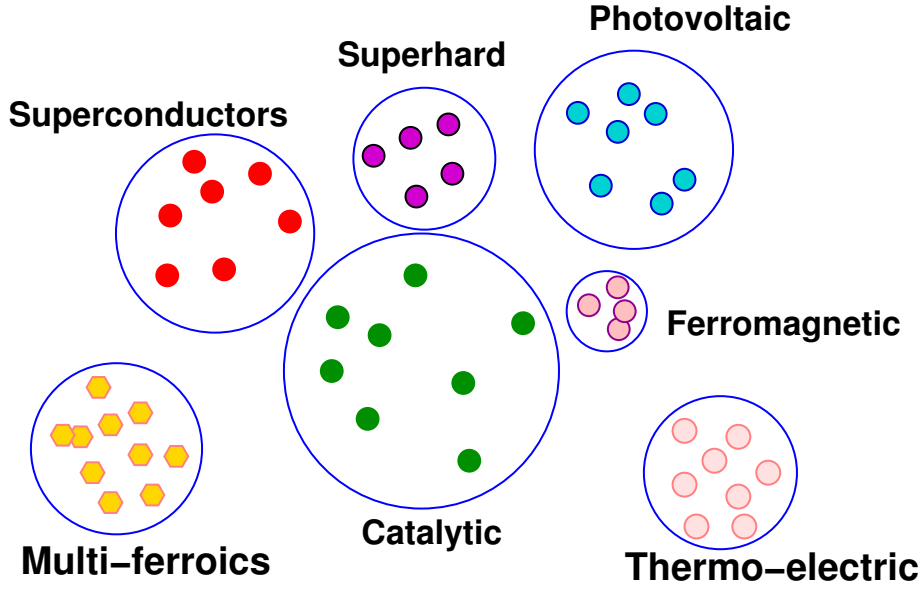


2-D 'reductions':



Unsupervised learning: Clustering

Problem: partition a given set into subsets such that items of the same subset are most similar and those of two different subsets most dissimilar.

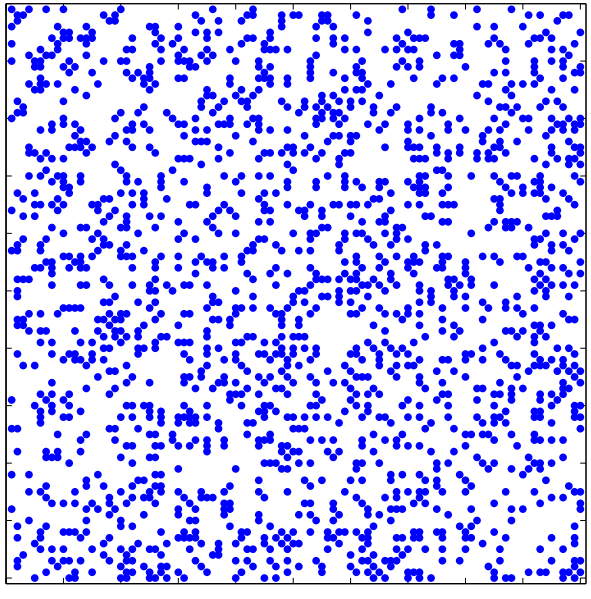


- Basic technique: K-means algorithm [slow but effective.]
- Example of application : cluster bloggers by 'social groups' (anarchists, ecologists, sports-fans, liberals-conservative, ...)

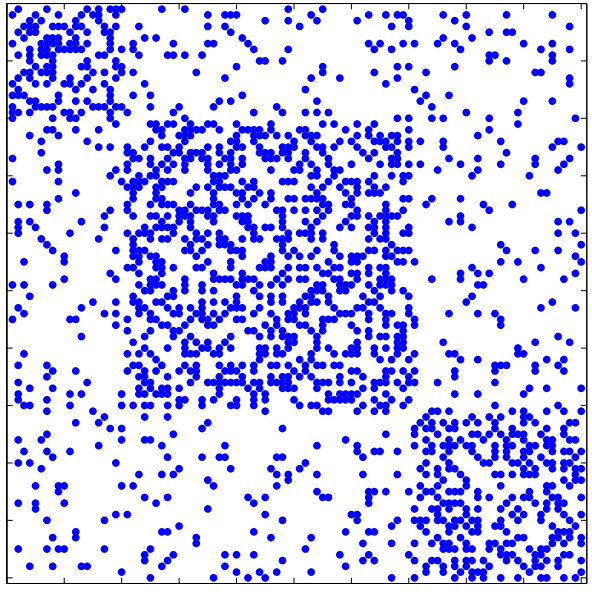
Sparse Matrices viewpoint

* Joint work with J. Chen

- Communities modeled by an 'affinity' graph [e.g., 'user A sends frequent e-mails to user B']
- Adjacency Graph represented by a sparse matrix



- Goal: find ordering so blocks are as dense as possible:



- Use 'blocking' techniques for sparse matrices

➤ Advantage of this viewpoint: need not know # of clusters

Example of application Data set from :

`http://www-personal.umich.edu/~mejn/netdata/`

➤ Network connecting bloggers of different political orientations [2004 US presidentual election]

➤ 'Communities': liberal vs. conservative

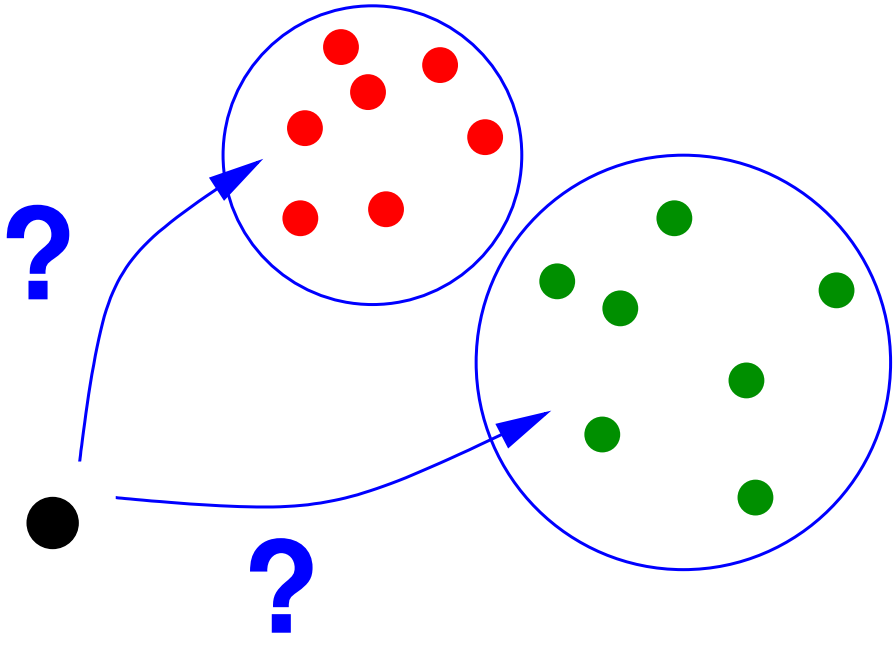
➤ Graph: $1,490$ vertices (blogs) : first 758: liberal, rest: conservative.

➤ Edge: $i \longrightarrow j$: a citation between blogs i and j

➤ Blocking algorithm (Density theshold=0.4): subgraphs [note: density = $|E|/|V|^2$.] ➤ Smaller subgraph: conservative blogs, larger one: liberals

Supervised learning: classification

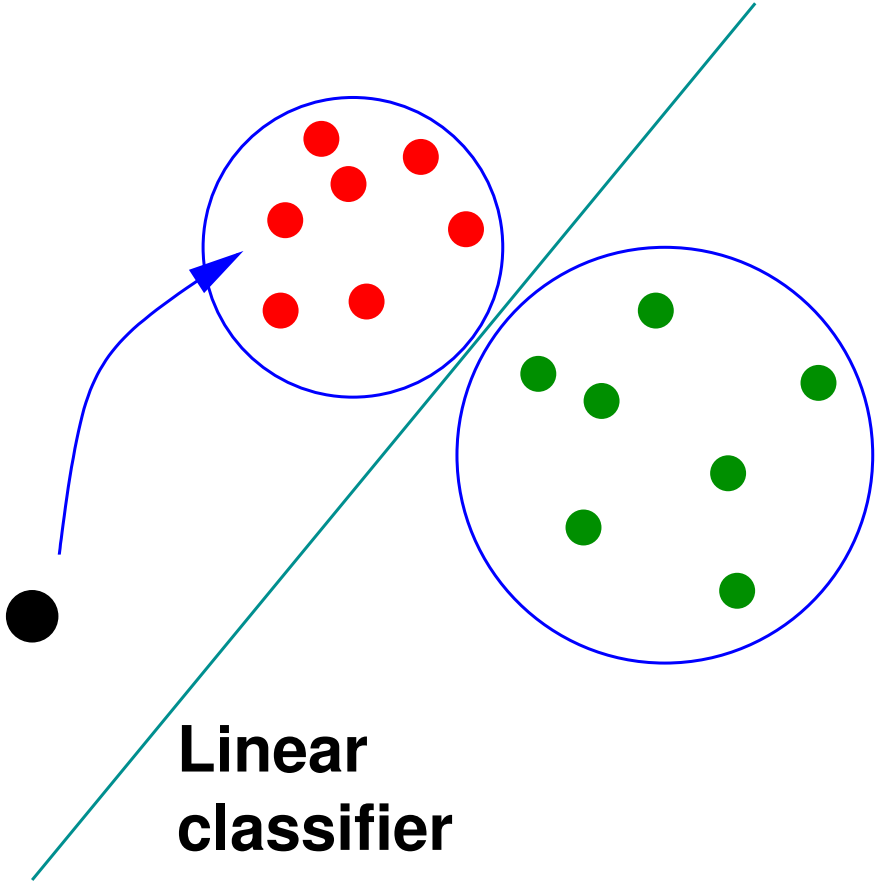
Problem: Given labels (say “A” and “B”) for each item of a given set, find a mechanism to classify an unlabelled item into either the “A” or the “B” class.



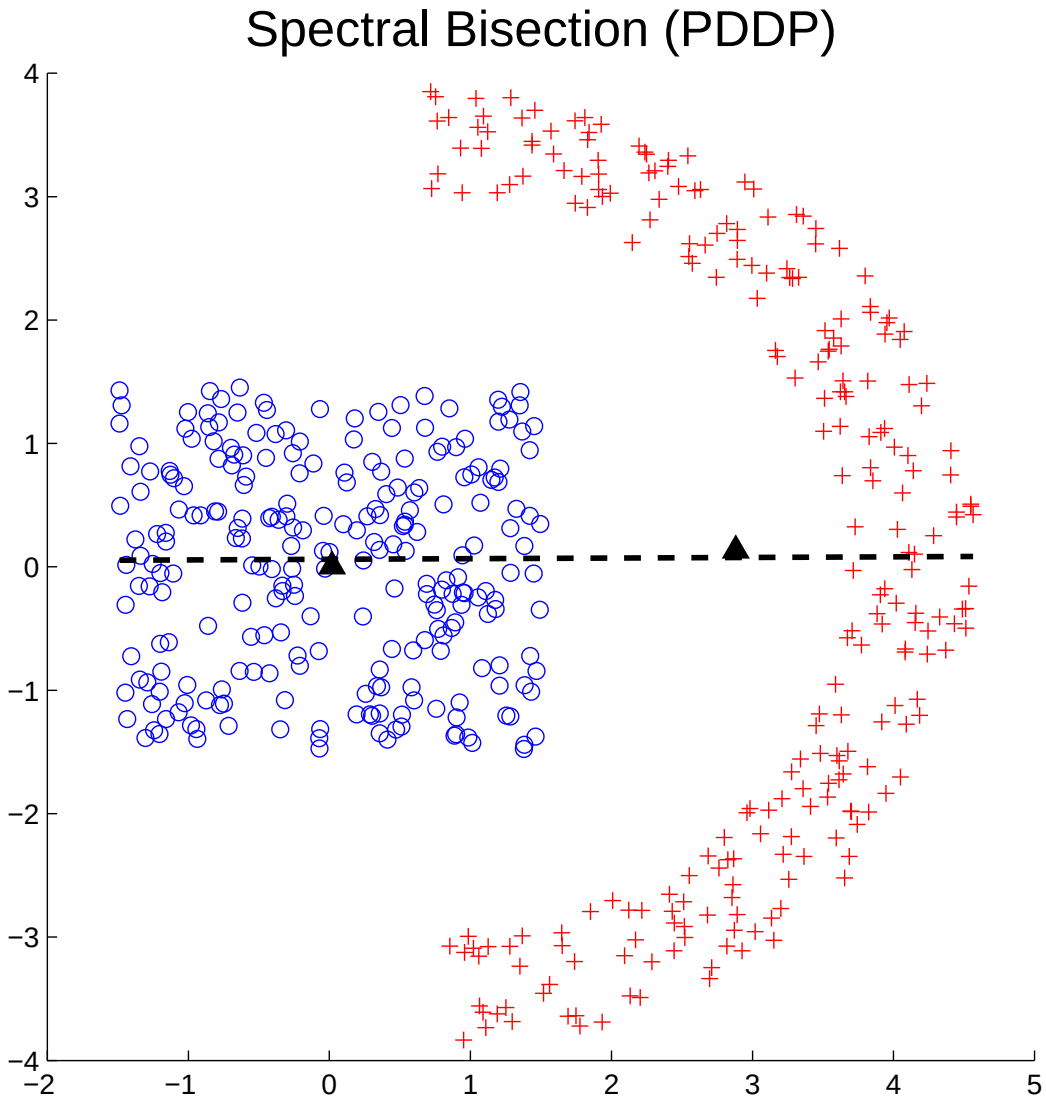
- Many applications.
- Example: distinguish SPAM and non-SPAM messages
- Can be extended to more than 2 classes.

Linear classifiers

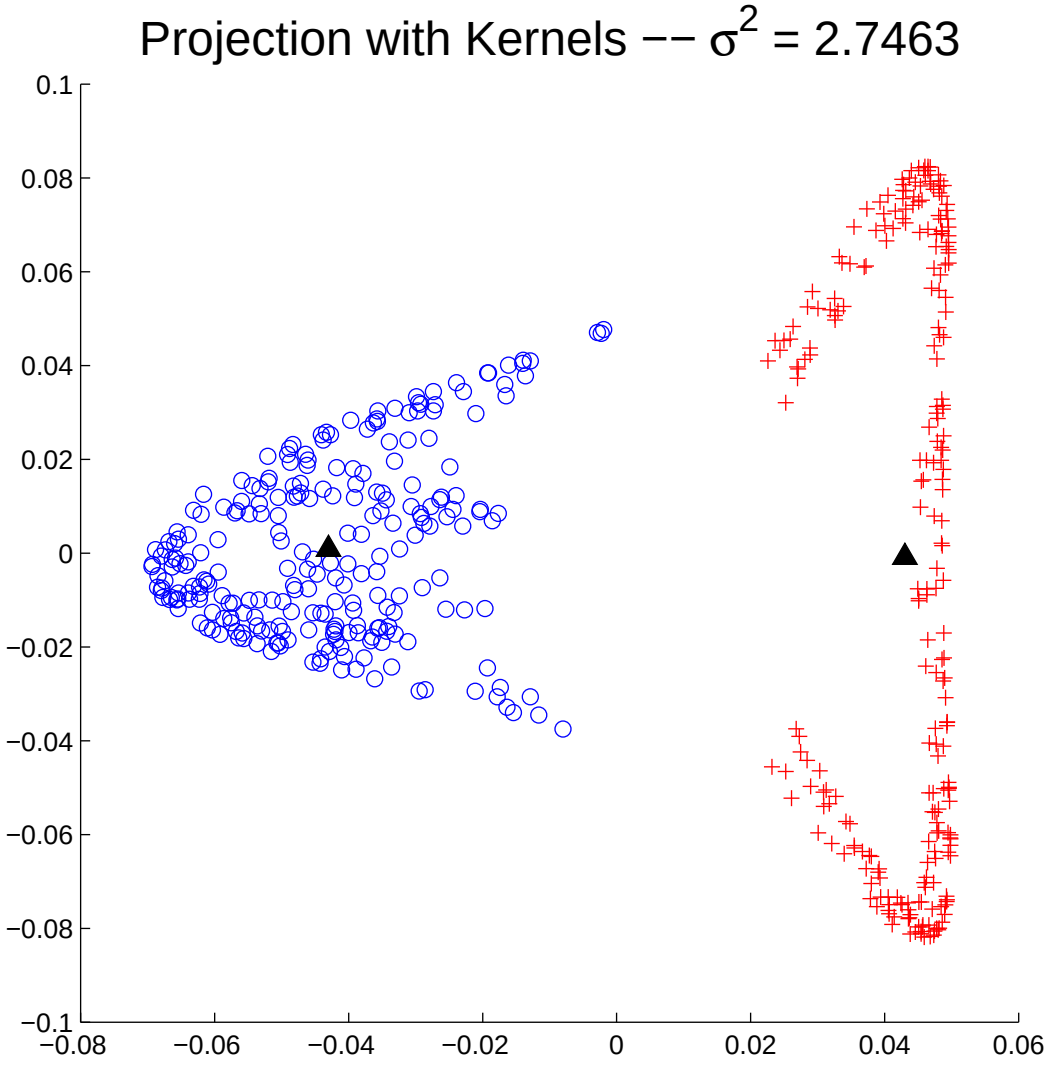
➤ Find a hyperplane which best separates the data in classes A and B.



A harder case:



- Use kernels to transform



Transformed data with a Gaussian Kernel

Fisher's Linear Discriminant Analysis (LDA)

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.

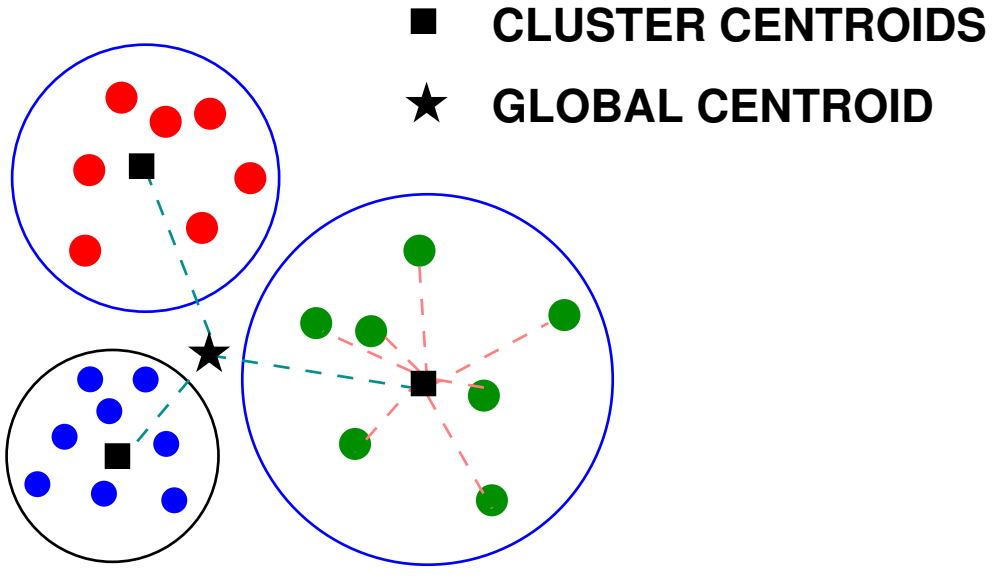
➤ Goal: to make "between scatter" measure large, while making "within scatter" small.

Idea: Project the data in low-dimensional space so as to maximize the ratio of the "between scatter" measure over "within scatter" measure of the classes.

Let μ = mean of X, and $\mu^{(k)}$ = mean of the k-th class (of size n_k). Define:

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

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



➤ Project set on a one-dimensional space spanned by a vector a.
Then:

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

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

➤ 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 – in d 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 $S_B u_i = \lambda_i S_W u_i$.

Trace ratio problem

* Joint work with Thanh Ngo and Mohamed Bellalij

➤ Main point: trace ratio can be maximized inexpensively

Let $$f(\rho) = \max_{U, U^T U = I} \textbf{Trace}[U^T (A - \rho B) U]$$

And let $U(\rho)$ the maximizer of above trace.

➤ Then, under some mild assumptions on A and B :

1) Optimal ρ for trace ratio is a zero of f
2) $f'(\rho) = -\text{Trace}[U(\rho)^T B U(\rho)]$
3) f is a decreasing function

... + Newton's method to find zero amounts to a fixed point iteration:

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

- Idea: Compute $U(\rho)$ by an inexpensive Lanczos procedure
- Note: Standard problem - [not generalized] $\longrightarrow$ cheap..
- 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*

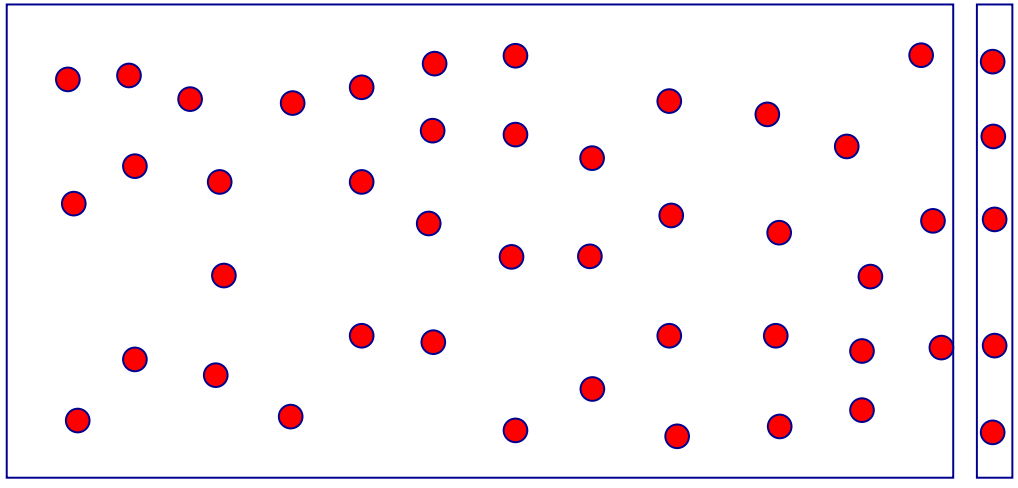
1. *Choose an initial vector* v_1 *of norm unity.*
   Set $\beta_1 \equiv 0, v_0 \equiv 0$
2. *For* $j = 1, 2, \ldots, m$ *Do:*
3. $\alpha_j := (w_j, v_j)$
4. $\boxed{w_j := Av_j - \alpha_j v_j - \beta_j v_{j-1}}$
5. $\beta_{j+1} := \|w_j\|_2$. *If* $\beta_{j+1} = 0$ *then Stop*
6. $v_{j+1} := w_j /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

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

INFORMATION RETRIEVAL

Information Retrieval: Vector Space Model

➤ Given: a collection of documents (columns of a matrix A) and a query vector q.



➤ Collection represented by an $m \times n$ term by document matrix with $\boxed{a_{ij} = L_{ij} G_i N_j}$

➤ Queries ('pseudo-documents') q are represented similarly to a column

Vector Space Model - continued

- Problem: find a column of A that best matches q
- Similarity metric: angle between the column and q - Use cosines:

$$\frac{|c^T q|}{\|c\|_2 \|q\|_2}$$

- To rank all documents we need to compute

$$s = A^T q$$

- s = similarity vector.
- Literal matching – not very effective.

Common approach: Use the SVD

- Need to extract intrinsic information – or underlying "semantic" information –
- LSI: replace A by a low rank approximation [from SVD]

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

- U_k : term space, V_k: document space.
- New similarity vector: $s_k = A_k^T q = V_k \Sigma_k U_k^T$
- Main issues: 1) computational cost 2) Updates

Use of polynomial filters

* Joint work with E. Kokiopoulou

The idea: Replace A_k by $A\phi(A^TA)$, where ϕ is a certain filter function

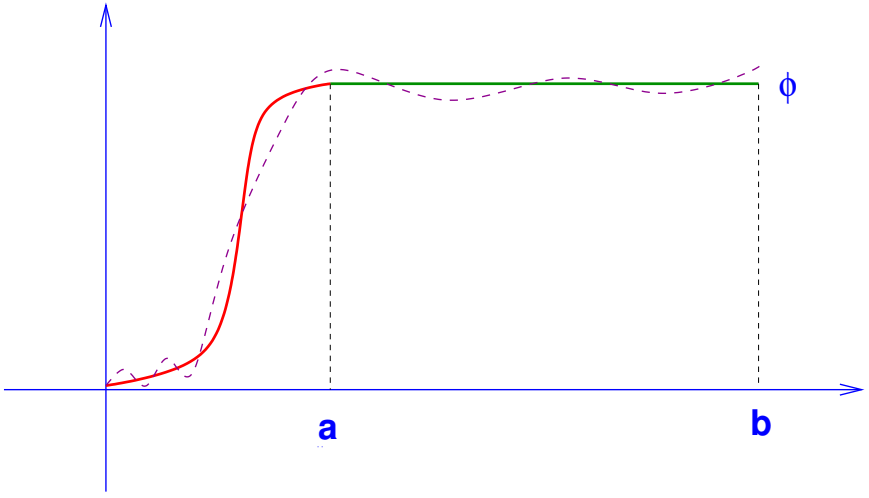
Consider the step-function (Heaviside function:)

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

- This would yield the same result as with TSVD but...
- ... Not easy to use this function directly
- Solution : use a polynomial approximation to ϕ
- Note: $\boxed{s^T = q^T A\phi(A^TA)}$, requires only Mat-Vec's

Polynomial filters - examples

Approach: Approximate a piecewise polynom. by a polynomial in least-squares sense



➤ 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

* Joint work with Jie Chen

➤ 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..

➤ Related idea: K. Blom and A. Ruhe [SIMAX, vol. 26, 2005]. Use Lanczos bidiagonalization.

➤ Use a similar approach – But directly with AA^T or A^TA.

IR: Use of the Lanczos algorithm (1)

- Let $A \in \mathbb{R}^{m \times n}$. Apply the Lanczos procedure to $M = AA^T$. Result:

$$Q_k^T A A^T Q_k = T_k$$

with Q_k orthogonal, T_k tridiagonal.

- Define $s_i \equiv$ orth. projection of Ab on subspace span$\{Q_i\}$

$$s_i := Q_i Q_i^T A b.$$

- s_i can be easily updated from s_{i-1}:

$$s_i = s_{i-1} + q_i q_i^T A b.$$

IR: Use of the Lanczos algorithm (2)

- If $n < m$ it may be more economial to apply Lanczos to $M = A^T A$ which is $n \times n$. Result:

$$\bar{Q}_k^T A^T A \bar{Q}_k = \bar{T}_k$$

- Define:

$$t_i := A \bar{Q}_i \bar{Q}_i^T b,$$

- Project b first before applying A to result.

➤ Theory well understood: works well because Lanczos produces good approximations to dominant sing. vectors

Advantages of Lanczos over polynomial filters:

(1) No need for eigenvalue estimates

(2) Mat-vecs performed only in preprocessing

Disadvantages:

(1) Need to store Lanczos vectors;

(2) Preprocessing must be redone when A changes.

(3) Need for reorthogonalization – expensive for large k.

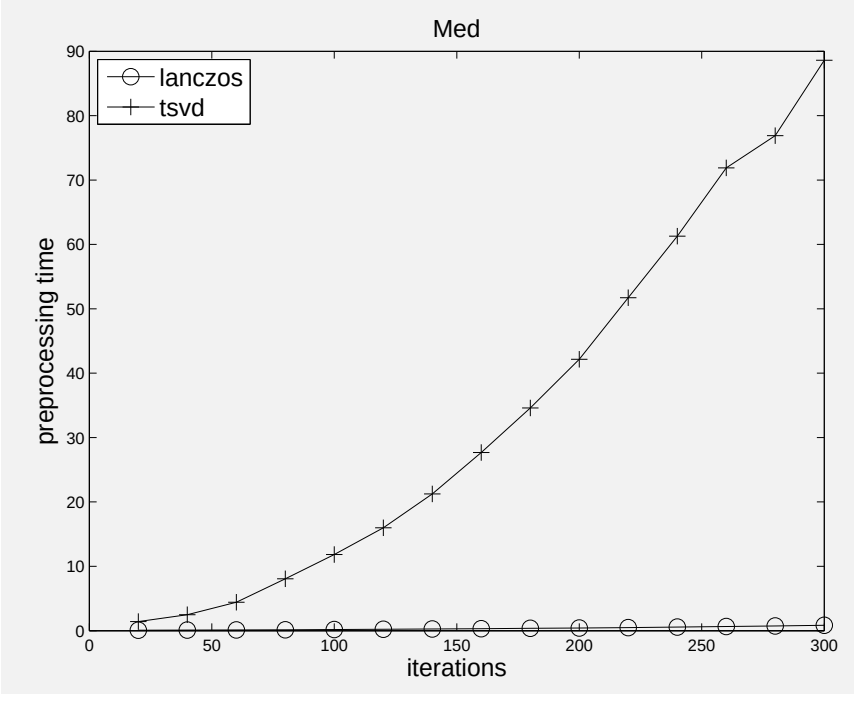
Tests: IR

Information retrieval datasets

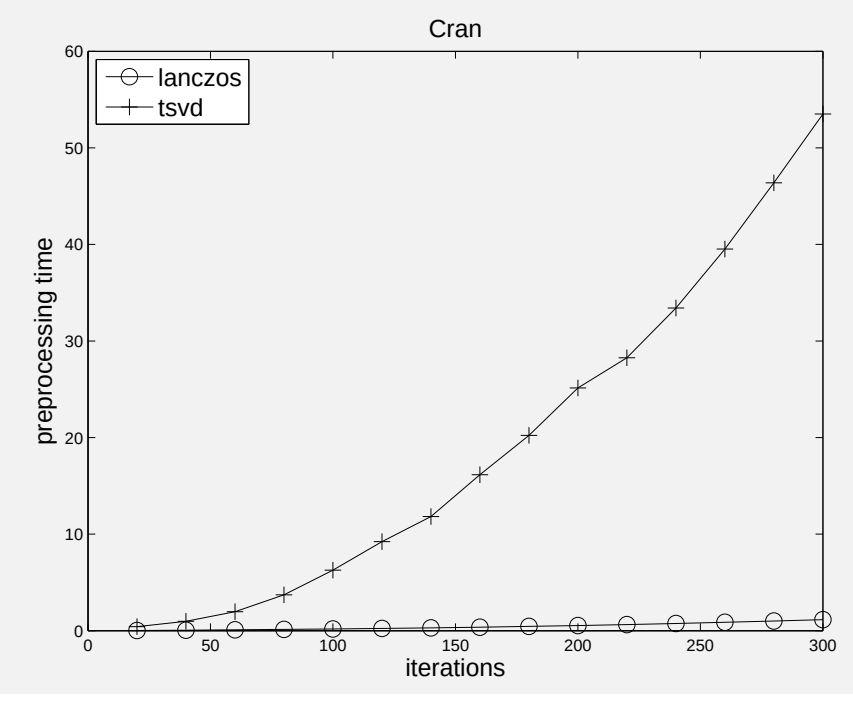
	# Terms	# Docs	# queries	sparsity
MED	7,014	1,033	30	0.735
CRAN	3,763	1,398	225	1.412

Preprocessing times

Med dataset.

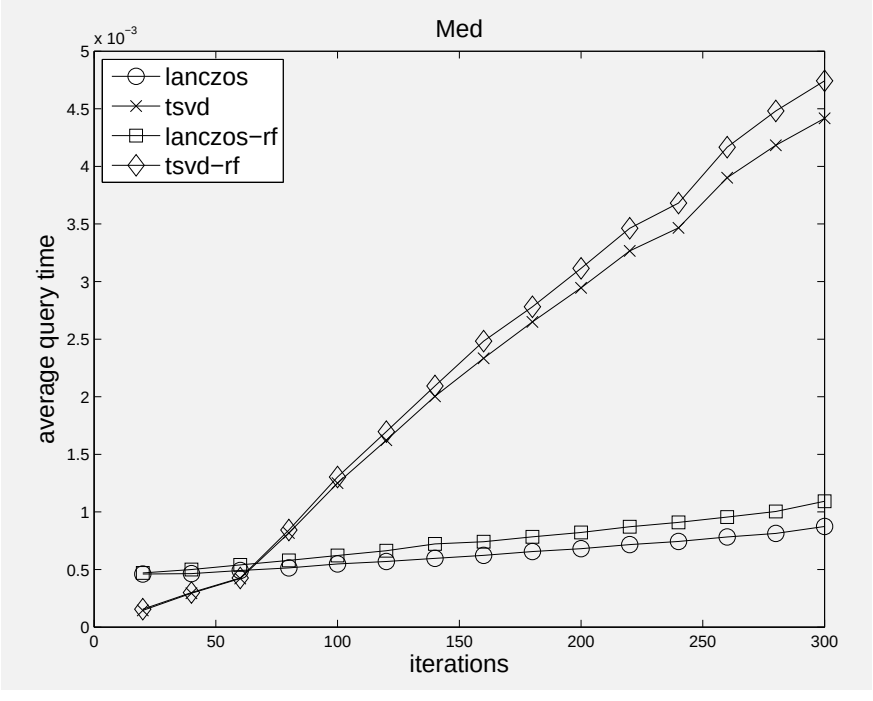


Cran dataset.

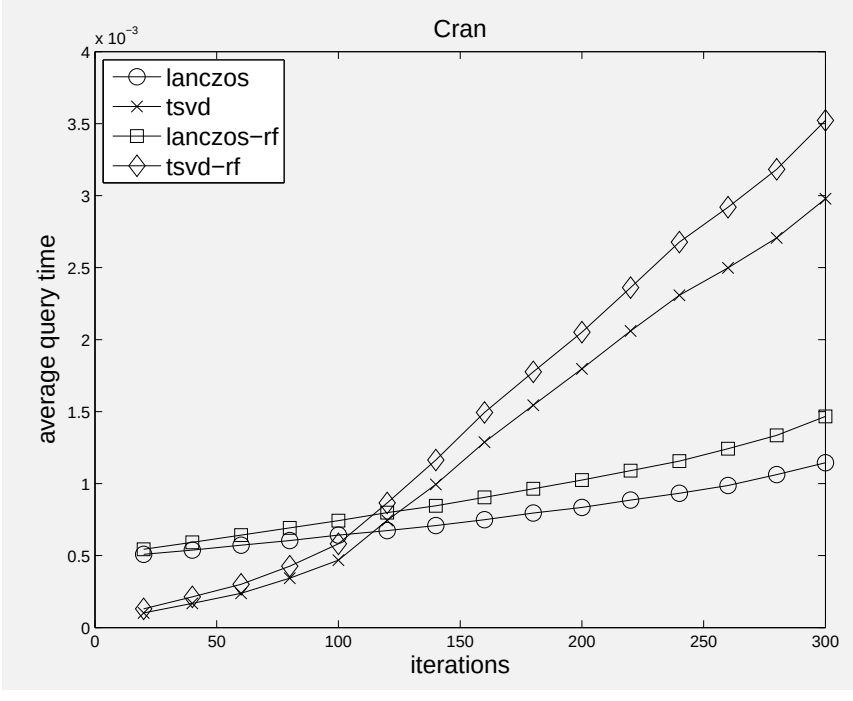


Average query times

Med dataset

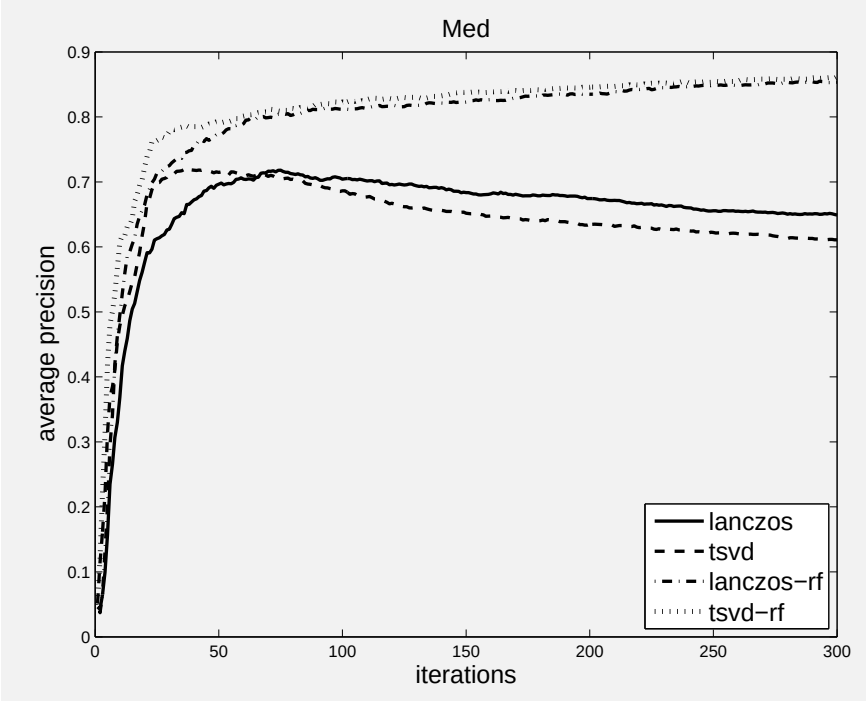


Cran dataset.

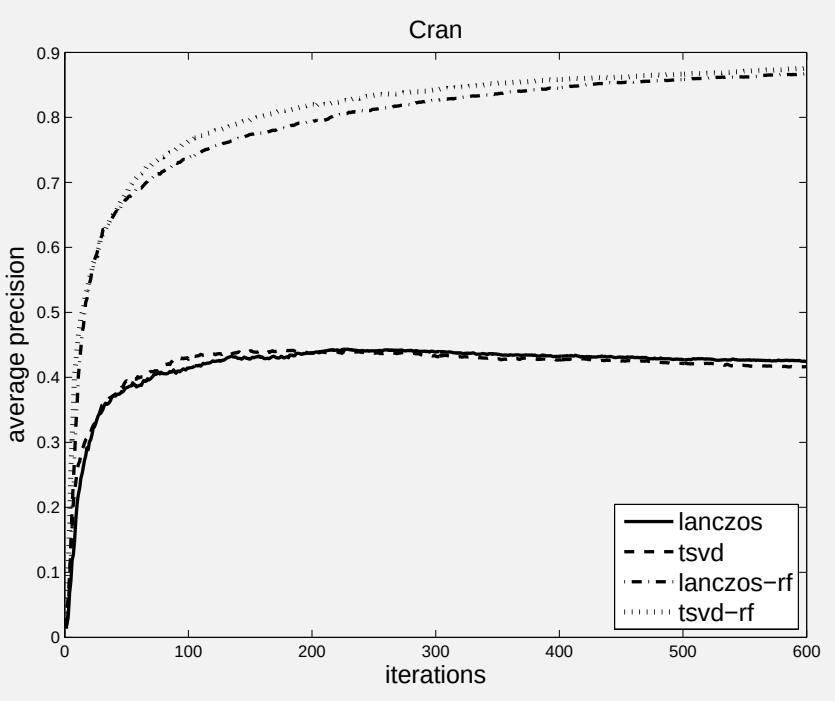


Average retrieval precision

Med dataset



Cran dataset



Retrieval precision comparisons

In summary:

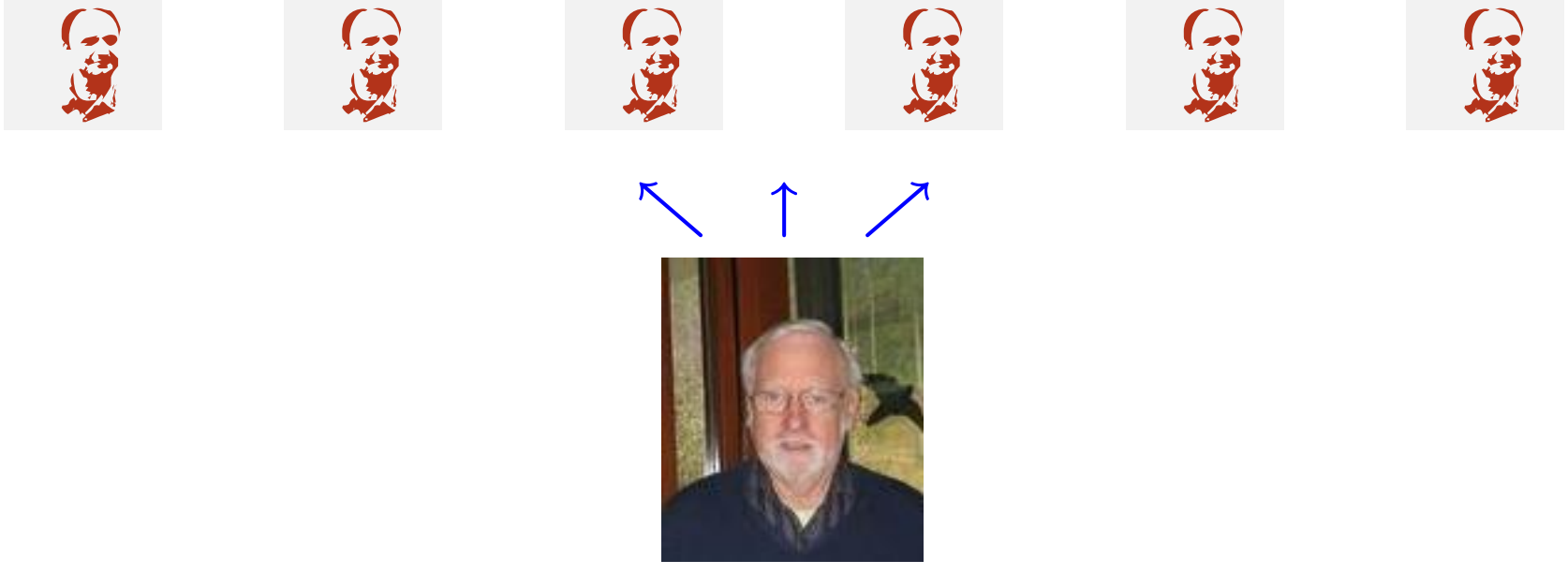
- Results comparable to those of SVD ...
- .. at a much lower cost.

Thanks:

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

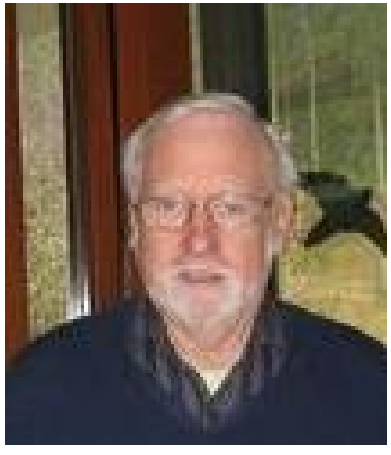
Face Recognition – background

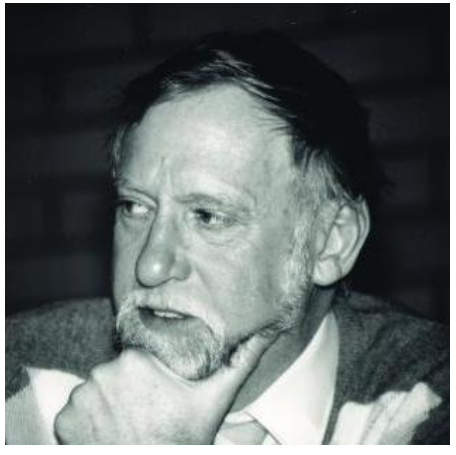
Problem: We are given a database of images: [arrays of pixel values]. And a test (new) image.



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

Difficulty Positions, Expressions, Lighting, ...,



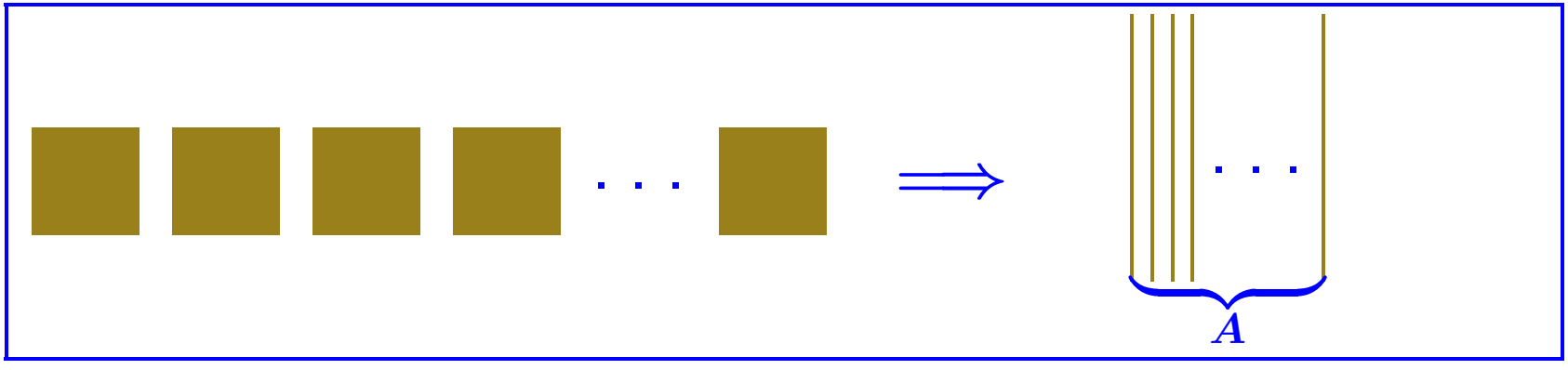
Eigenfaces: Principal Component Analysis technique

- Specific situation: Poor images or deliberately altered images ['occlusion']
- See real-life examples – [international man-hunt]

Eigenfaces

– 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.

Idea: replace SVD by Lanczos vectors (same as for IR)

Tests: Face Recognition

Tests with 2 well-known data sets:

ORL 40 subjects, 10 sample images each – example:

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

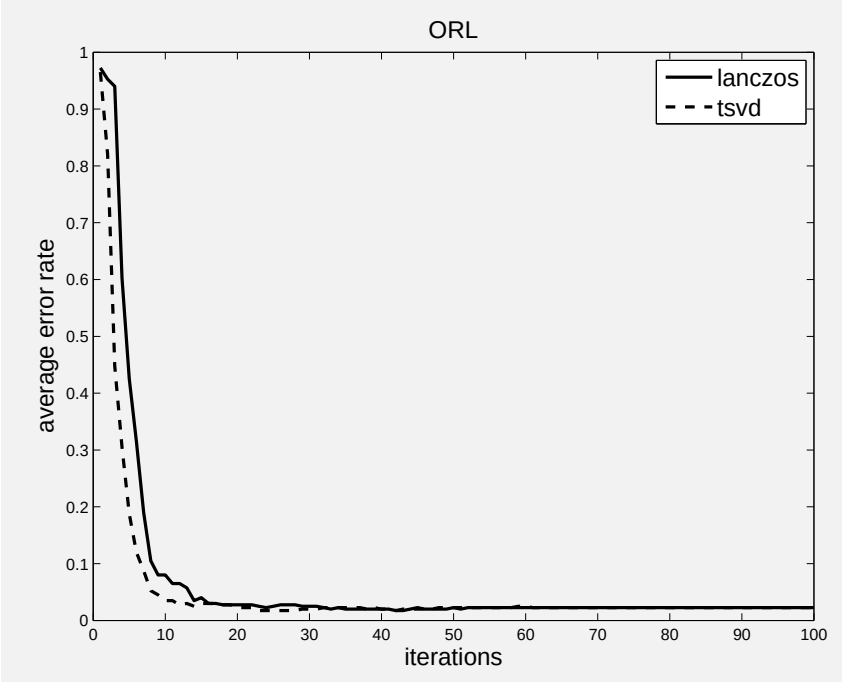
AR set 126 subjects – 4 facial expressions selected for each [natural, smiling, angry, screaming] – example:

# of pixels : 112×92 # TOT. # images : 504

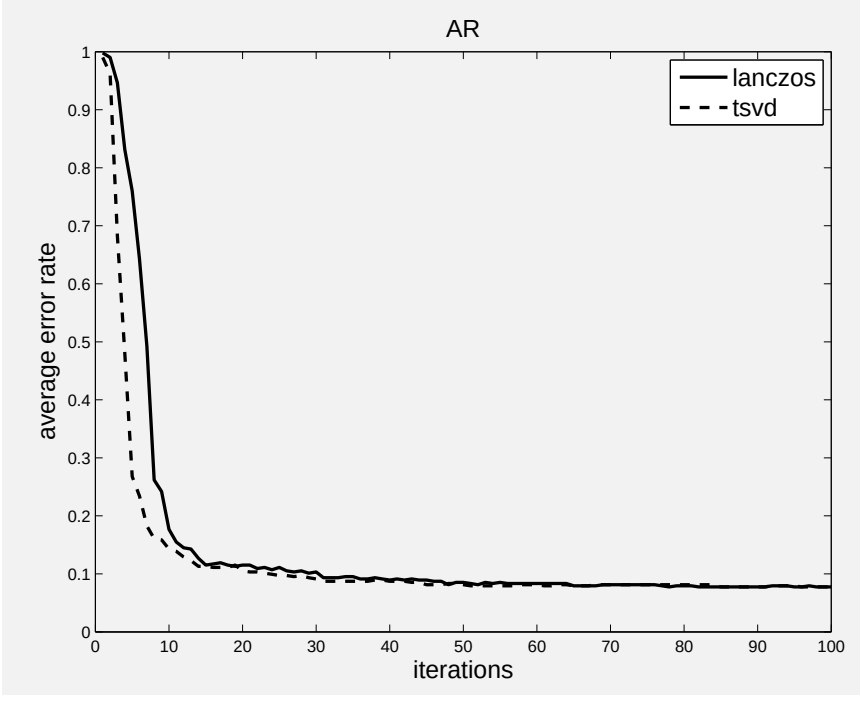
Tests: Face Recognition

Recognition accuracy of Lanczos approximation vs SVD

ORL dataset



AR dataset



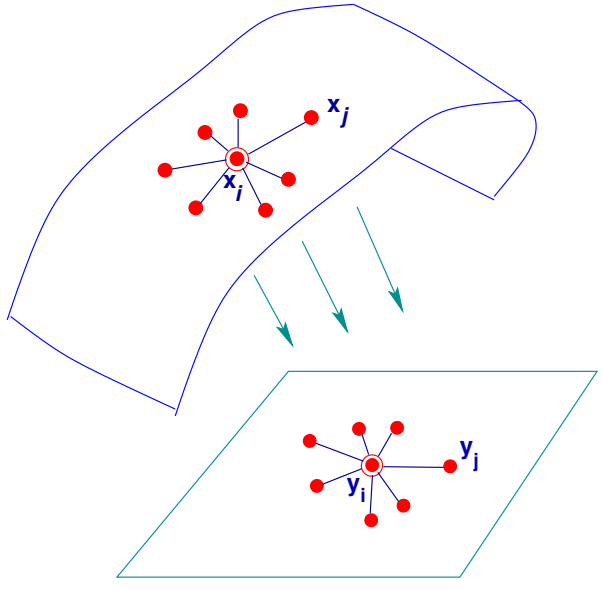
Vertical axis shows average error rate. Horizontal = Subspace dimension

GRAPH-BASED METHODS

Graph-based methods

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

Want: Do a projection so as to preserve the graph in some sense



➤ Define a *graph Laplacean:*

$$L = D - W$$

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

with N_i = neighborhood of i (excl. i)

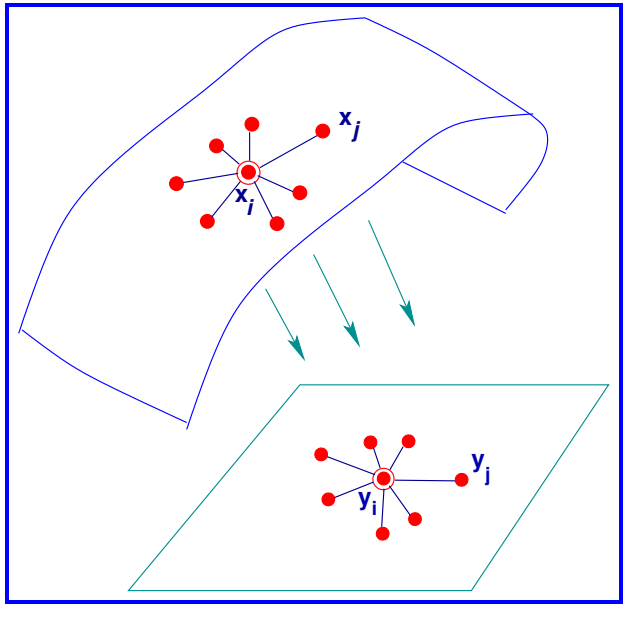
Example: The Laplacean eigenmaps

Laplacean Eigenmaps [Belkin & Niyogi'02] *minimizes*

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

Notes:

1. Motivation: if $\|x_i - x_j\|$ is small (orig. data), we want $\|y_i - y_j\|$ to be also small (low-dim. data)
2. Note: Min instead of Max as in PCA [counter-intuitive]
3. Above problem uses original data indirectly through its graph



- Problem translates to:

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

- Solution (sort eigenvalues increasingly):

$$(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{u}_i = \boldsymbol{\lambda}_i \boldsymbol{D}\boldsymbol{u}_i \; ; \quad \boldsymbol{y}_i = \boldsymbol{u}_i^\top ; \quad \boldsymbol{i} = 1, \cdots, \boldsymbol{d}$$

- An $n \times n$ sparse eigenvalue problem [In 'sample' space]

- Note: can assume $\boldsymbol{D} = \boldsymbol{I}$. Amounts to rescaling data.

Problem becomes

$$(\boldsymbol{I}-\boldsymbol{W})\boldsymbol{u}_i = \boldsymbol{\lambda}_i \boldsymbol{u}_i \; ; \quad \boldsymbol{y}_i = \boldsymbol{u}_i^\top ; \quad \boldsymbol{i} = 1, \cdots, \boldsymbol{d}$$

A unified view

* Joint work with Efi Kokiopoulou and J. Chen

➤ Most techniques lead to one of two types of problems

First :

➤ $\boldsymbol{Y}$ results directly from computing eigen-vectors

➤ LLE, Eigenmaps, ...

$$\min \quad \textbf{Tr}\left[\boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^\top\right] \quad \begin{cases} \boldsymbol{Y} \in \mathbb{R}^{d\times n} \\ \boldsymbol{Y}\boldsymbol{Y}^\top = \boldsymbol{I} \end{cases}$$

Second:

➤ Low-Dimens. data: $\boldsymbol{Y} = \boldsymbol{V}^\top \boldsymbol{X}$

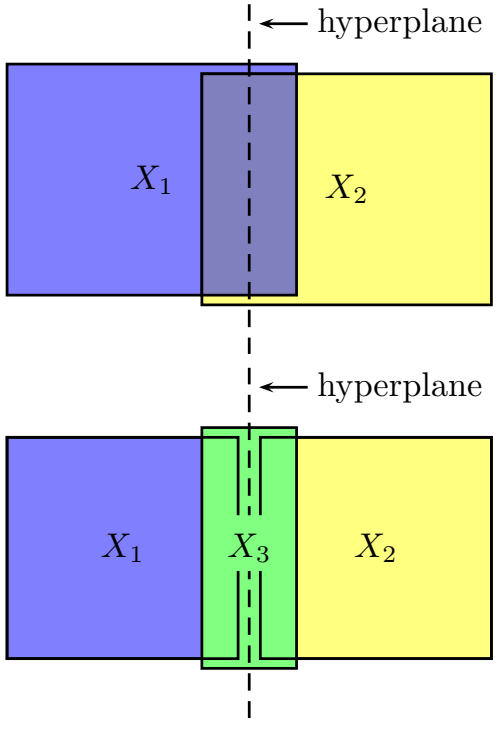
➤ $\boldsymbol{G}$ == identity, or $\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top$, or $\boldsymbol{X}\boldsymbol{X}^\top$

$$\min \quad \textbf{Tr}\left[\boldsymbol{V}^\top\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{V}\right] \quad \begin{cases} \boldsymbol{V} \in \mathbb{R}^{m\times d} \\ \boldsymbol{V}^\top\boldsymbol{G}\,\boldsymbol{V} = \boldsymbol{I} \end{cases}$$

Observation: 2nd is just a projected version of the 1st.

Computing k-nn graphs

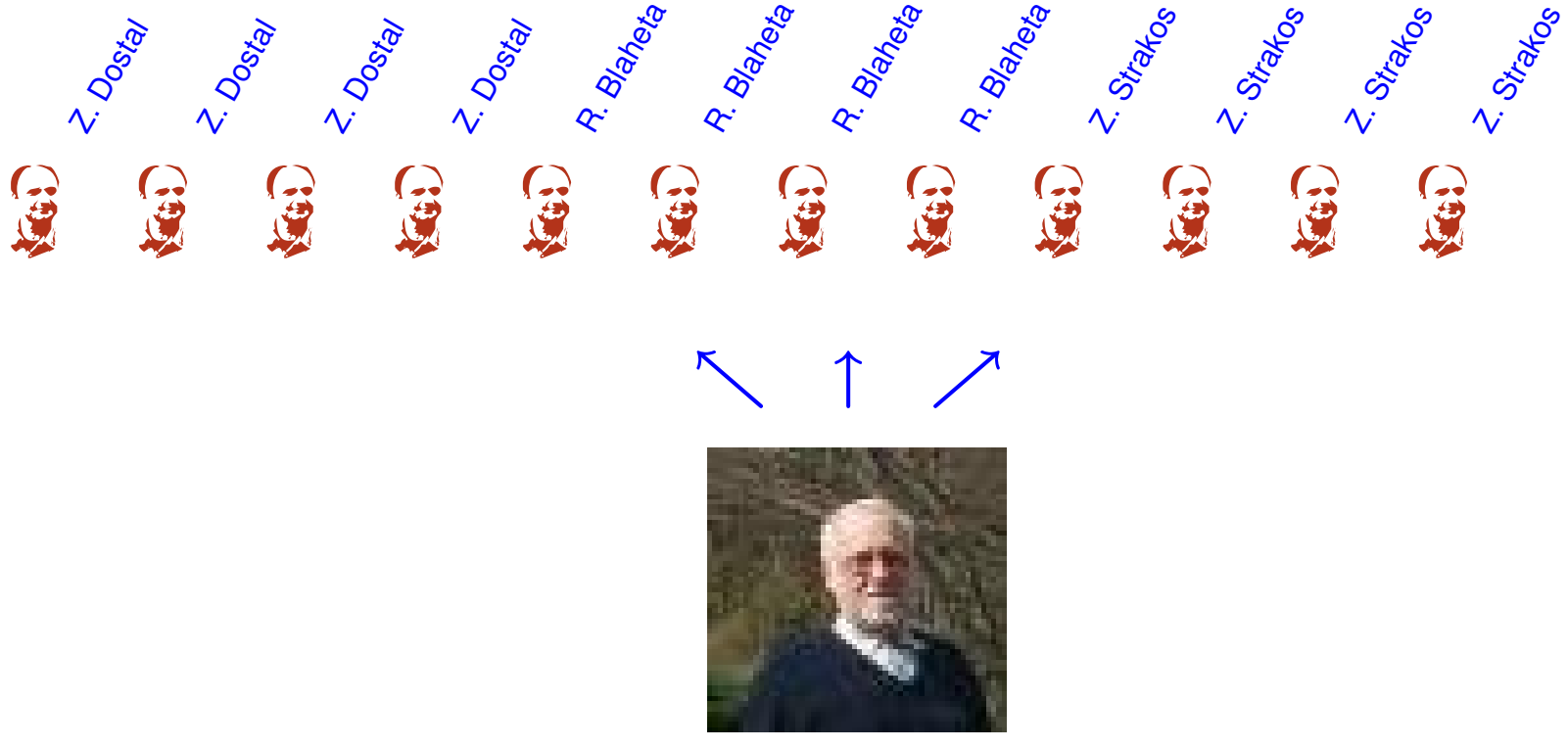
* Joint work with J. Chen and H. R. Fang



- Main idea: Divide and Conquer
- 2-nd smallest eigenvector roughly separates set in two.
- Recursively divide + handle interface with care
- Again exploit Lanczos!
- Cost: $O(n^{1+\delta})$, δ depends on parameters. Less than $O(n^2)$ but higher than $O(n \log n)$
- C-implementation [J. Chen] publically available [GPL]

Graph-based methods in a supervised setting

➤ Subjects of training set are known (labeled). Q: given a test image (say) find its label.



Question: Find label (best match) for test image.

Methods can be adapted to supervised mode by building the graph to use class labels. Idea: Build G so that nodes in the same class are neighbors. If c = # classes, G consists of c cliques.

- Matrix W is block-diagonal

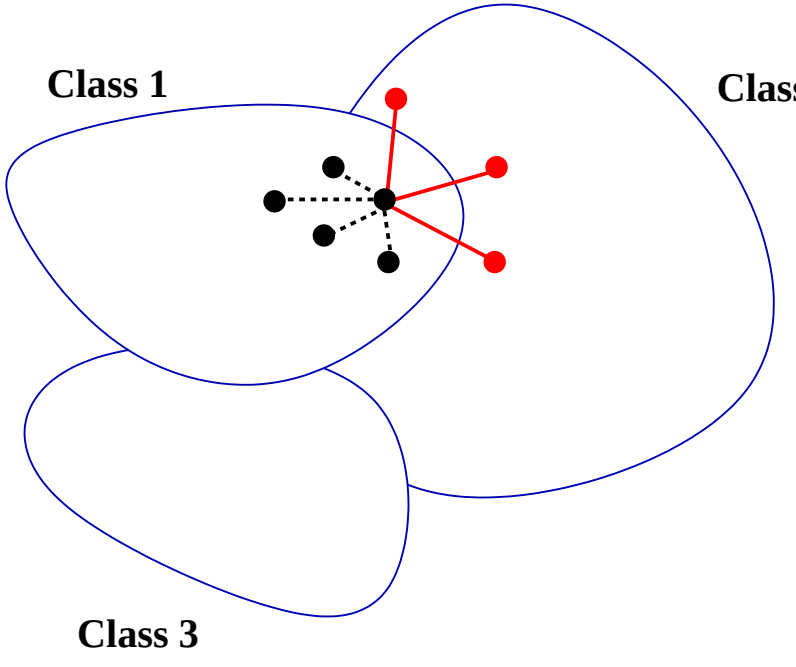
$$W = \begin{pmatrix} W_1 & & & & \\ & W_2 & & & \\ & & W_3 & & \\ & & & W_4 & \\ & & & & W_5 \end{pmatrix}$$

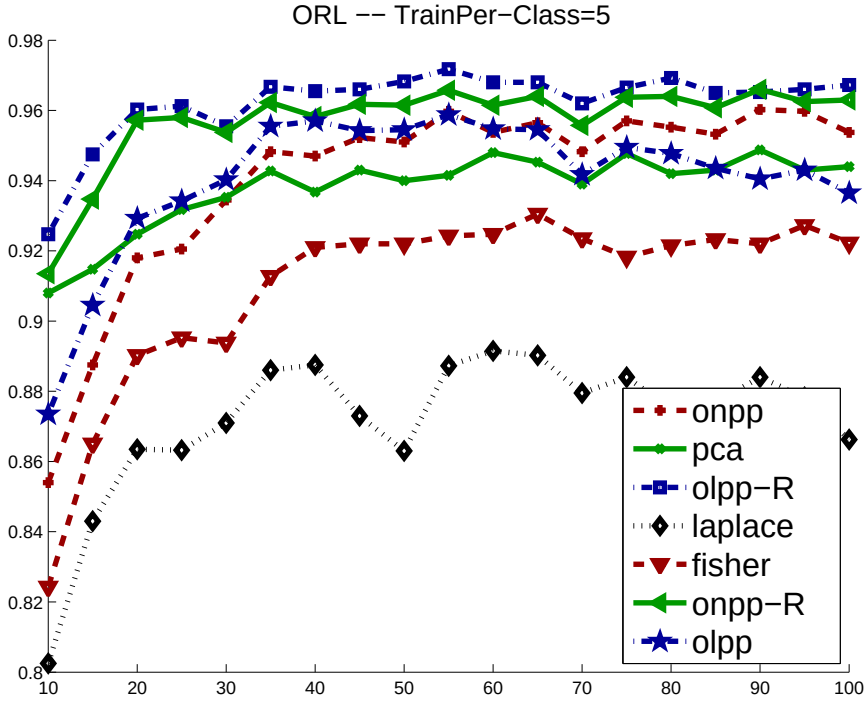
- Note: $\mathbf{rank}(W) = n - c$.
- Can be used for LPP, ONPP, etc..

Repulsion Laplaceans

* Joint work with E. Kokiopoulou ('09)

➤ Idea: Create repulsion forces (energies) between items that are close but *not* in the same class.





Data mining for materials: Materials Informatics

- Huge potential in exploiting two trends:

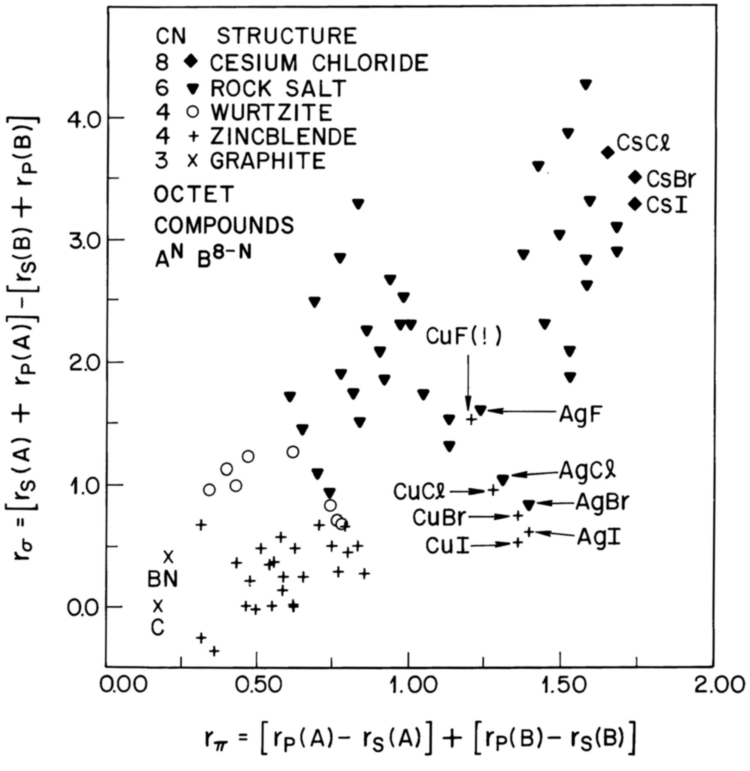
1 Enormous improvements in efficiency and capabilities in computational methods for materials

2 Recent progress in data mining techniques

- For example, cluster materials into classes according to properties, types of atomic structures ('point groups') ...
- Current practice: "One student, one alloy, one PhD" $\longrightarrow$ Slow pace of discovery
- Data Mining: help speed-up process - look at more promising alloys

Materials informatics at work. Illustrations

- 1970s: Data Mining "by hand": Find coordinates to cluster materials according to structure
- 2-D projection from physical knowledge



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

- 'Anomaly Detection': helped find that compound Cu F does not exist

Example 1: [Norskov et al., '03, ...]

- Use of genetic algorithms to 'search' through database of binary materials. Lead to discovery of a promising catalytic material with low cost.

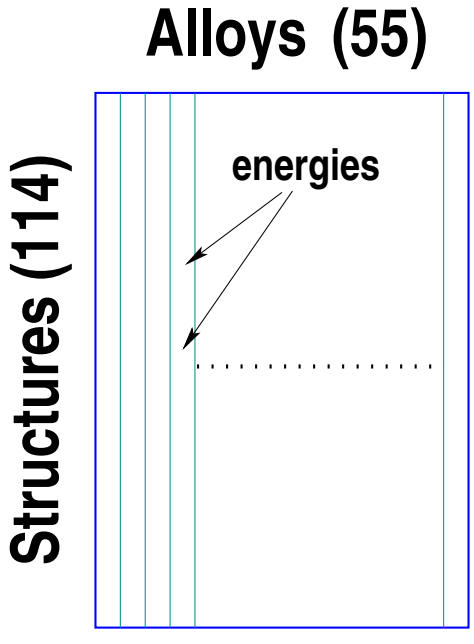
Example 2 : [Curtalano et al. PRL vol 91, 1003]

➤ Goal: narrow search to do fewer electronic structures calculations

➤ 55 binary metallic alloys considered in 114 crystal structures

➤ Observation: Energies of different crystal structures are correlated

➤ Use PCA: 9 dimensions good enough to yield OK accuracy –



Conclusion

➤ Many, many, interesting New matrix problems related to the new economy and new emerging scientific fields:

1. Information technologies [learning, data-mining, ...]
2. Computational Chemistry / materials science
3. Bio-informatics: computational biology, genomics, ..

➤ *Important:* Many resources for data-mining available on-line: repositories, tutorials, Very easy to get started

➤ Materials informatics very likely to become a major force

➤ For a few recent papers and pointers visit my web-site at

`www.cs.umn.edu/~saad`

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)

Thank you !