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

Modelling 2009 – Roznov pod Radhostem

June 22nd – 26th, 2009

To Owe Axelsson:

Happy birthday!

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 marked tanked .. and the interest disappeared .

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

Observations:

➤ New forces are starting to reshape numerical linear algebra

➤ Numerical analysts are often becoming “data analysts”, or “bio-informaticians...”

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

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

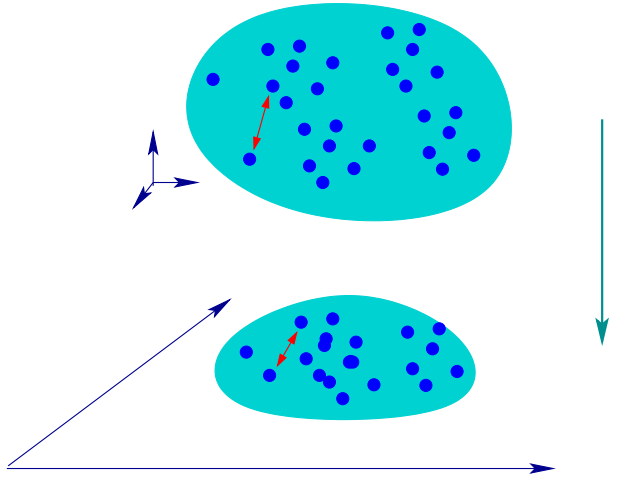
- Mapping may be explicit [typically linear], e.g.:

$$y = V^T x$$

- Or implicit (nonlinear)

Practically:

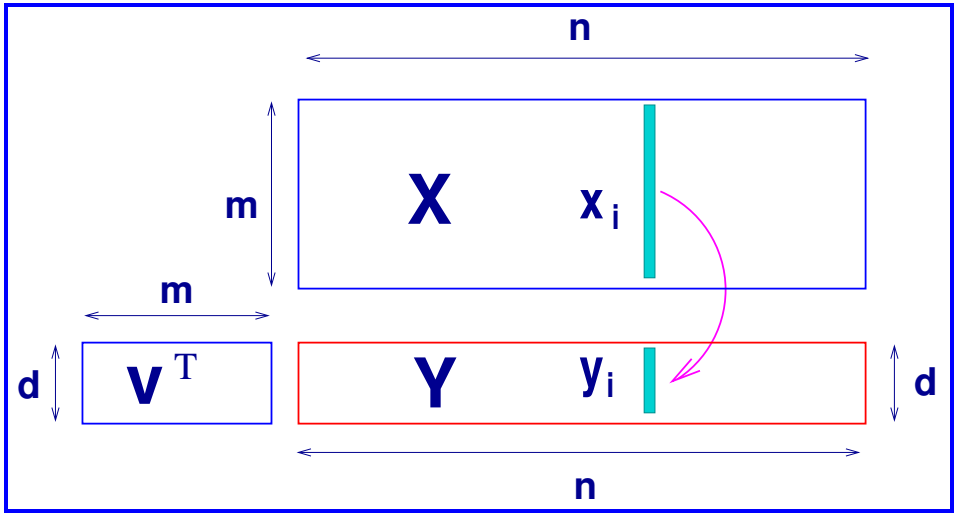
Given: $X \in \mathbb{R}^{m \times n}$.
Want: a low-dimensional representation $Y \in \mathbb{R}^{d \times n}$ of X



Linear Dimensionality Reduction

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

Want: A linear transformation from $\boldsymbol{X}$ to $\boldsymbol{Y}$



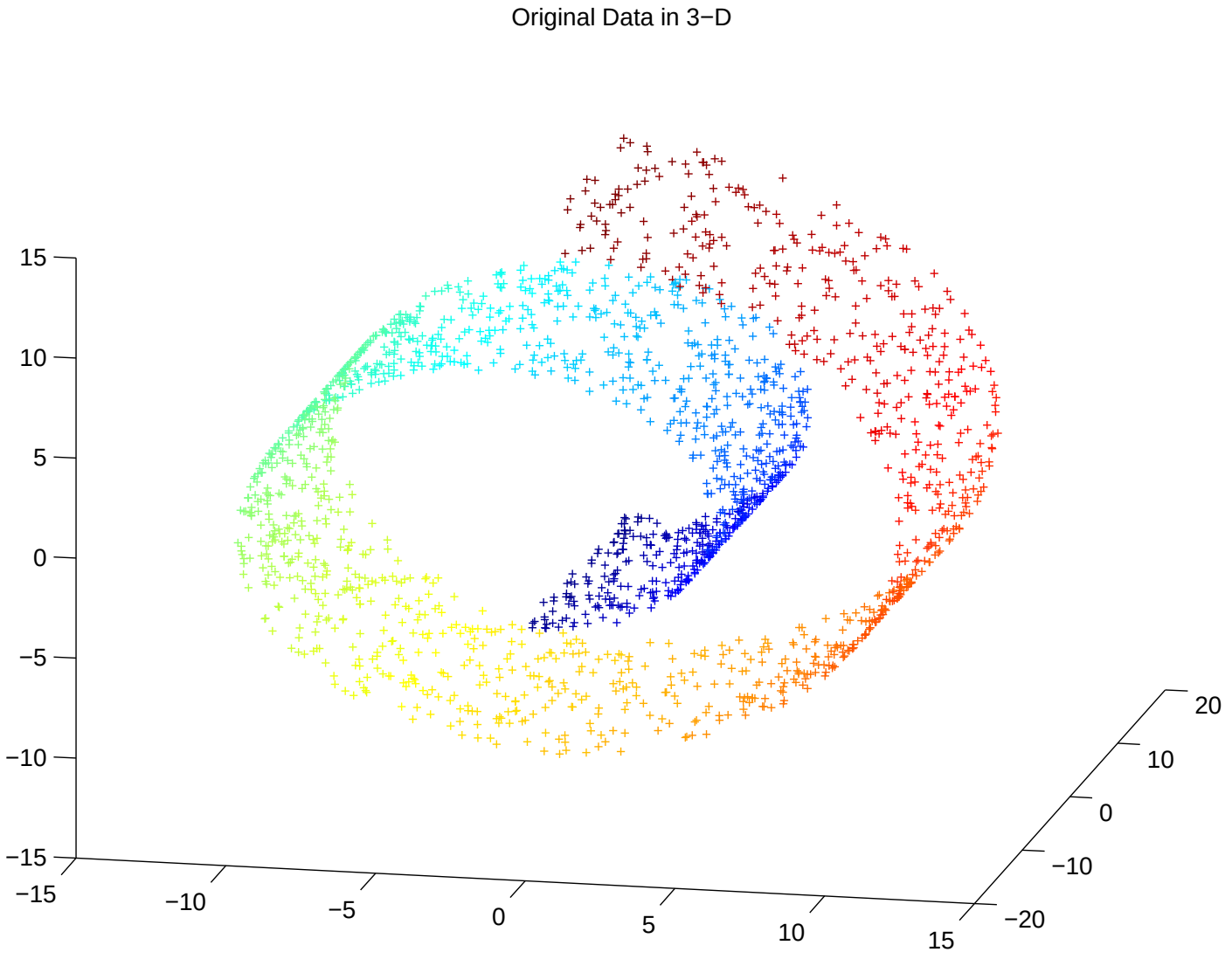
$$\boldsymbol{X} \in \mathbb{R}^{m \times n}$$
$$\boldsymbol{V} \in \mathbb{R}^{m \times d}$$
$$\boxed{\boldsymbol{Y} = \boldsymbol{V}^\top \boldsymbol{X}}$$
$$\longrightarrow \quad \boldsymbol{Y} \quad \in \mathbb{R}^{d \times n}$$

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

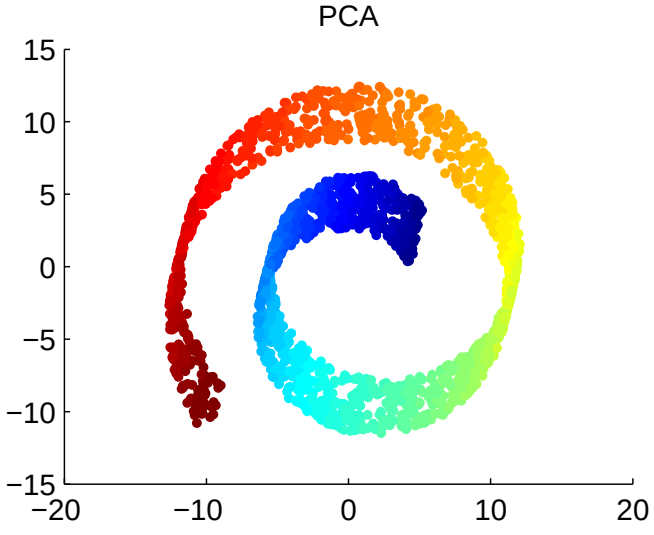
Constraint: The y_i's must satisfy certain properties

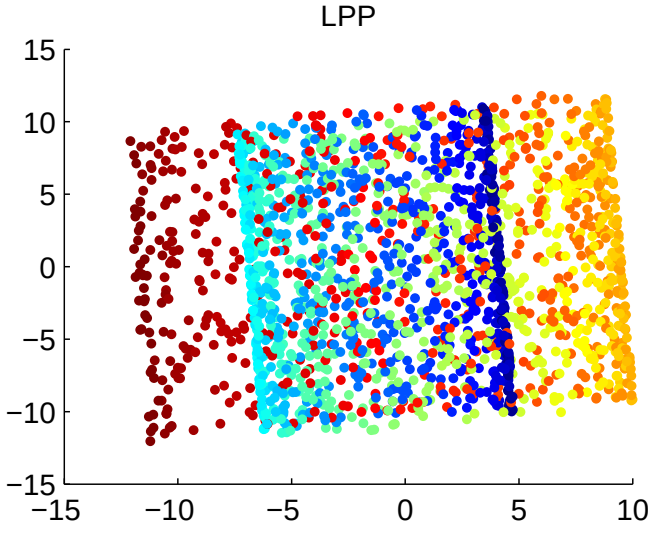
- Optimization problem

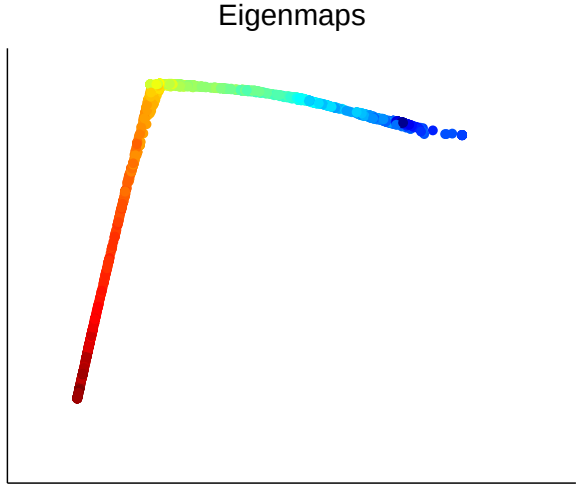
Example 1: The 'Swill-Roll' (2000 points in 3-D)

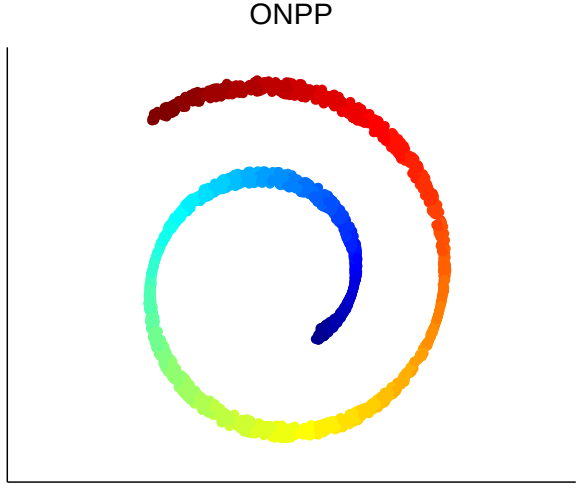


2-D ‘reductions’:

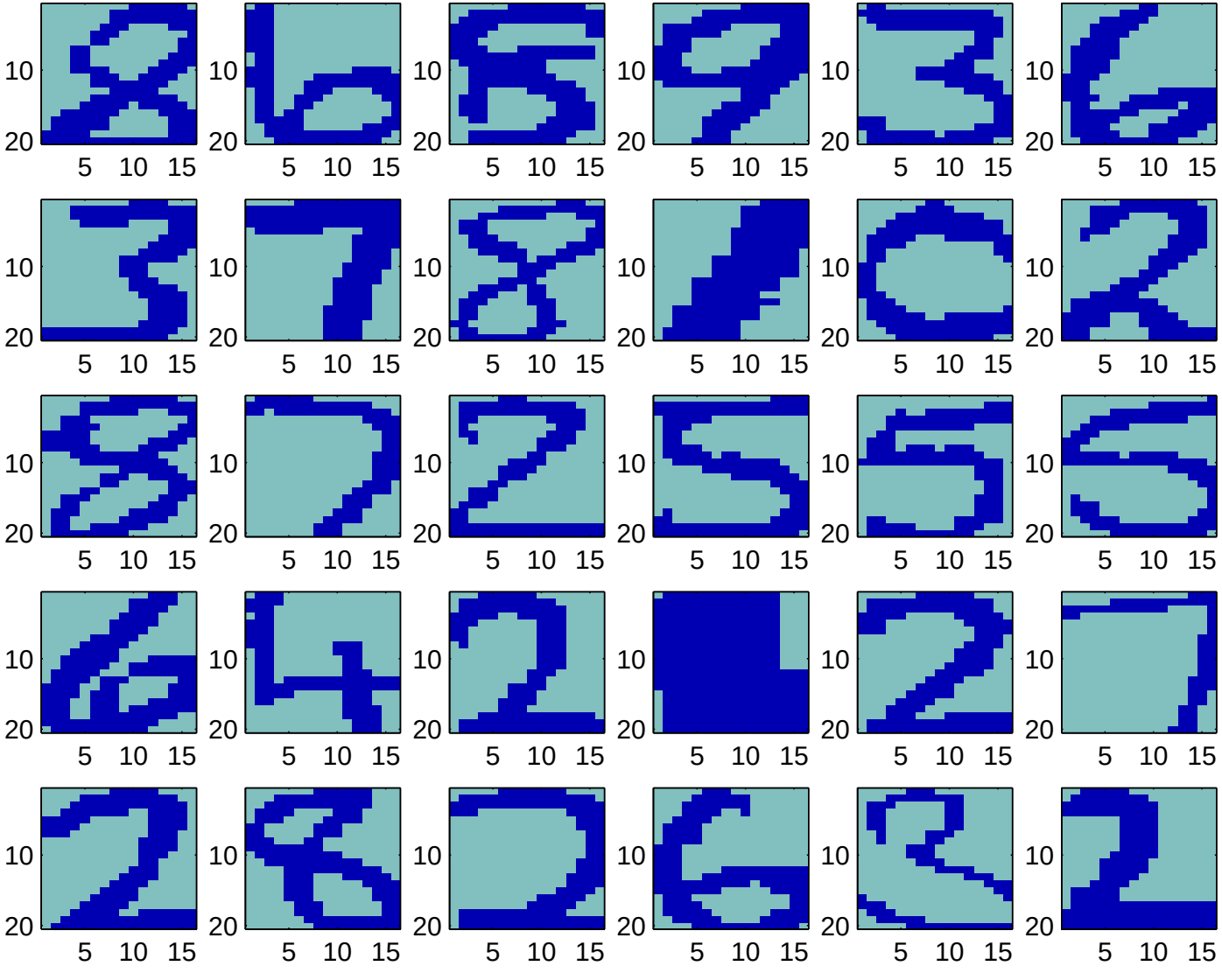




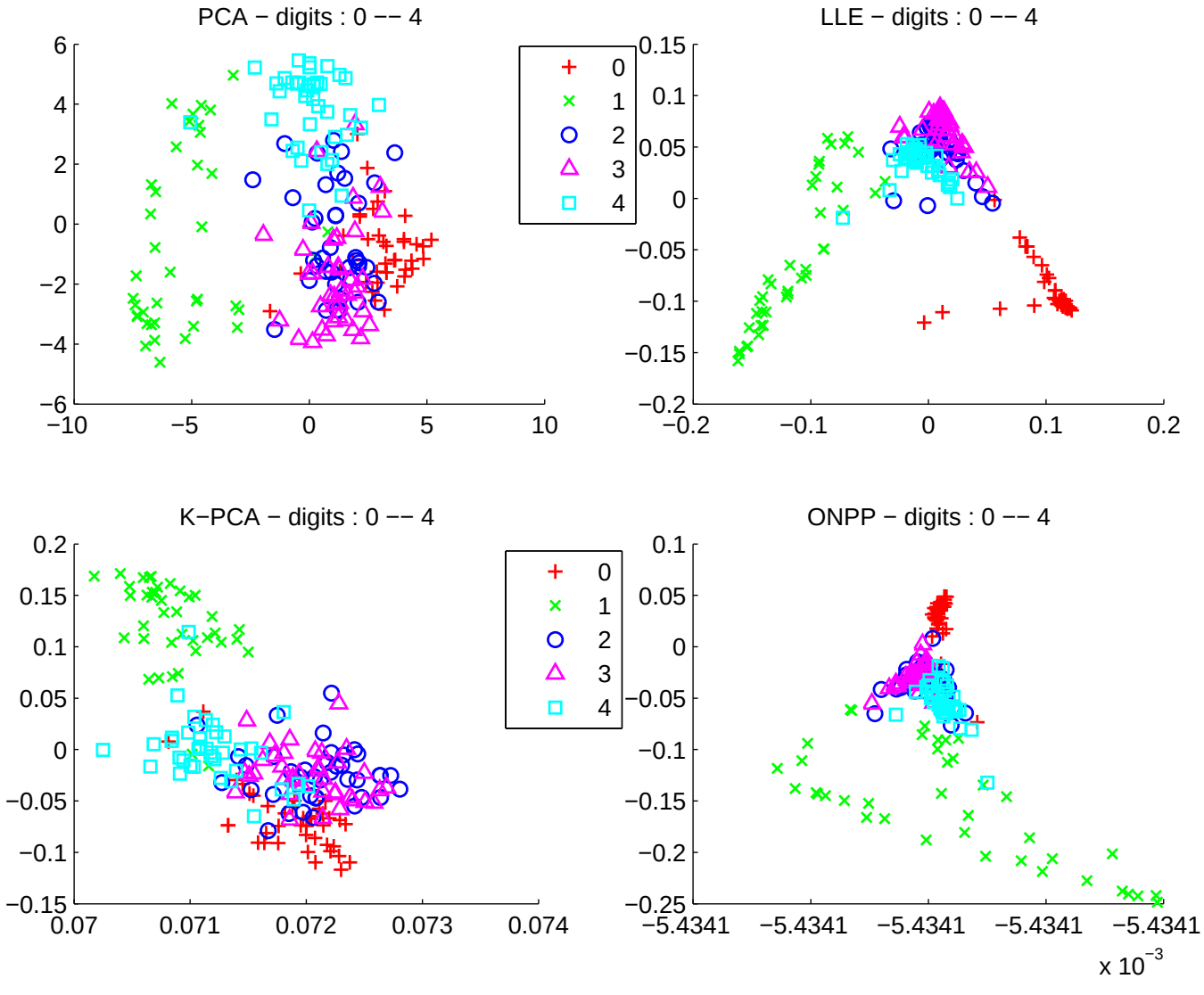




Example 2: Digit images (a random sample of 30)

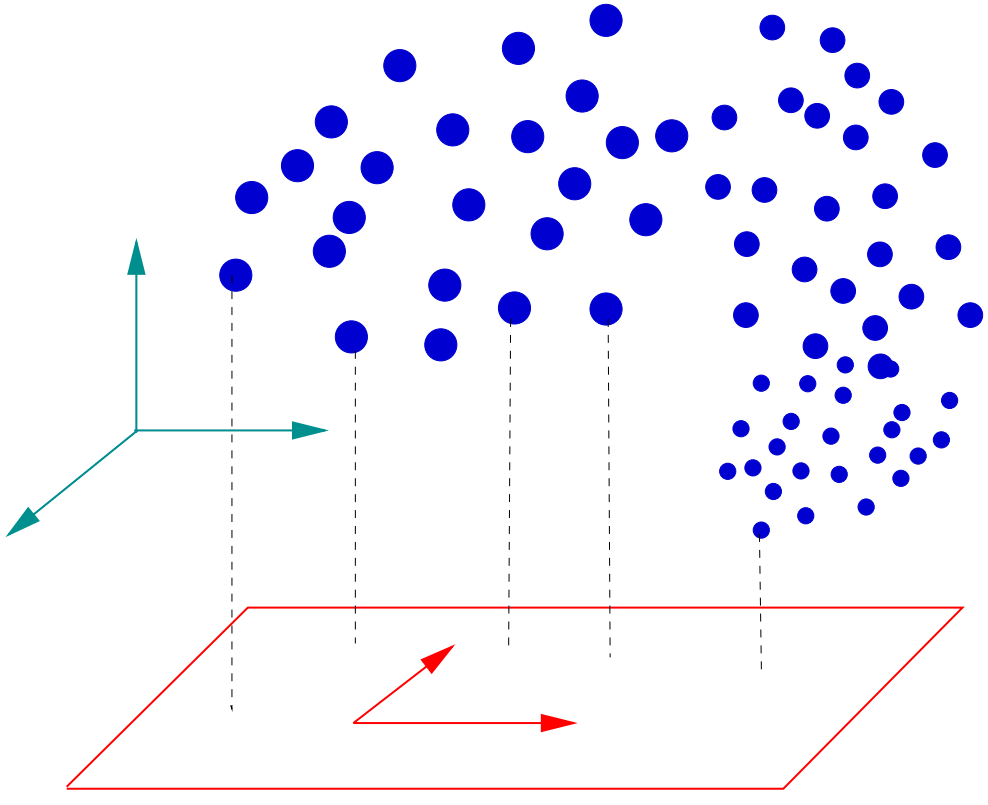


2-D 'reductions':



Basic linear dimensionality reduction: PCA

- We are given points in $\mathbb{R}^n$ and we want to project them in $\mathbb{R}^d$. Best way to do this?
- i.e.: find the best axes for projecting the data
- Q: "best in what sense"?
- A: maximize variance of new data



- Principal Component Analysis [PCA]

- Recall $\boldsymbol{y_i = V^T x_i}$, where $\boldsymbol{V}$ is $\boldsymbol{m \times d}$ orthogonal
- Need to maximize over all orthogonal $\boldsymbol{m \times d}$ matrices $\boldsymbol{V}$:

$$\boxed{\sum_i \|\boldsymbol{y_i} - \tfrac{1}{n}\sum_j \boldsymbol{y_j}\|_2^2 = \cdots = \text{Tr}\left[\boldsymbol{V}^\top \bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^\top \boldsymbol{V}\right]}$$

Where: $\bar{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$ == origin-recentered version of $\boldsymbol{X}$

- Solution $\boldsymbol{V} = \{$ dominant eigenvectors $\}$ of the covariance matrix == Set of left singular vectors of $\bar{\boldsymbol{X}}$

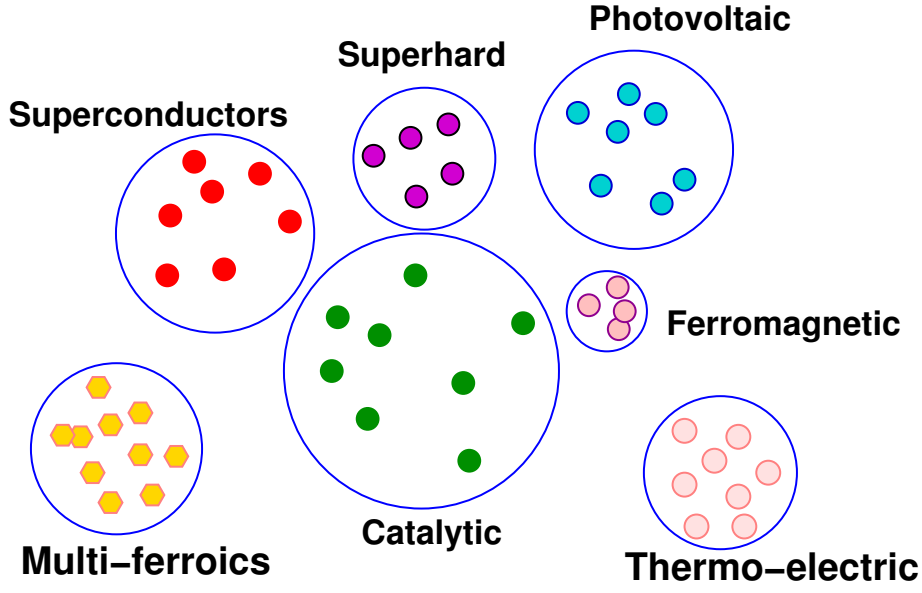
- Solution $\boldsymbol{V}$ also minimizes 'reconstruction error' ..

$$\sum_i \|\boldsymbol{x_i} - \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{x_i}\|^2 = \sum_i \|\boldsymbol{x_i} - \boldsymbol{V}\boldsymbol{y_i}\|^2$$

- Also maximizes [Korel and Carmel 04] $\sum_{i,j} \|\boldsymbol{y_i} - \boldsymbol{y_j}\|^2$

Unsupervised learning: Clustering

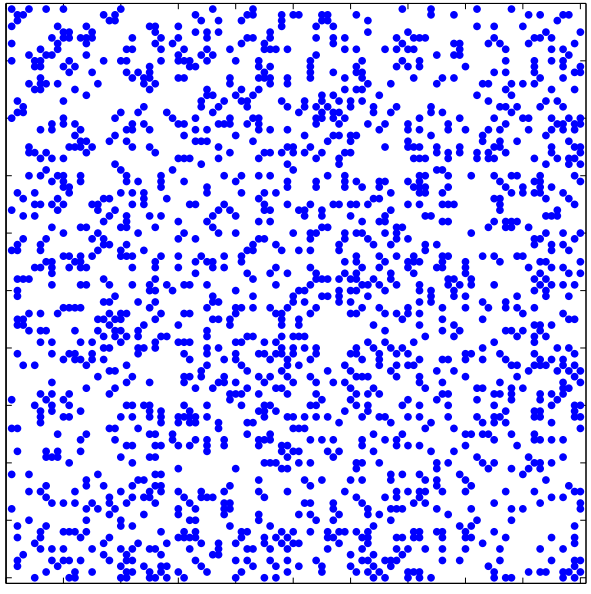
Problem: partition a given set into subsets such that items of the same subset are most similar and those of two different subsets most dissimilar.

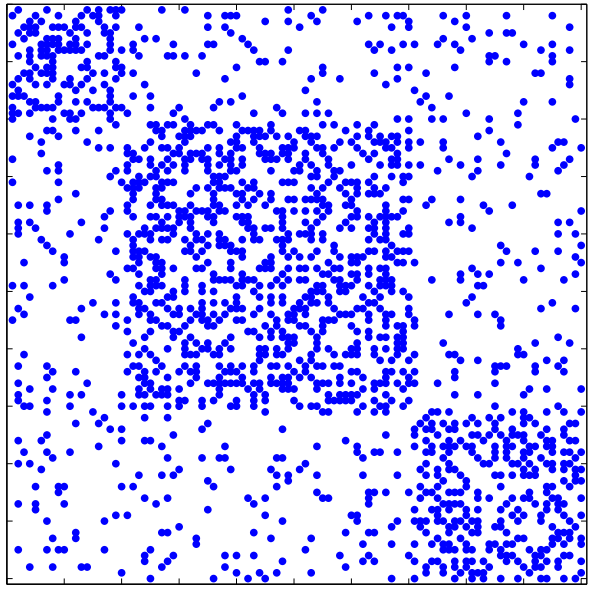


- Basic technique: K-means algorithm [slow but effective.]
- Example of application : cluster bloggers by 'social groups' (anarchists, ecologists, sports-fans, liberals-conservative, ...)

Sparse Matrices viewpoint

- 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:

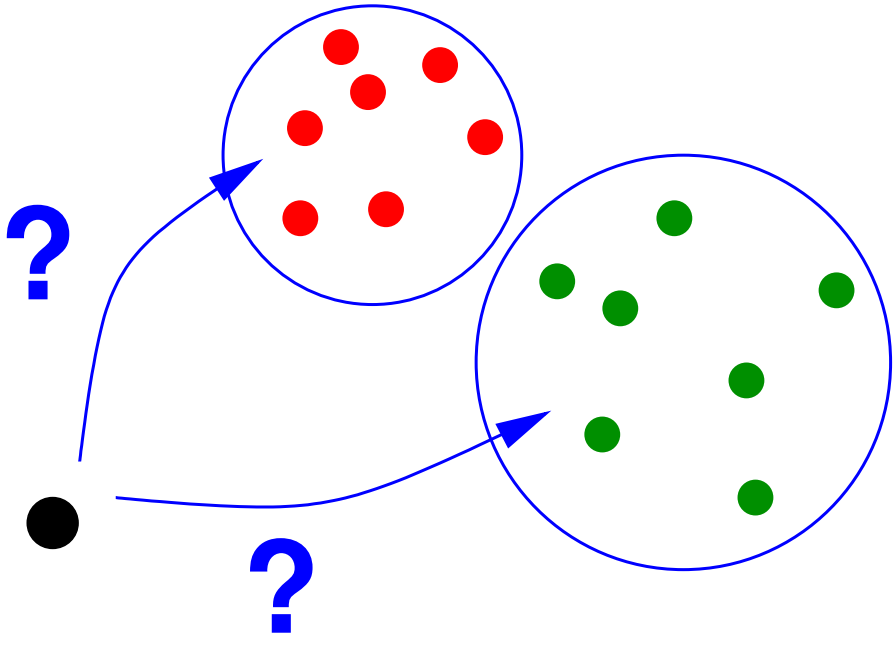




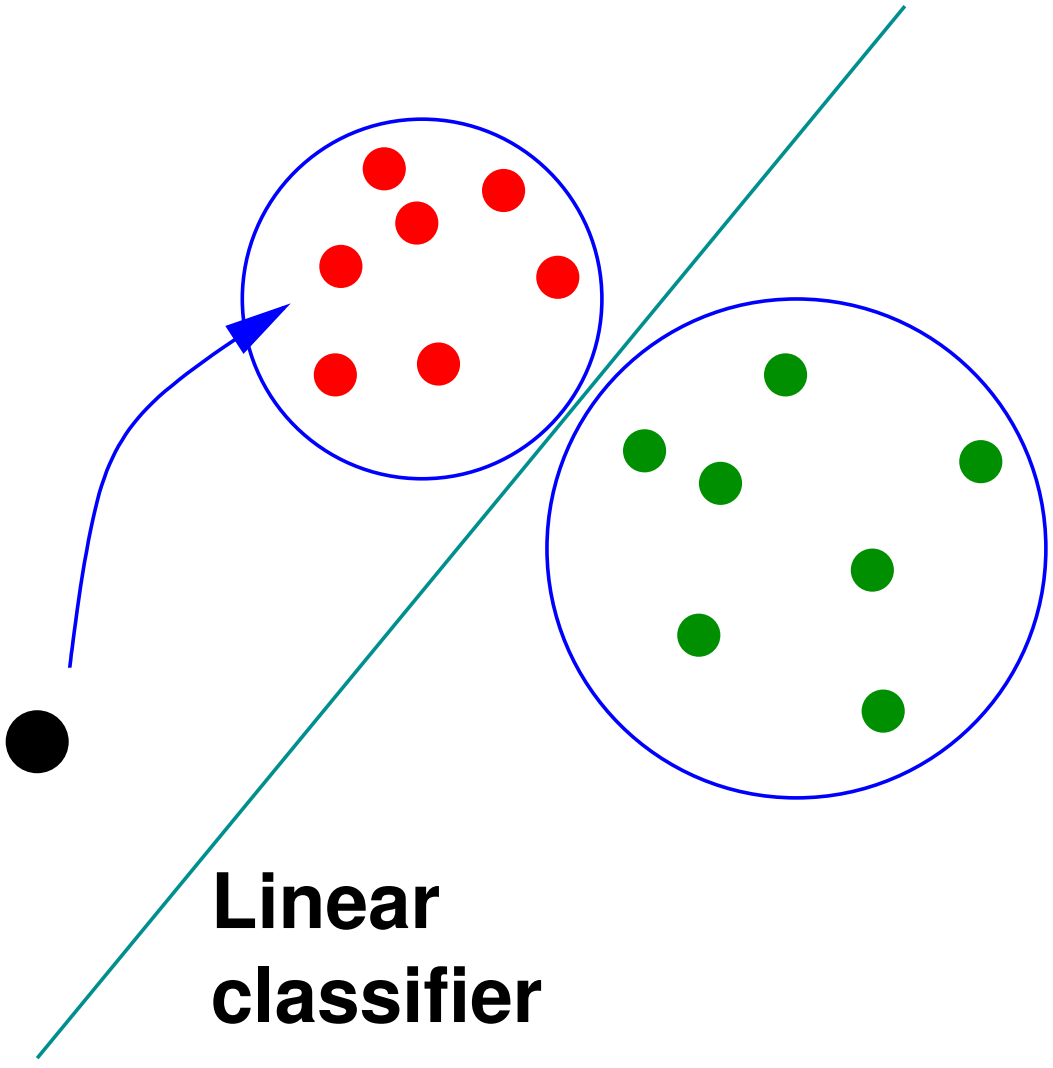
- Advantage of this viewpoint: need not know # of clusters
- Use 'blocking' techniques for sparse matrices

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.

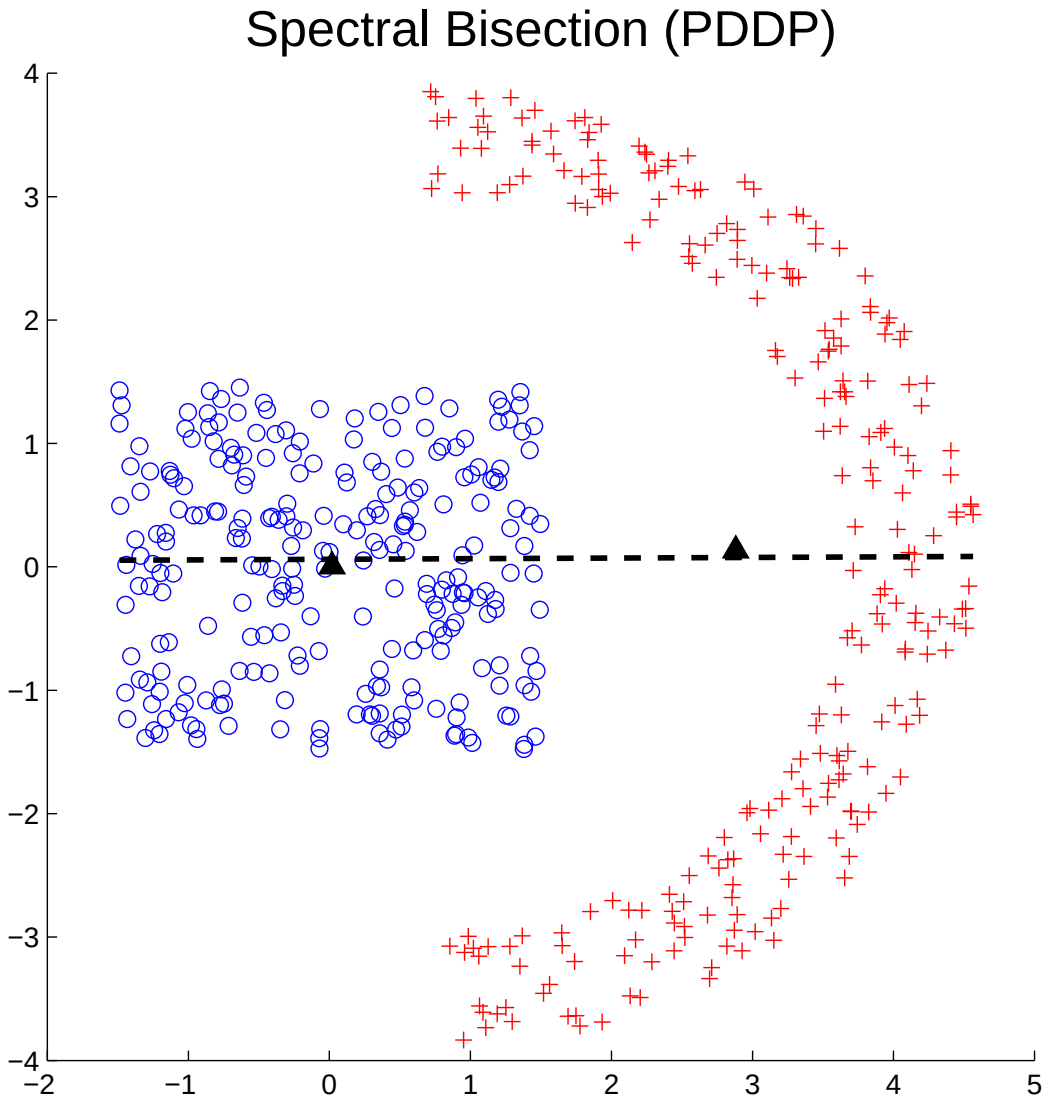


- Many applications.
- Example: distinguish SPAM and non-SPAM messages
- Can be extended to more than 2 classes.

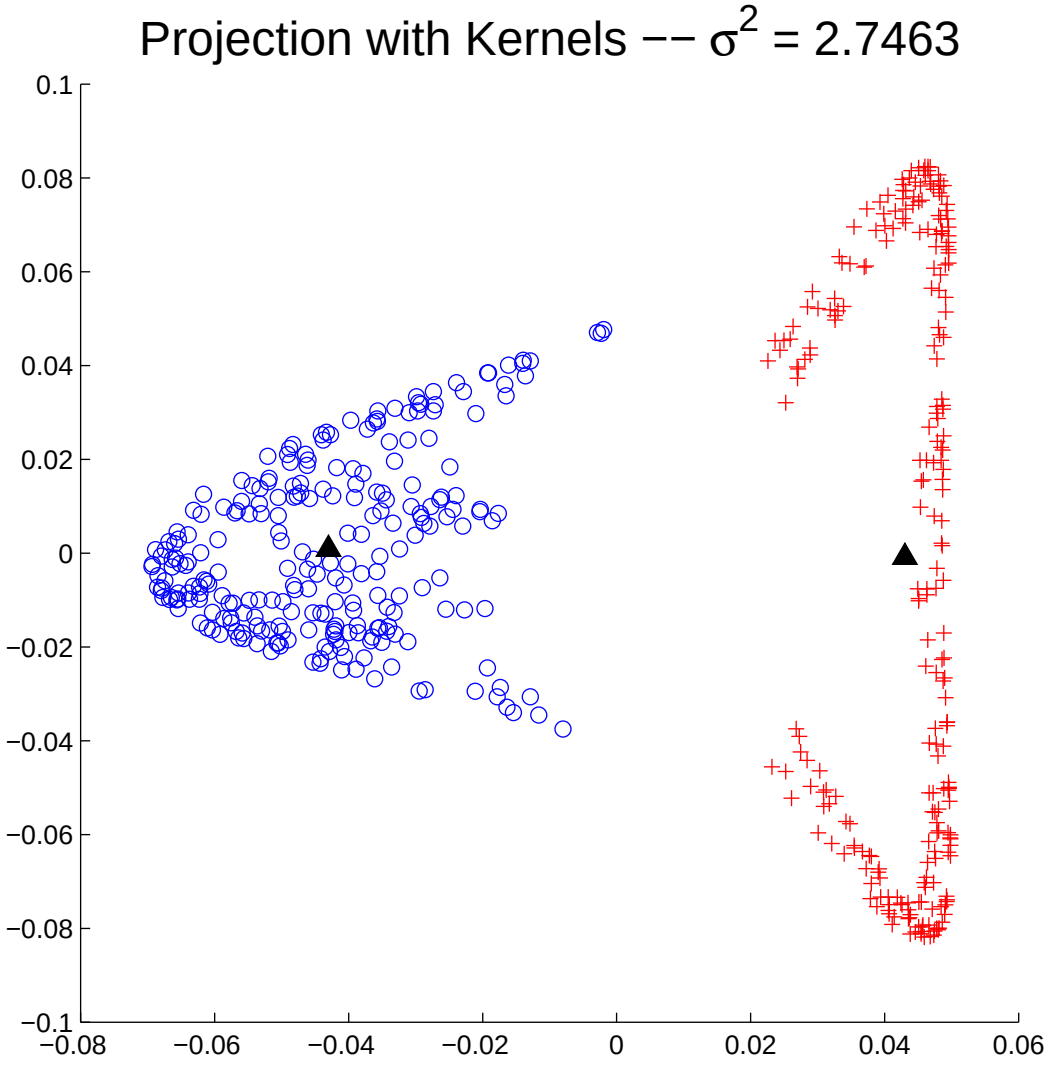


Linear classifiers: Find a hyperplane which best separates the data in classes A and B.

A harder case:



- Use kernels to transform

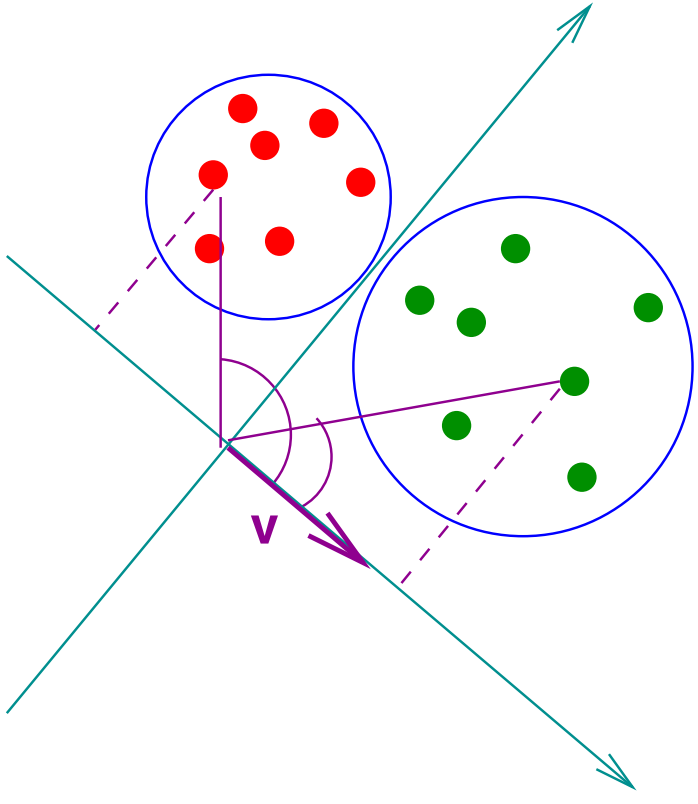


Transformed data with a Gaussian Kernel

Linear classifiers

- Let $X = [x_1, \cdots, x_n]$ be the data matrix.
- and $L = [l_1, \cdots, l_n]$ the labels either +1 or -1.

- 1st Solution: Find a vector v such that $v^T x_i$ close to $l_i \; \forall i$
- Common solution: SVD to reduce dimension of data [e.g. 2-D] then do comparison in this space. e.g. A: $v^T x_i \geq 0$, B: $v^T x_i < 0$



[Note: v principal axis drawn below where it should be]

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.

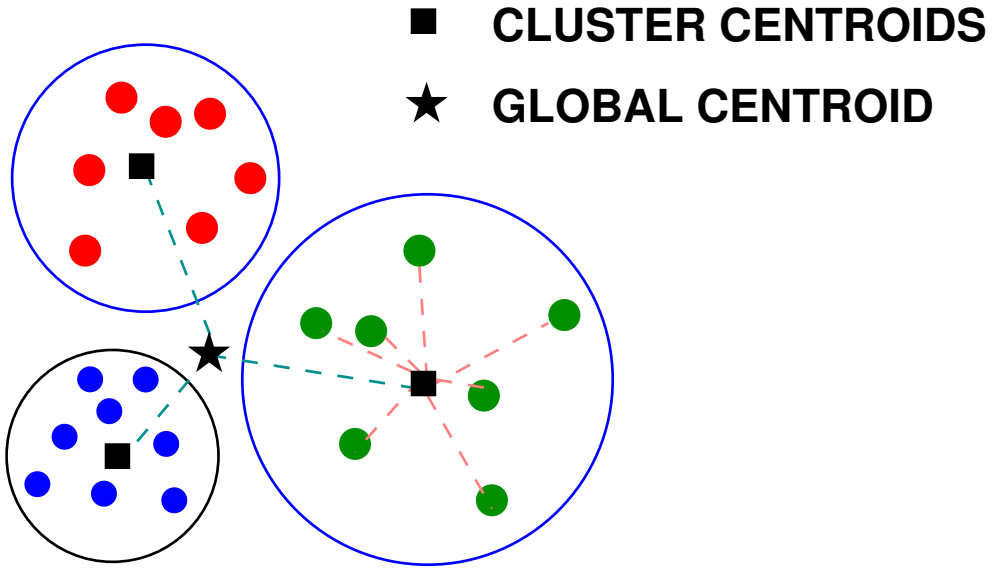
➤ Goal: to make "between scatter" measure large, while making "within scatter" small.

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 – Extension to arbitrary dimension

- Would like to project in d dimensions –
- Normally we wish to maximize the ratio of two traces:

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

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

Difficulty: Hard to maximize. See Bellalij & YS (in progress)

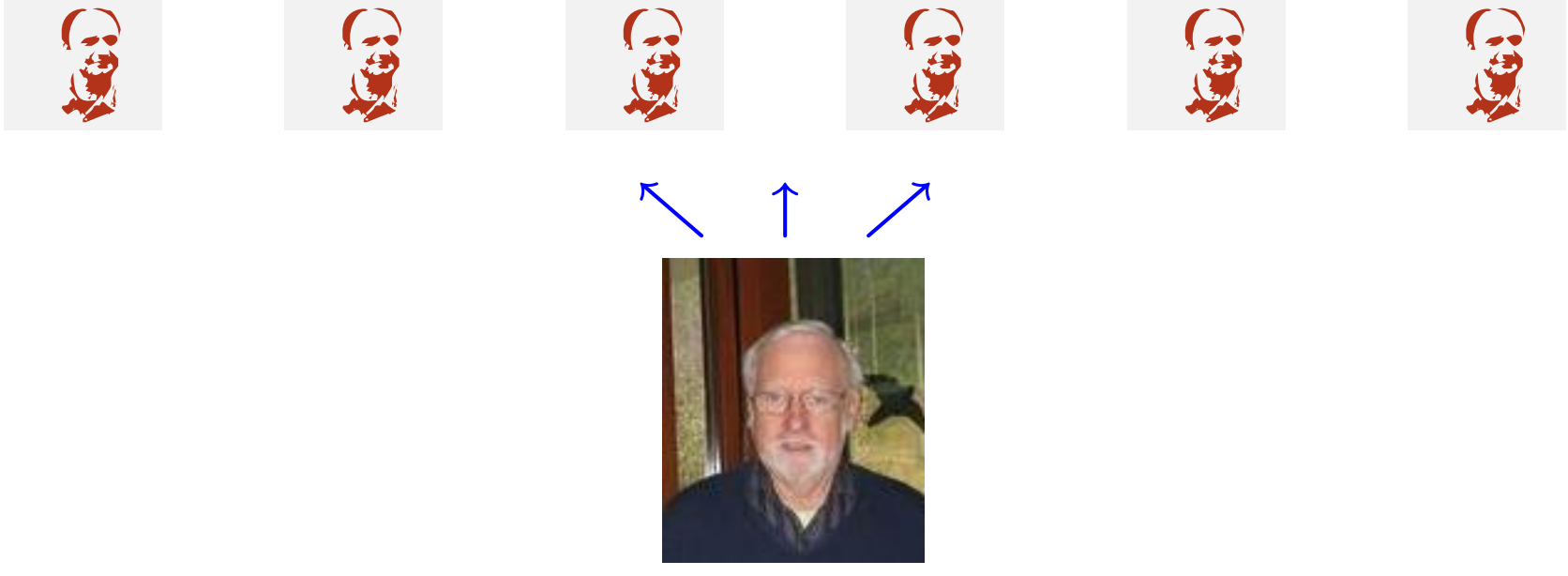
- Common alternative: Solve instead the (easier) problem:

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

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

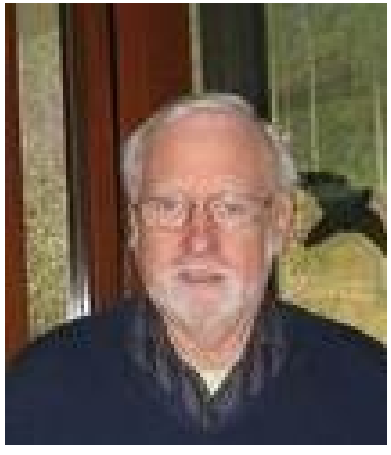
Face Recognition – background

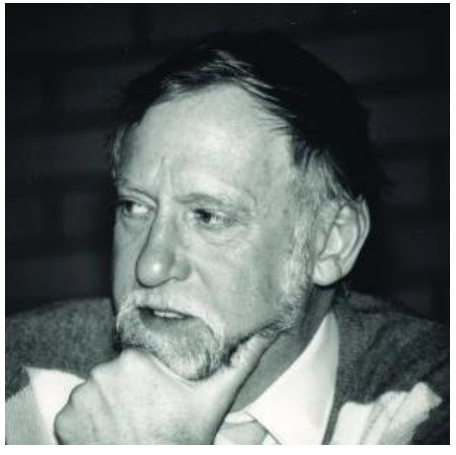
Problem: We are given a database of images: [arrays of pixel values]. And a test (new) image.



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

Difficulty Positions, Expressions, Lighting, ...,



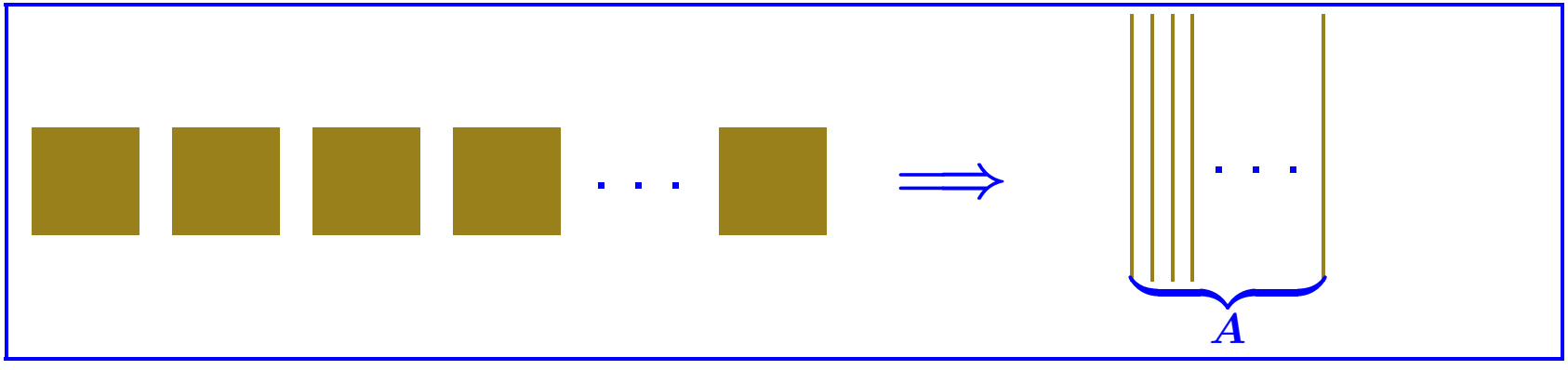
Eigenfaces: Principal Component Analysis technique

- 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

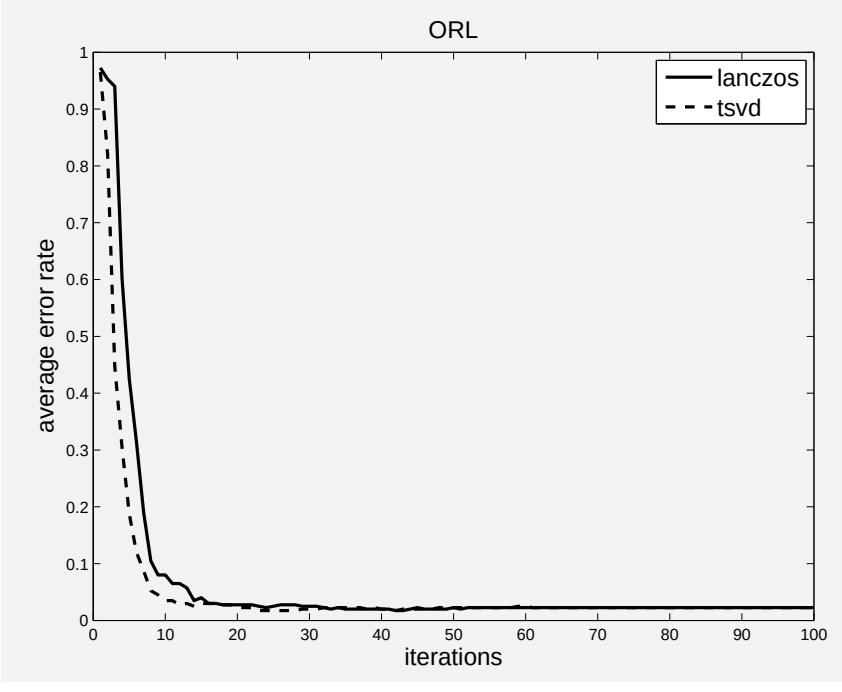
AR set 126 subjects – 4 facial expressions selected for each [natural, smiling, angry, screaming] – example:

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

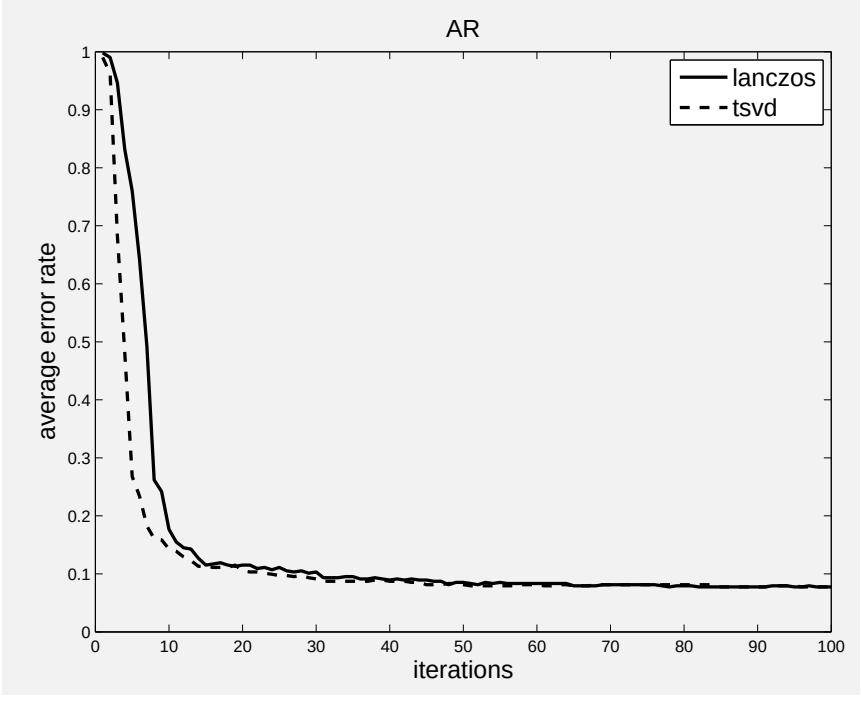
Tests: Face Recognition

Recognition accuracy of Lanczos approximation vs SVD

ORL dataset



AR dataset



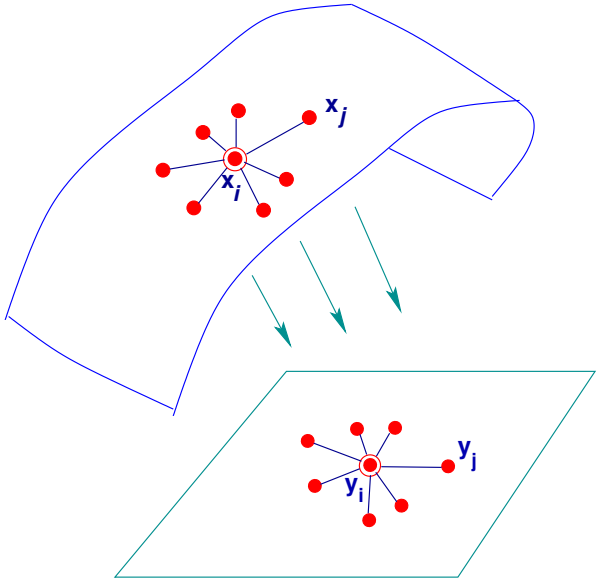
Vertical axis shows average error rate. Horizontal = Subspace dimension

GRAPH-BASED TECHNIQUES

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)

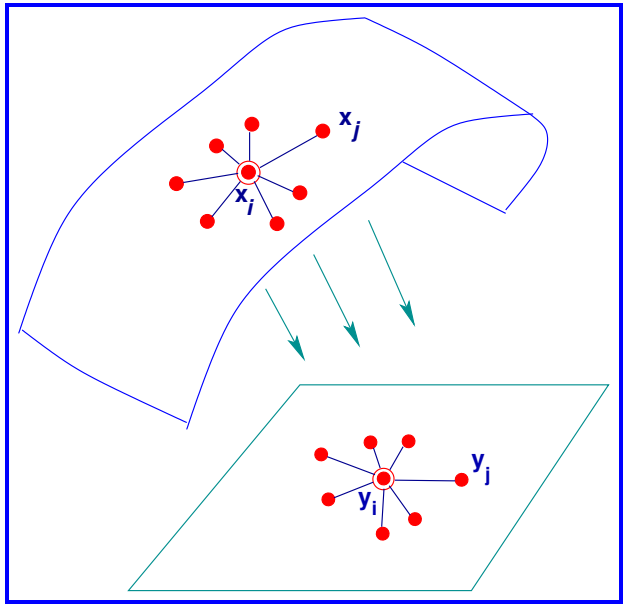
The Laplacean eigenmaps approach

Laplacean Eigenmaps *minimizes*

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

Notes:

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

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

- An $n \times n$ sparse eigenvalue problem [In 'sample' space]
- Note: can assume $\boldsymbol{D} = \boldsymbol{I}$. Amounts to rescaling data.

Problem becomes

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

A unified view

- Most techniques lead to one of two types of problems

First :

- Y results directly from computing eigen-vectors
- LLE, Eigenmaps, ...

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

Second:

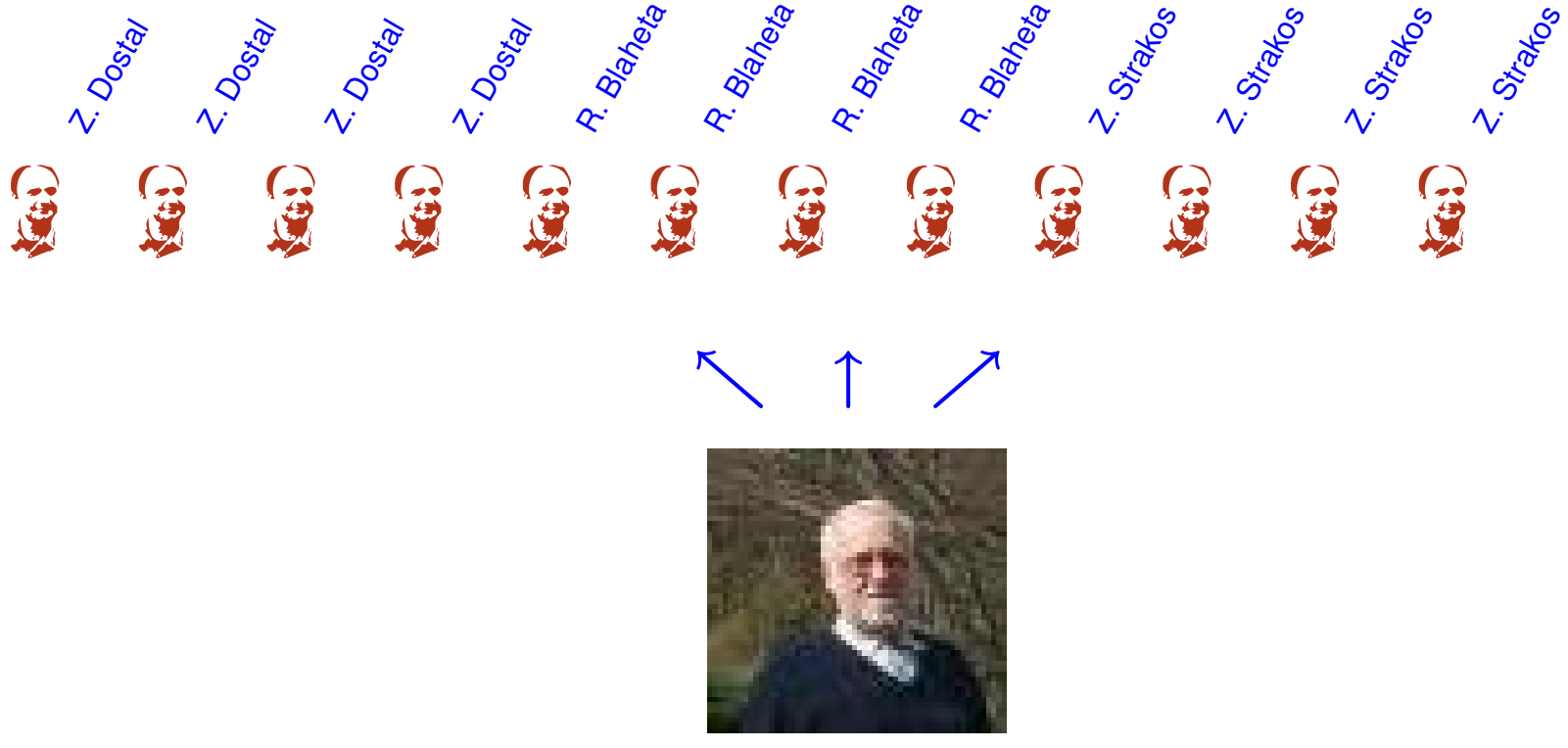
- Low-Dimens. data: $Y = V^\top X$
- G == identity, or $XDX^\top$, or $XX^\top$

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

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

Graph-based methods in a supervised setting

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



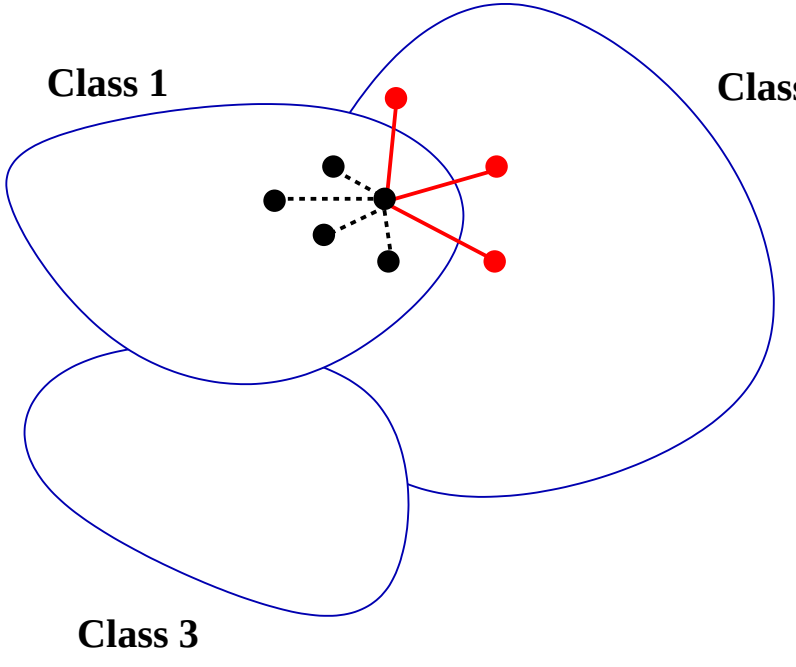
Question: Find label (best match) for test image.

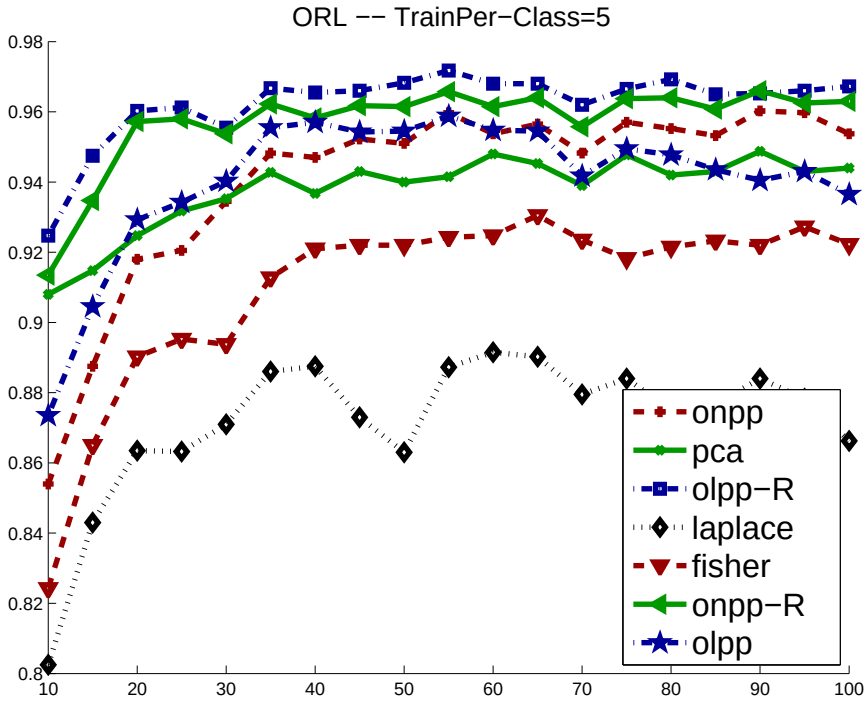
Methods can be adapted to supervised mode by building the graph to use class labels. Idea: Build G so that nodes in the same class are neighbors. If c = # classes, G consists of c cliques.

- Matrix W is block-diagonal
- Note: $\mathbf{rank}(W) = n - c$.

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

- Can be used for LPP, ONPP, etc..
- Recent improvement: add repulsion Laplacean [Kokiopoulou, YS 09]


Class 1
Class 2
Class 3


ORL −− TrainPer−Class=5
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
10
20
30
40
50
60
70
80
90
100
onpp
pca
olpp−R
laplace
fisher
onpp−R
olpp

ELECTRONIC STRUCTURE CALCULATIONS

Electronic structure and Schrödinger's equation

➤ 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 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.

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: An equation of the form

$$\left[-\frac{h^2}{2m}\nabla^2 + v_0(r) + \int \frac{\rho(r')}{|r-r'|}dr' + \frac{\delta E_{xc}}{\delta \rho}\right]\Psi = E\Psi$$

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

Kohn-Sham equations → nonlinear eigenvalue Pb

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

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

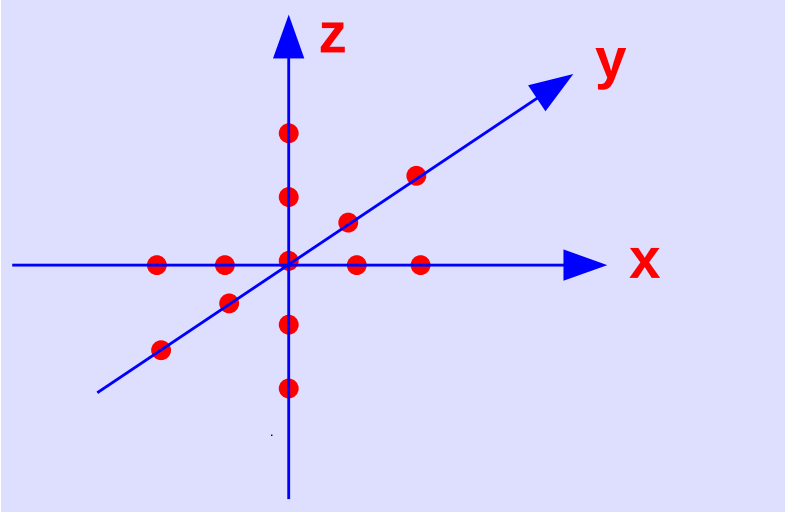
$$\nabla^2 V_H = -4\pi\rho(r)$$

- 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

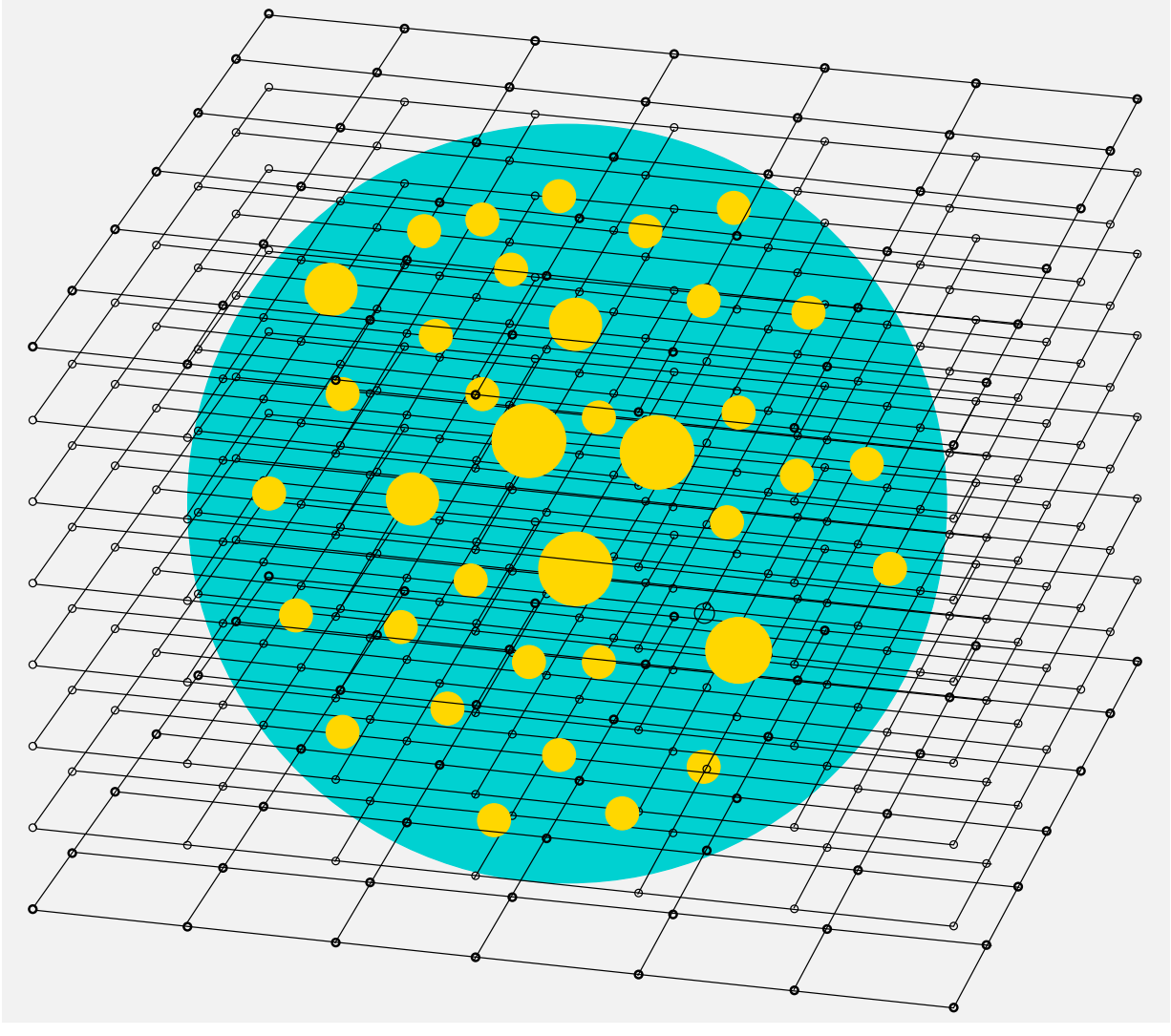
Real-space Finite Difference Methods

- 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.

Diagonalization

Note:

Standard packages (ARPACK) do not take advantage of specificity of problem: self-consistent loop, large number of eigenvalues, ...

Observations made: for efficiency it is important to

- Focus on eigen-space - not individual eigenvectors.
- Take outer (SCF) loop into account

- Future: eigenvector-free or basis-free methods or
- .. 'spectrum slicing' methods

CHEBYSHEV FILTERING

Chebyshev Subspace iteration

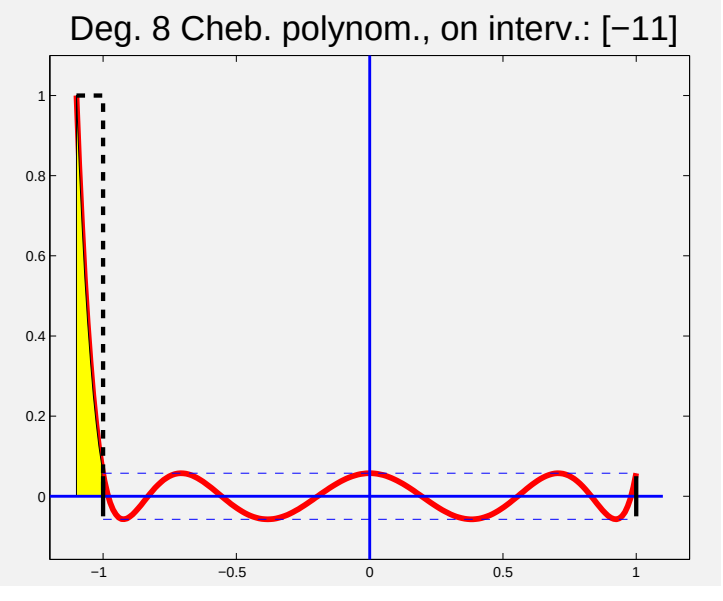
- Main ingredient: Chebyshev filtering

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. Enhances wanted eigencomponents

The filtering step is not used to compute eigenvectors accurately ➤
SCF & diagonalization loops merged
Important: convergence still good and robust



Main step:

$$
\begin{array}{ll}
\text{Previous basis} & V = [v_1, v_2, \cdots, v_m] \\
& \downarrow \\
\text{Filter} & \hat{V} = [p(A)v_1, p(A)v_2, \cdots, p(A)v_m] \\
& \downarrow \\
\text{Orthogonalize} & [V, R] = qr(\hat{V}, 0)
\end{array}
$$

- The basis V is used to do a Ritz step (basis rotation)

$$C = V^T A V \rightarrow [U, D] = eig(C) \rightarrow V := V * U$$

- Update charge density using this basis.
- Update Hamiltonian — repeat
- In effect: Nonlinear subspace iteration

➤ Main advantages: (1) very inexpensive, (2) uses minimal storage (m is a little $\geq$ # states).

➤ 3-term recurrence of Chebyshev polynommial exploited to compute $p_k(A)v$.

Reference:

Yunkai Zhou, Y.S., Murilo L. Tiago, and James R. Chelikowsky, Parallel Self-Consistent-Field Calculations with Chebyshev Filtered Subspace Iteration, Phy. Rev. E, vol. 74, p. 066704 (2006).

[See http://www.cs.umn.edu/∼saad]

Chebyshev Subspace iteration - experiments

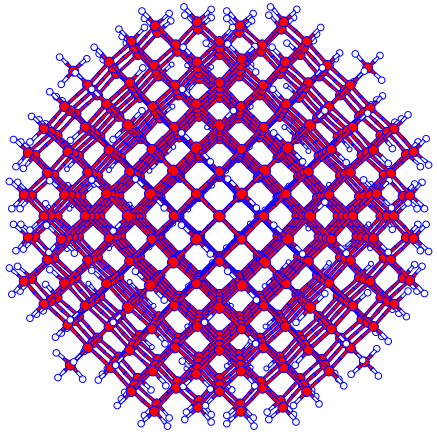
- A large calculations: $Si_{9041}H_{1860}$, using 48 processors.

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

Pol_deg. = 17 For first iteration, 8 for CheFS.

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



Data mining for materials: Materials Informatics

- Huge potential in exploiting two trends:

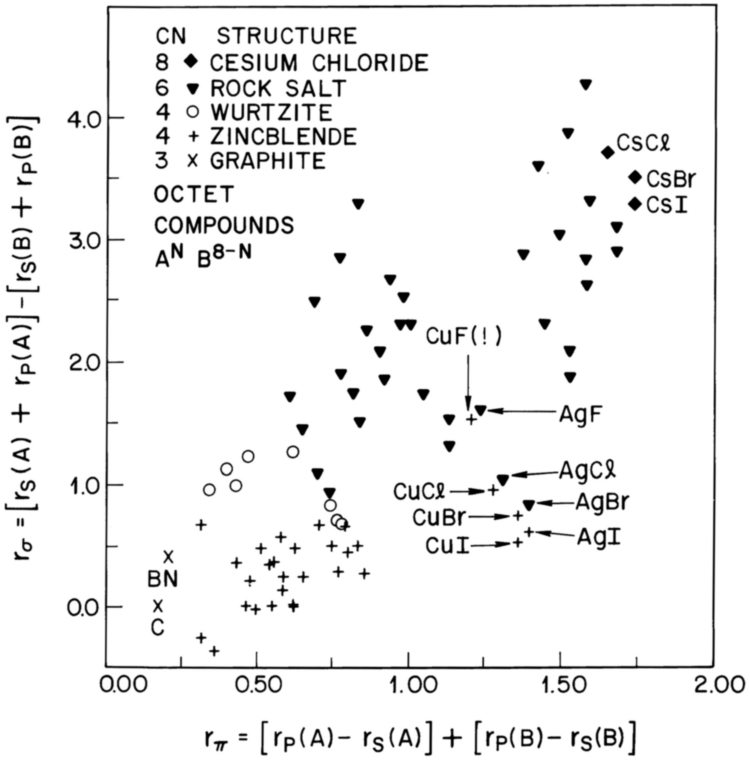
1 Enormous improvements in efficiency and capabilities in computational methods for materials

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.

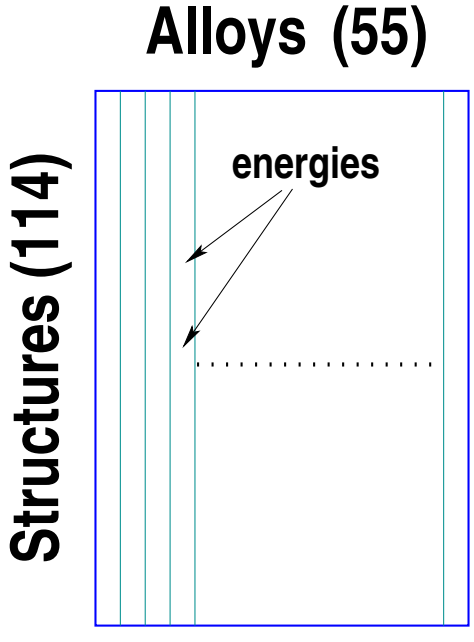
Example 2 : [Curtalano et al. PRL vol 91, 1003]

➤ 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 !