UNIVERSITY OF MINNESOTA TWIN CITIES

Dimension reduction methods: Algorithms and Applications

Yousef Saad

Department of Computer Science and Engineering

University of Minnesota

University of Padua

June 7, 2018

Introduction, background, and motivation

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

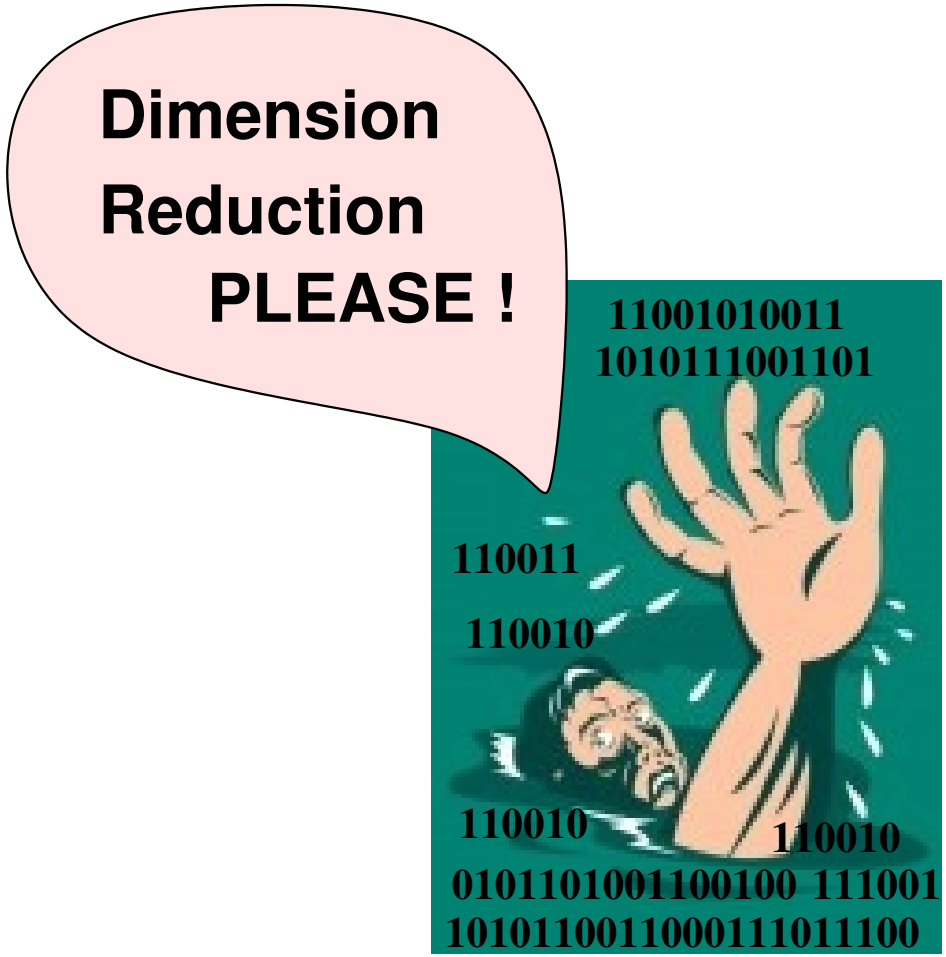
➤ Main tools used: linear algebra; graph theory; approximation theory; optimization; ...

➤ In this talk: emphasis on dimension reduction techniques and the interrelations between techniques

Introduction: a few factoids

- We live in an era increasingly shaped by 'DATA'
  - $\approx 2.5 \times 10^{18}$ bytes of data created in 2015
  - 90 % of data on internet created since 2016
  - 3.8 Billion internet users in 2017.
  - 3.6 Million Google searches worldwide / minute (5.2 B/day)
  - 15.2 Million text messages worldwide / minute
- Mixed blessing: Opportunities & big challenges.
- Trend is re-shaping & energizing many research areas ...
- ... including : numerical linear algebra

Drowning in data



Picture modified from http://www.123rf.com/photo_7238007_man-drowning-reaching-out-for-help.html

Major tool of Data Mining: Dimension reduction

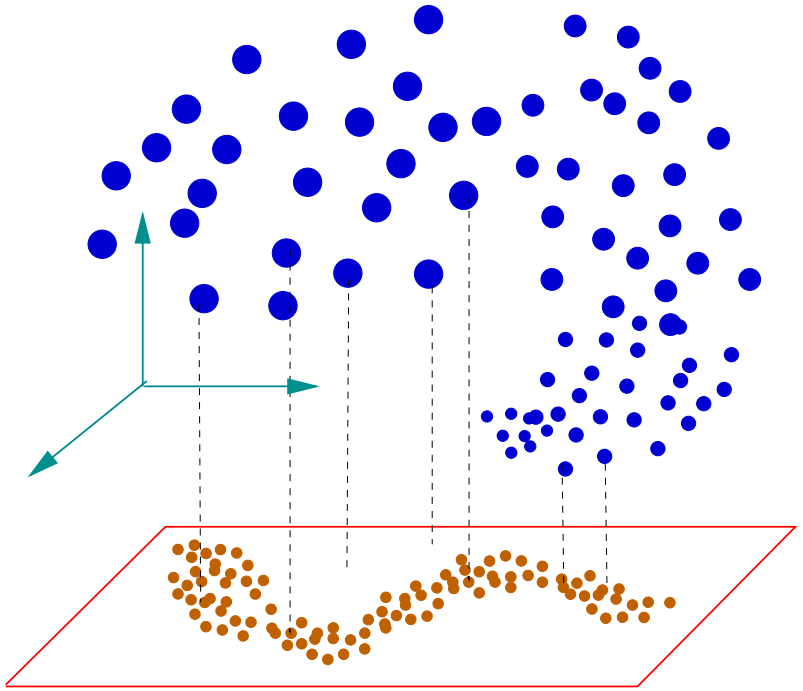
➤ Goal is not as much to reduce size (& cost) but to:

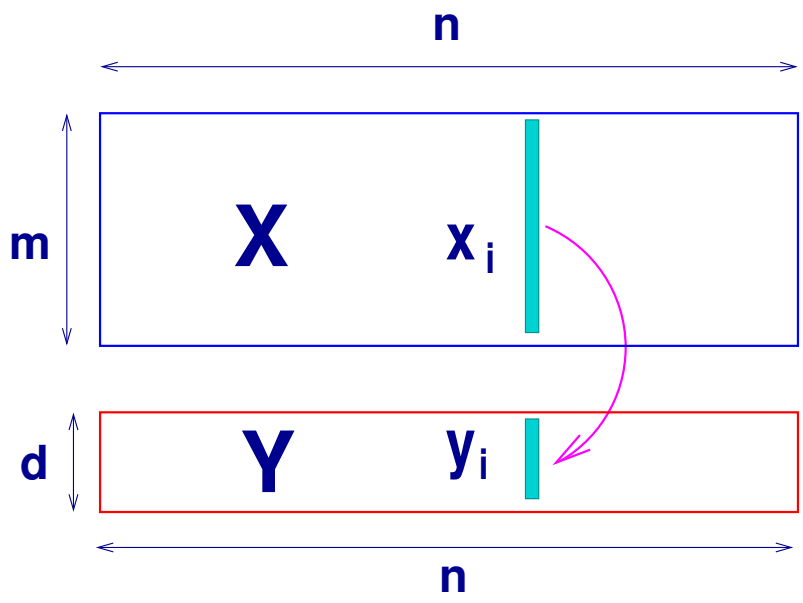
- Reduce noise and redundancy in data before performing a task [e.g., classification as in digit/face recognition]
- Discover important 'features' or 'paramaters'

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 : $y_j = W^\top x_j,\ \forall j$, or, $Y = W^\top X$
- ... or nonlinear (implicit).
- Mapping Φ required to: Preserve proximity? Maximize variance? Preserve a certain graph?

Basics: Principal Component Analysis (PCA)

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

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

- 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 $\bar{\boldsymbol{X}}=\boldsymbol{X}-\boldsymbol{\mu}\boldsymbol{e}^\top$

SVD:

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

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

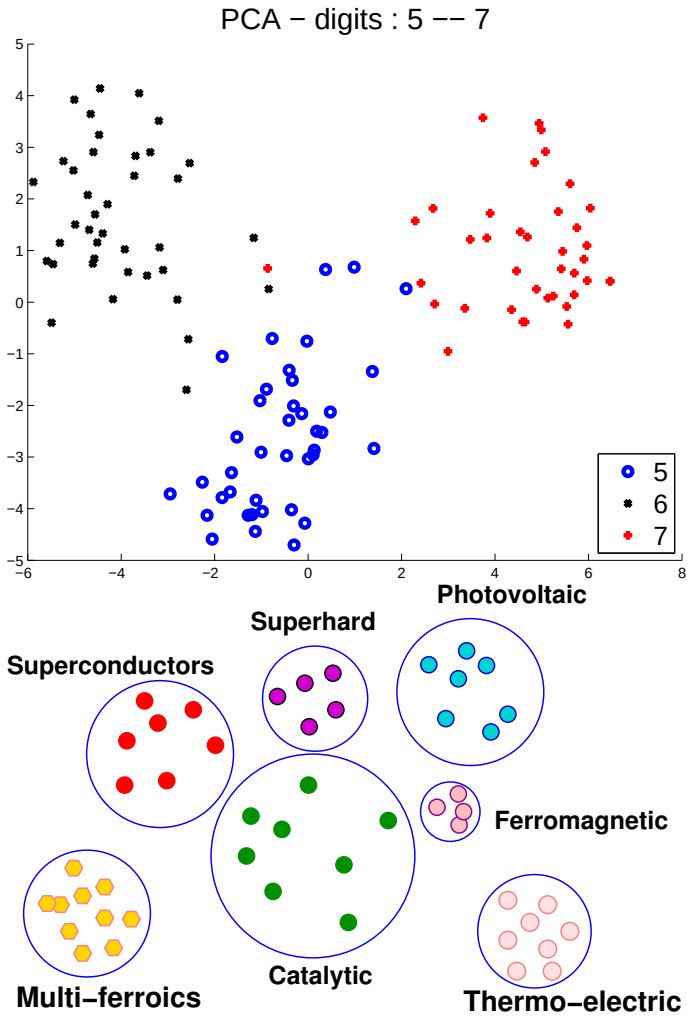
$$\sum_i \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_i\|^2 = \sum_i \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{y}_i\|^2$$

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

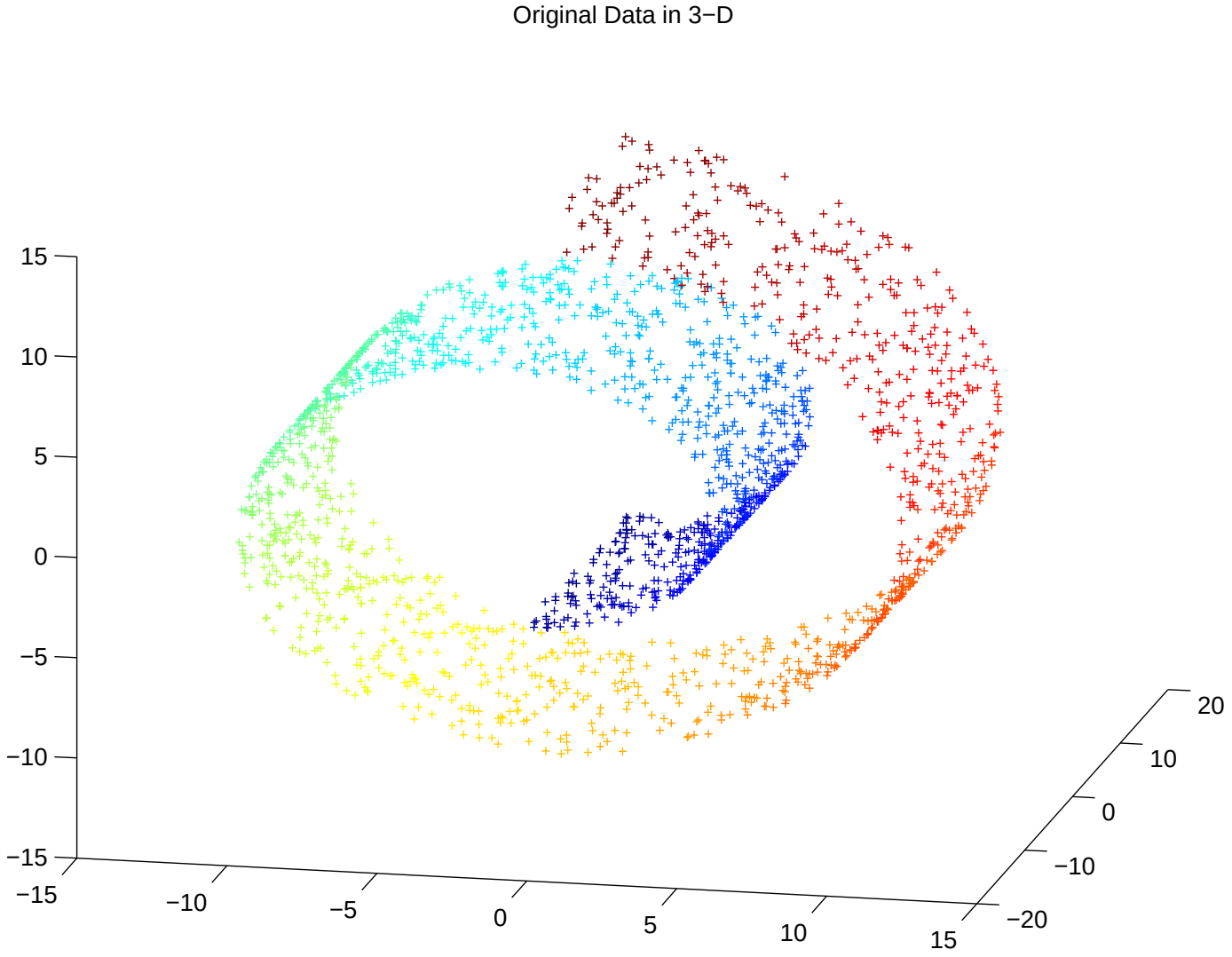
Unsupervised learning

"Unsupervised learning" : methods do not exploit labeled data

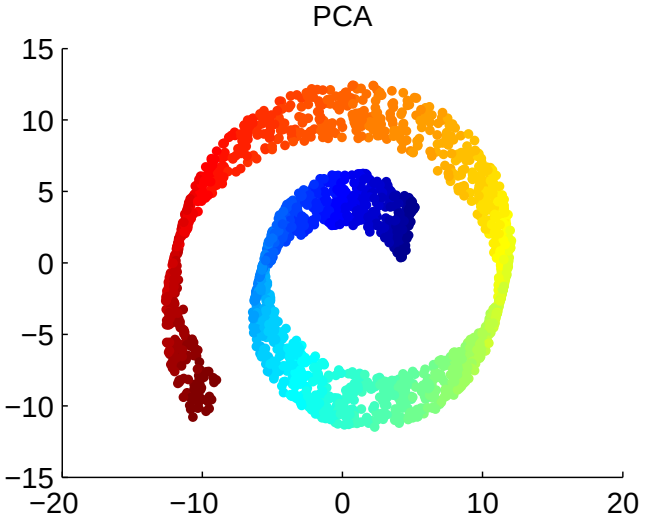
- Example of digits: perform a 2-D projection
- Images of same digit tend to cluster (more or less)
- Such 2-D representations are popular for visualization
- Can also try to find natural clusters in data, e.g., in materials
- Basic clusterning technique: K-means

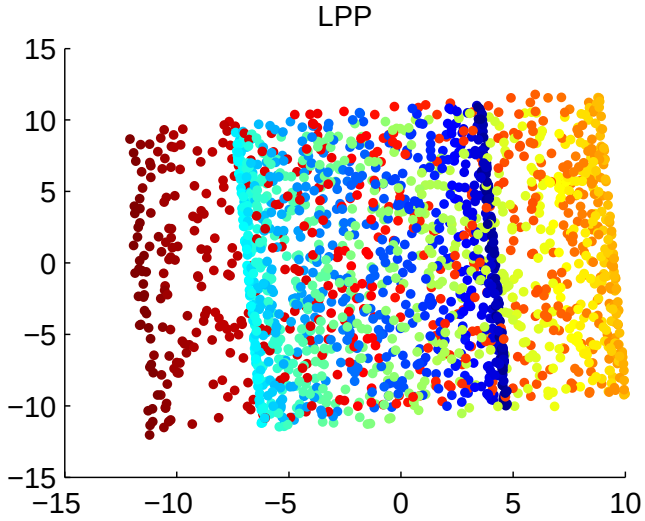


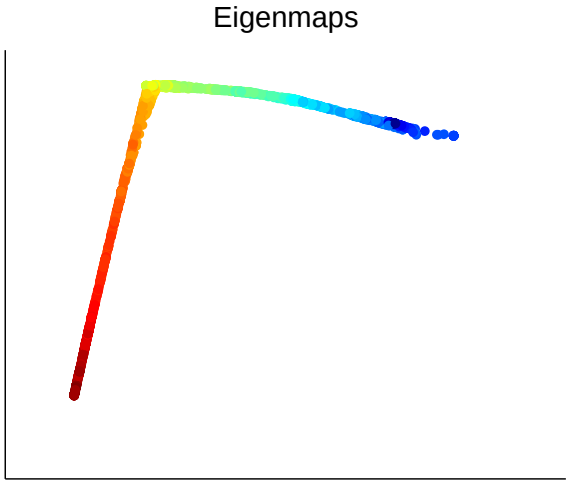
Example: The 'Swill-Roll' (2000 points in 3-D)

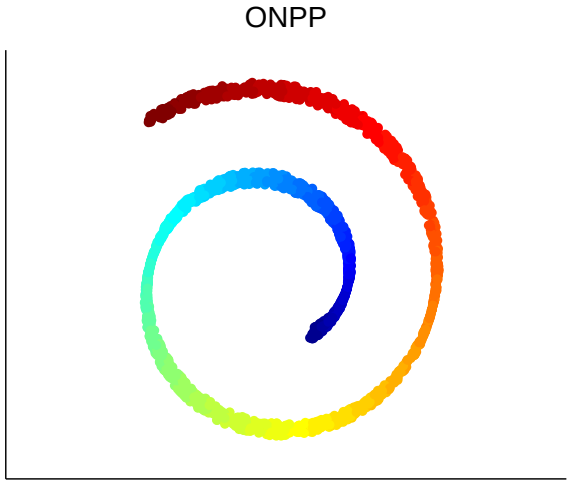


2-D ‘reductions’:

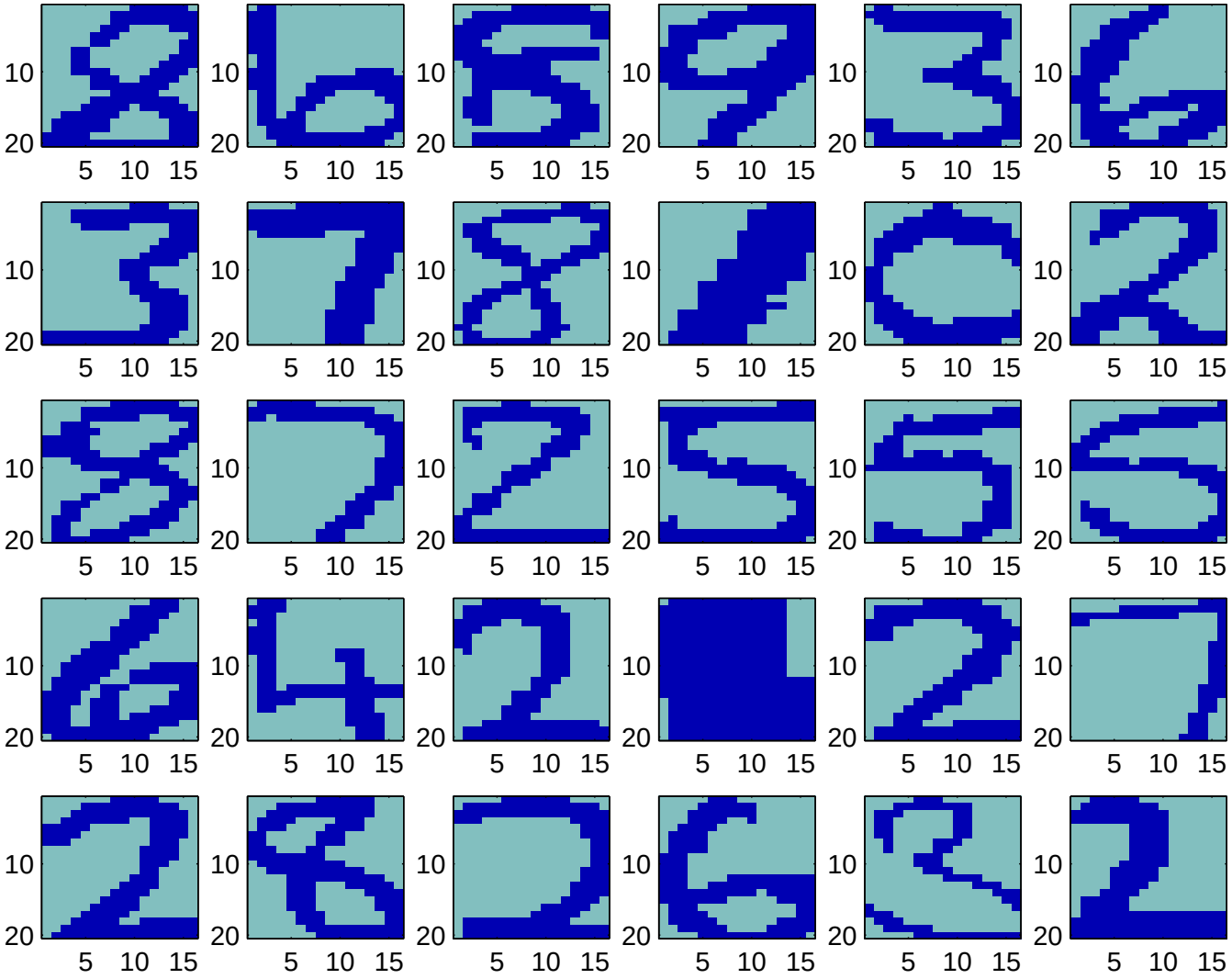




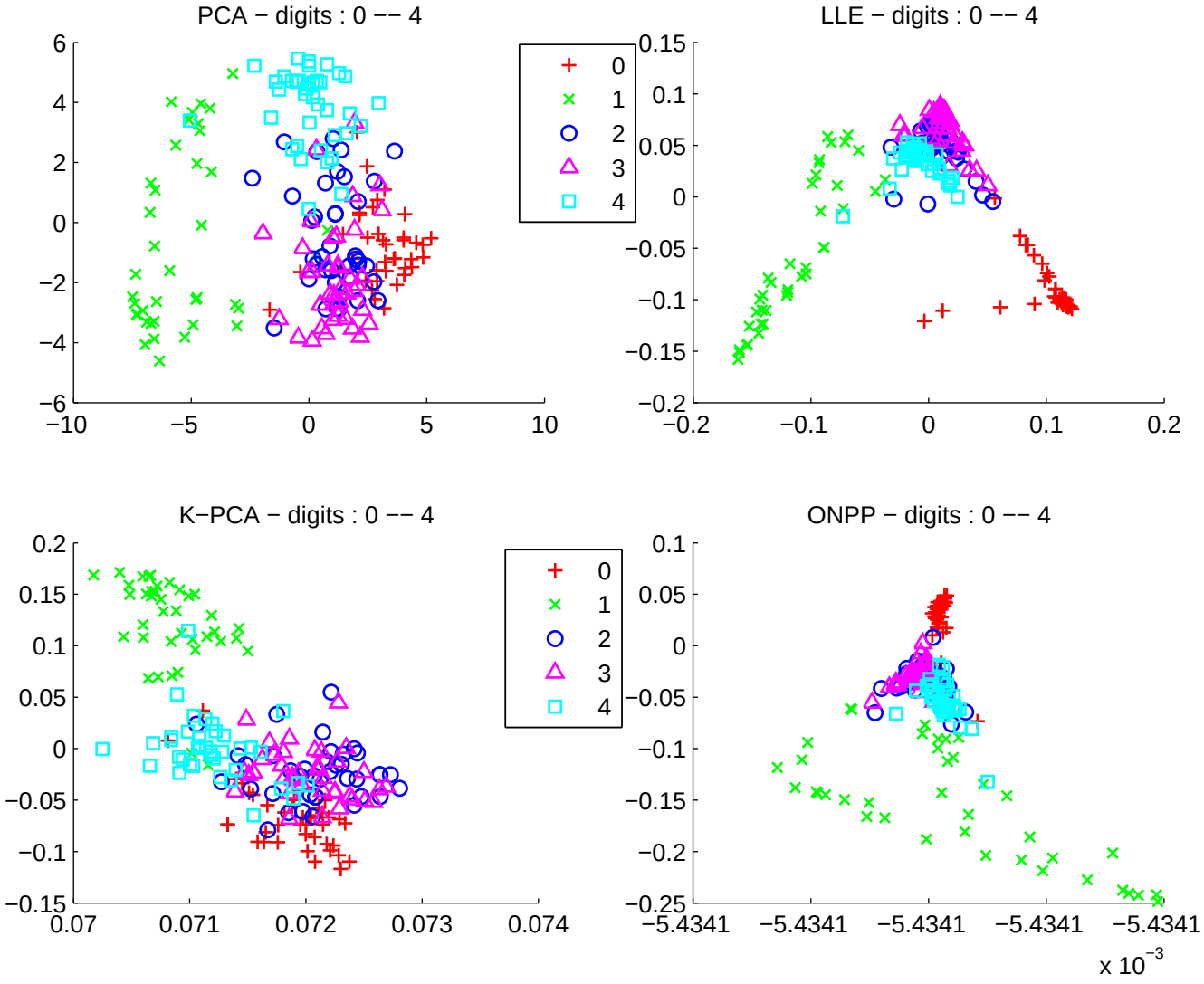




Example: Digit images (a random sample of 30)



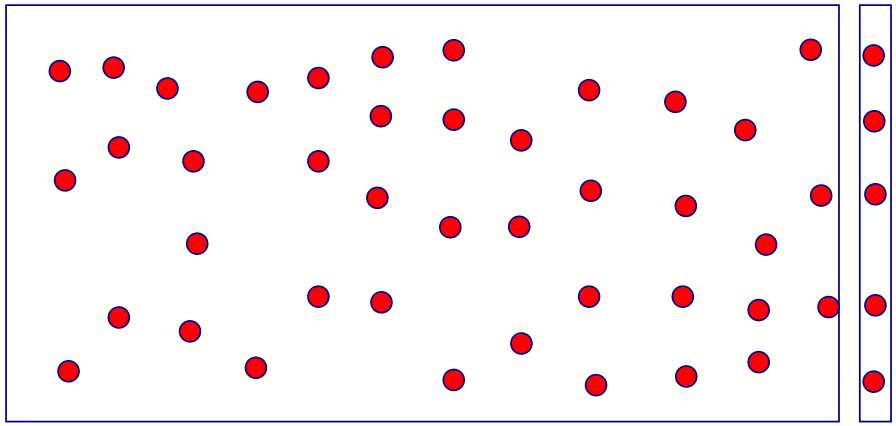
2-D 'reductions':



DIMENSION REDUCTION EXAMPLE: 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: $\cos$ of angle between a column of A and q

$$\frac{|q^T A(:,j)|}{\|q\|_2 \, \|A(:,j)\|_2}$$

- To rank all documents we need to compute

$$s = q^T A$$

- s = similarity (row) 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: $s_k = q^T A_k = q^T U_k \Sigma_k V_k^T$
- Called Truncated SVD or TSVD in context of regularization
- Main issues: 1) computational cost 2) Updates

Use of polynomial filters

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

Consider the step-function (Heaviside):

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

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

* See: E. Kokiopoulou & YS '04

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.

Background: The Lanczos procedure

- Let A an $n \times n$ symmetric matrix

ALGORITHM : 1. *Lanczos*

1. *Choose vector* v_1 *with* $\|v_1\|_2 = 1$. *Set* $\beta_1 \equiv 0, v_0 \equiv 0$
2. *For* $j = 1, 2, \ldots, m$ *Do:*
3. $\alpha_j := v_j^T A v_j$
4. $\boxed{w := A v_j - \alpha_j v_j - \beta_j v_{j-1}}$
5. $\beta_{j+1} := \|w\|_2$. *If* $\beta_{j+1} = 0$ *then Stop*
6. $v_{j+1} := w/\beta_{j+1}$
7. *EndDo*

- Note: Scalars β_{j+1}, α_j, selected so that $v_{j+1} \perp v_j$, $v_{j+1} \perp v_{j-1}$, and $\|v_{j+1}\|_2 = 1$.

Background: Lanczos (cont.)

- Lanczos recurrence:

$$\beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}$$

- Let $V_m = [v_1, v_2, \cdots, v_m]$. Then we have:

$$V_m^T A V_m = T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{bmatrix}$$

- In theory $\{v_j\}$'s are orthonormal. In practice need to re-orthogonalize

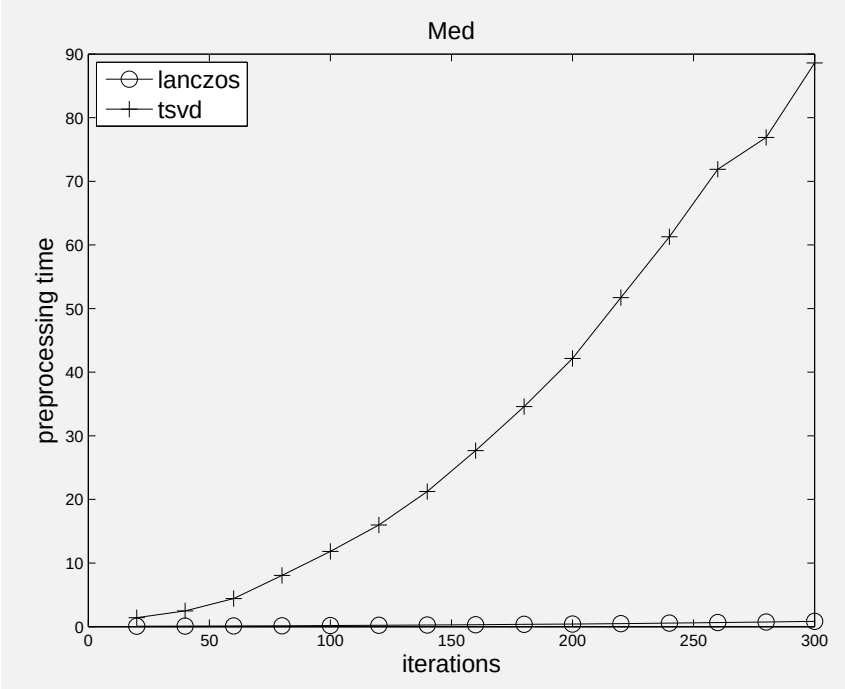
Tests: IR

Information retrieval datasets

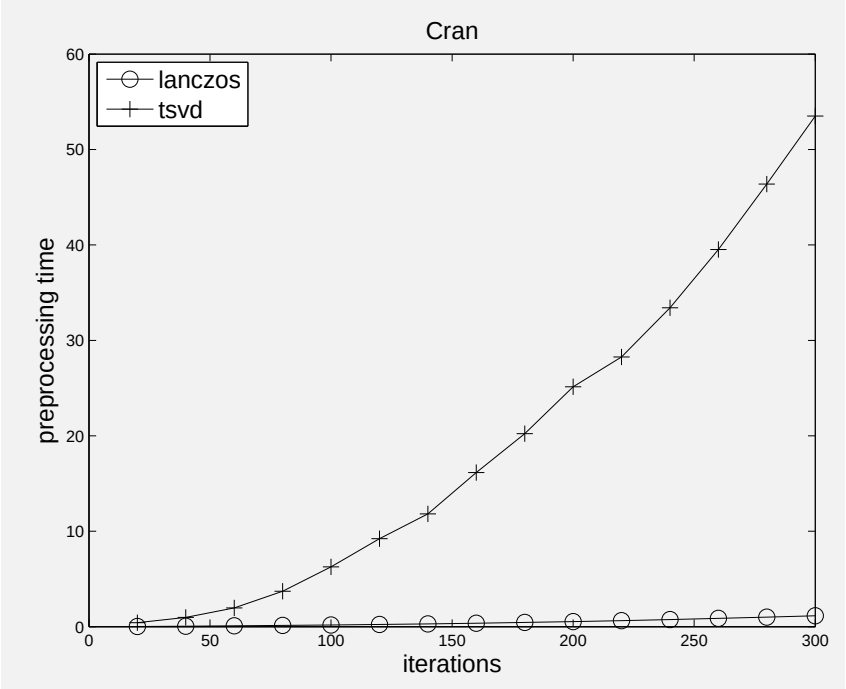
	# Terms	# Docs	# queries	sparsity
MED	7,014	1,033	30	0.735
CRAN	3,763	1,398	225	1.412

Preprocessing times

Med dataset.

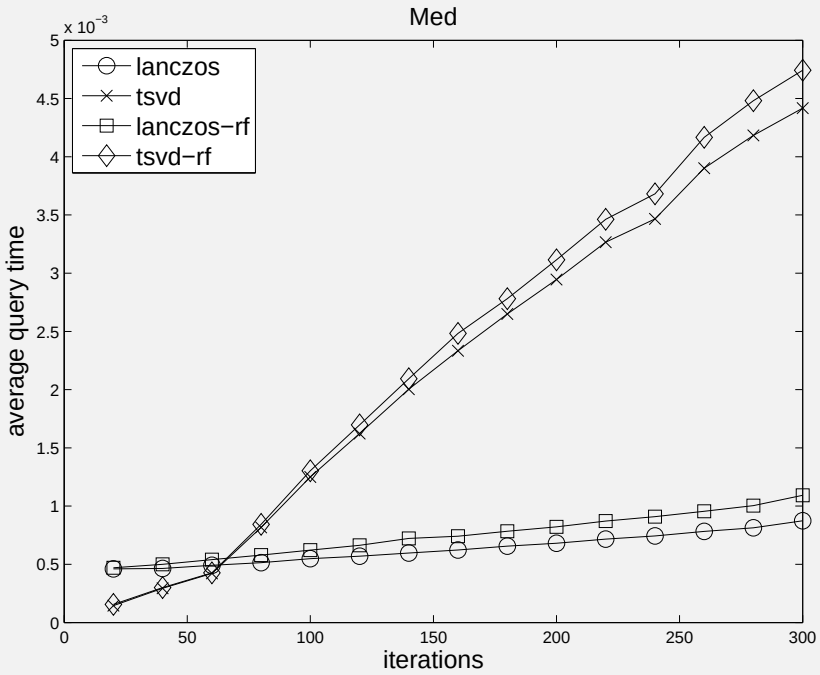


Cran dataset.

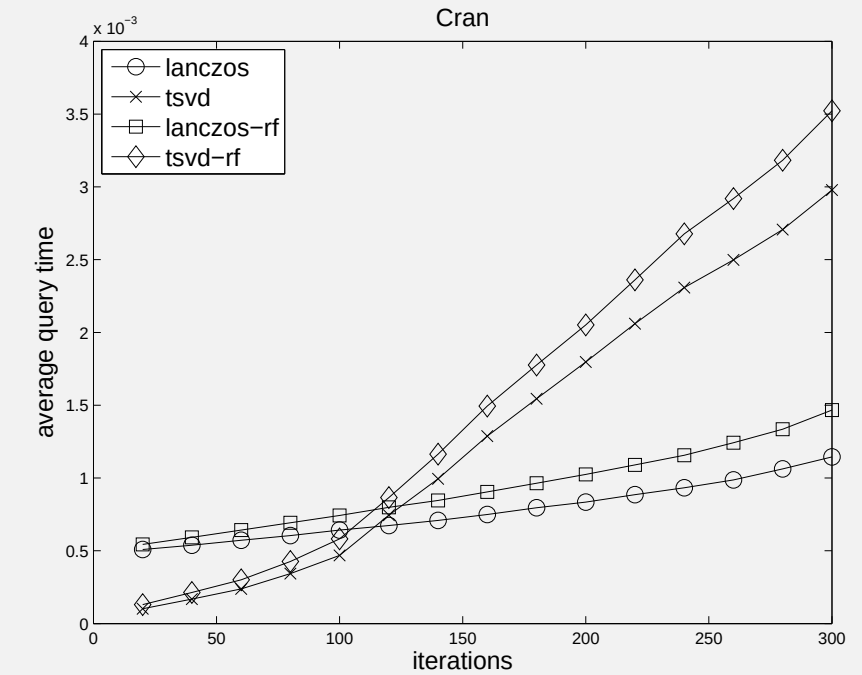


Average query times

Med dataset

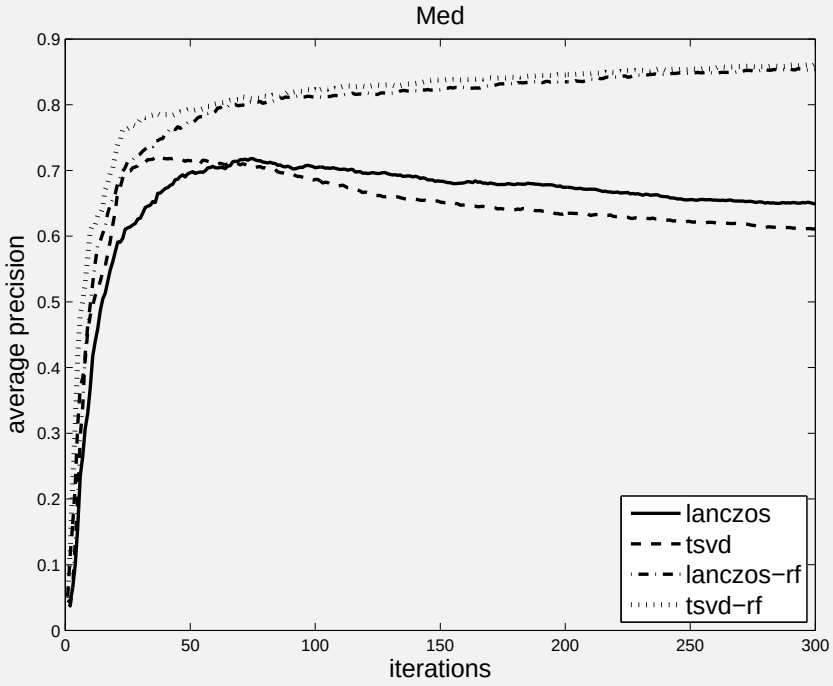


Cran dataset.

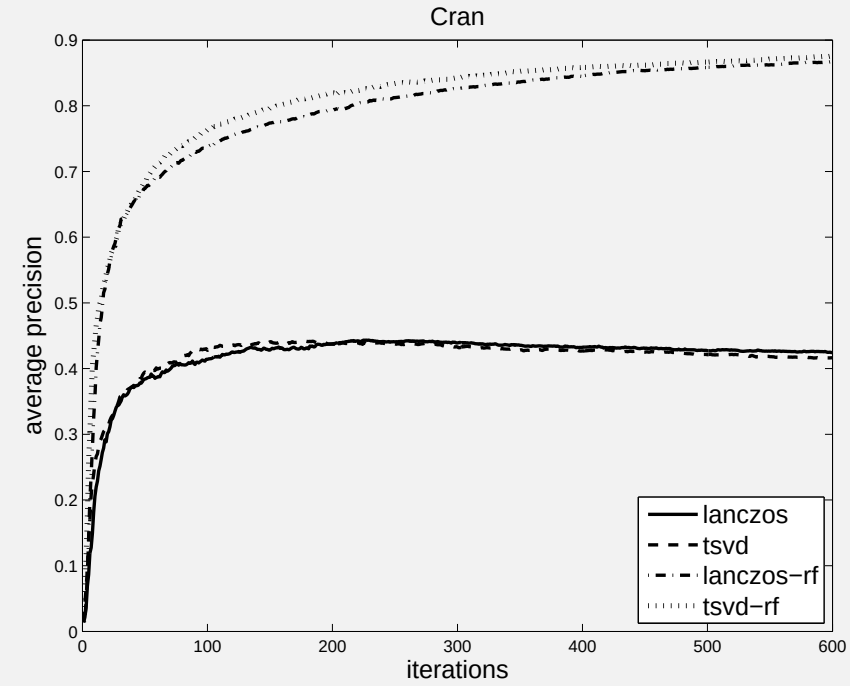


Average retrieval precision

Med dataset



Cran dataset

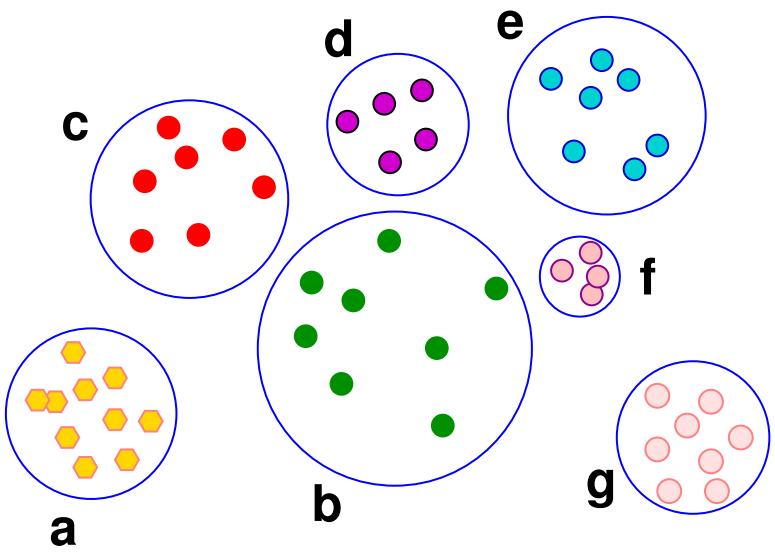


Retrieval precision comparisons

Supervised learning

We now have data that is ‘labeled’

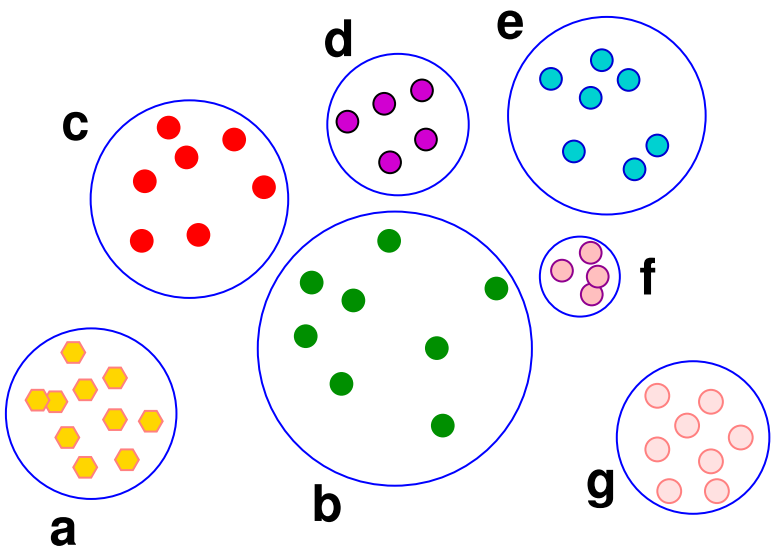
- Example: (health sciences) ‘malignant’- ’non malignant’
- Example: (materials) ’photovoltaic’, ’hard’, ’conductor’, ...
- Example: (Digit recognition) Digits ’0’, ’1’, ..., ’9’

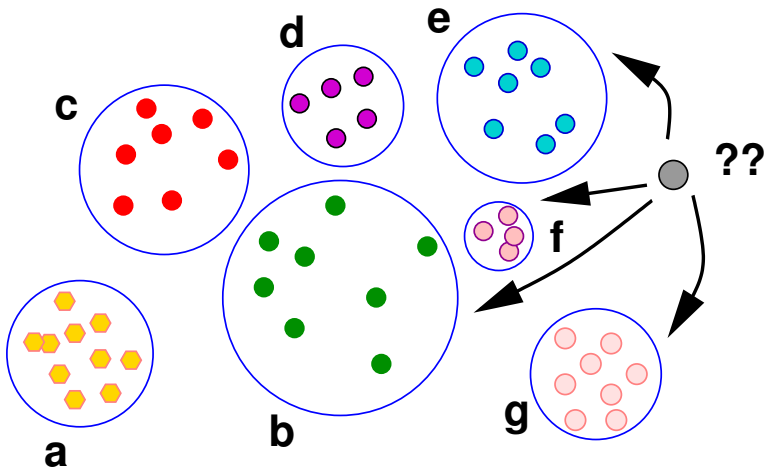


Supervised learning

We now have data that is ‘labeled’

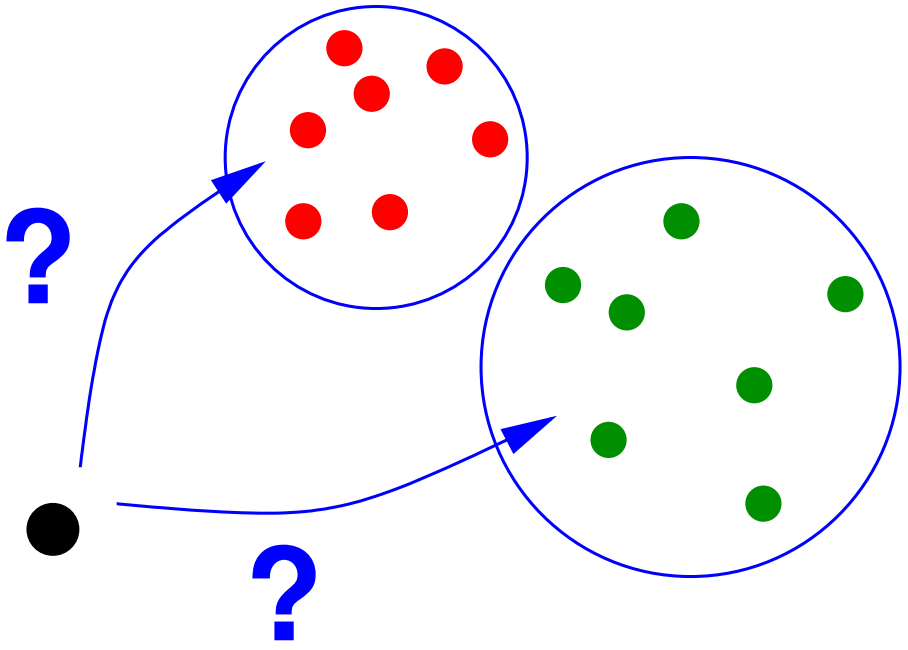
- Example: (health sciences) ‘malignant’- ’non malignant’
- Example: (materials) ’photovoltaic’, ’hard’, ’conductor’, ...
- Example: (Digit recognition) Digits ’0’, ’1’, ..., ’9’





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.

Another application:

IBM's Data Science Project to Build Analytics to Predict Heart Failure

by **Michael Goldberg** | October 15, 2013 1:45 pm | 0 Comments

00:00 00:00

Heart disease is the biggest and most costly health care challenge in the United States. Heart failure is the primary cause of more than 55,000 deaths per year, according to data cited by the Centers for Disease Control and Prevention. It costs the country an estimated $34.4 billion each year.

Now researchers are applying predictive analytics to design diagnostics and evaluate treatments. IBM announced Oct. 9 that the National Institutes of Health has awarded it and two health care organizations a $2 million grant to develop predictive analytics for primary care doctors to identify patients at risk of heart failure as much as two years ahead of time. IBM is working on this research project with Sutter Health, a network of doctors and hospitals in Northern California, and Geisinger Health Systems, a health care services organization in Pennsylvania.

On this episode of the Data Informed podcast, Shahram Ebadollahi, the program director of health informatics research at IBM, explains details about research project.

Join the Conversation

Search

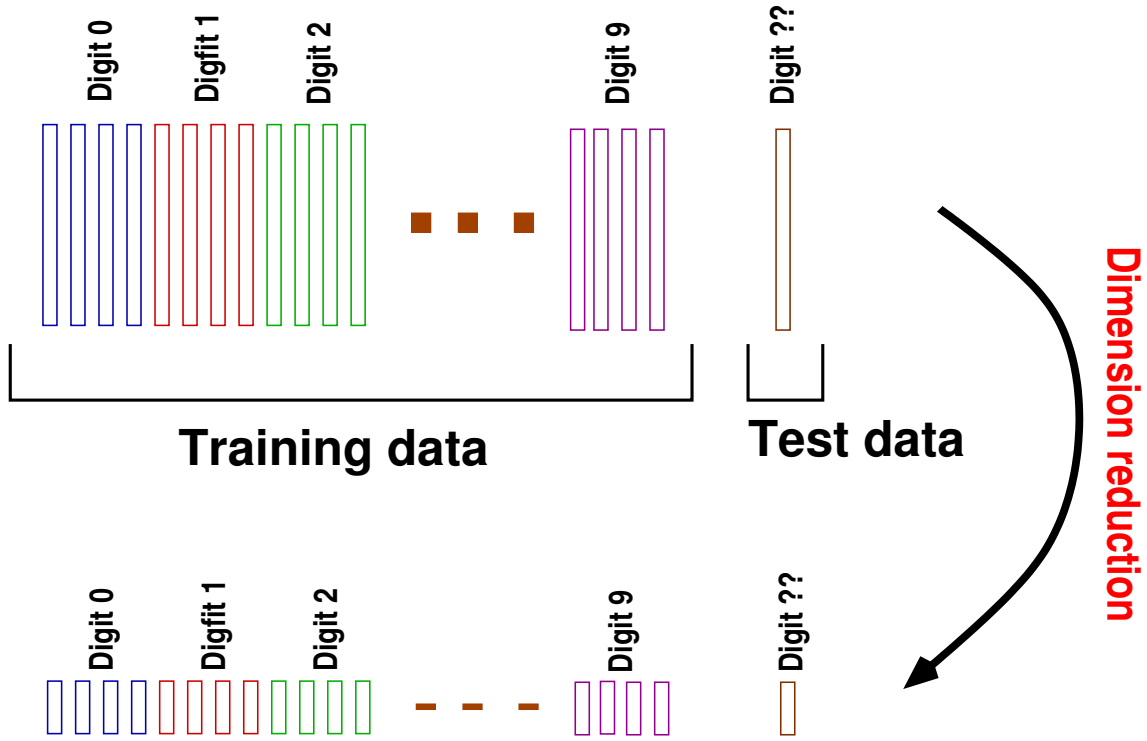
submit

Supervised learning: classification

- Best illustration: written digits recognition example

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

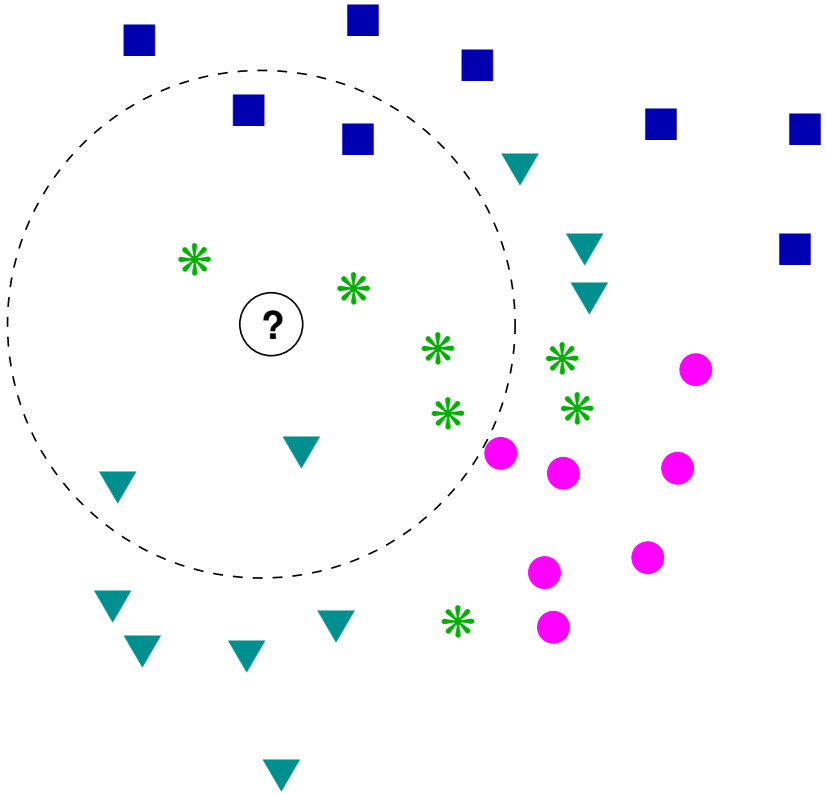
Problem: find label of test image



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

Basic method: K-nearest neighbors (KNN) classification

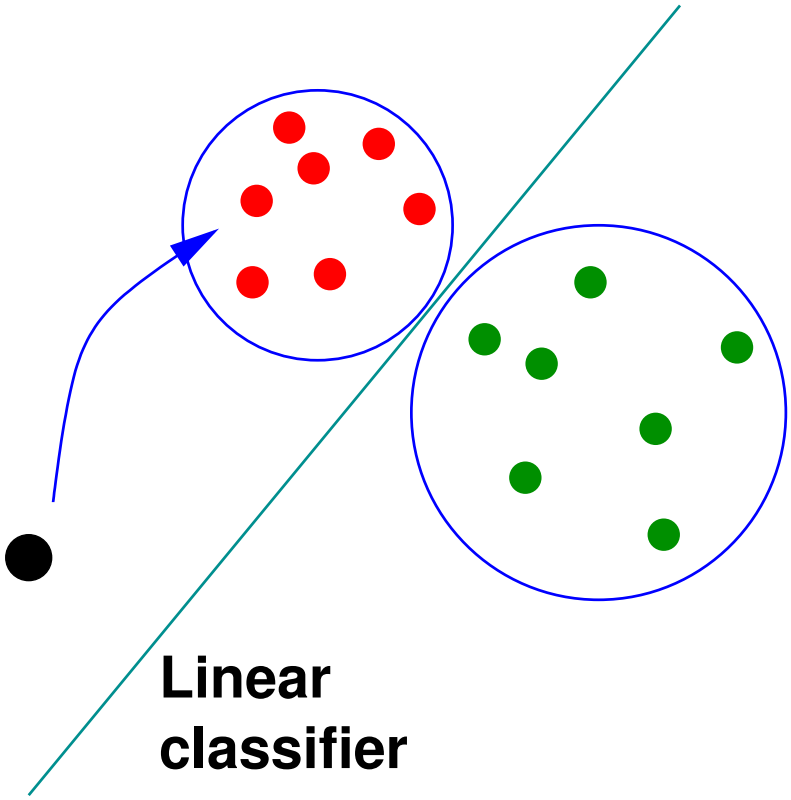
- Idea of a voting system: get distances between test sample and training samples
- Get the k nearest neighbors (here $k = 8$)
- Predominant class among these k items is assigned to the test sample ("$*$" here)



Supervised learning: Linear classification

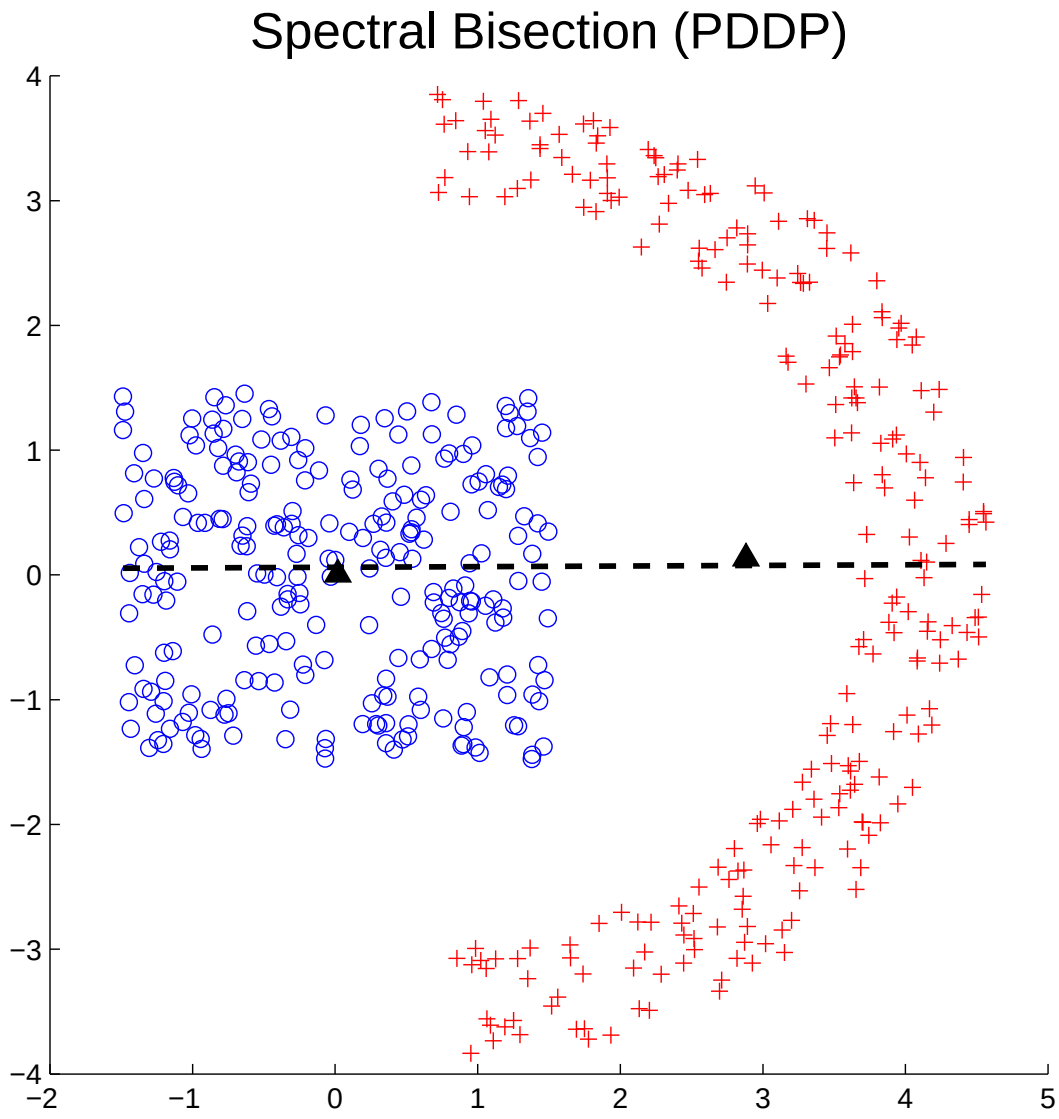
Linear classifiers: Find a hyperplane which best separates the data in classes A and B.

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

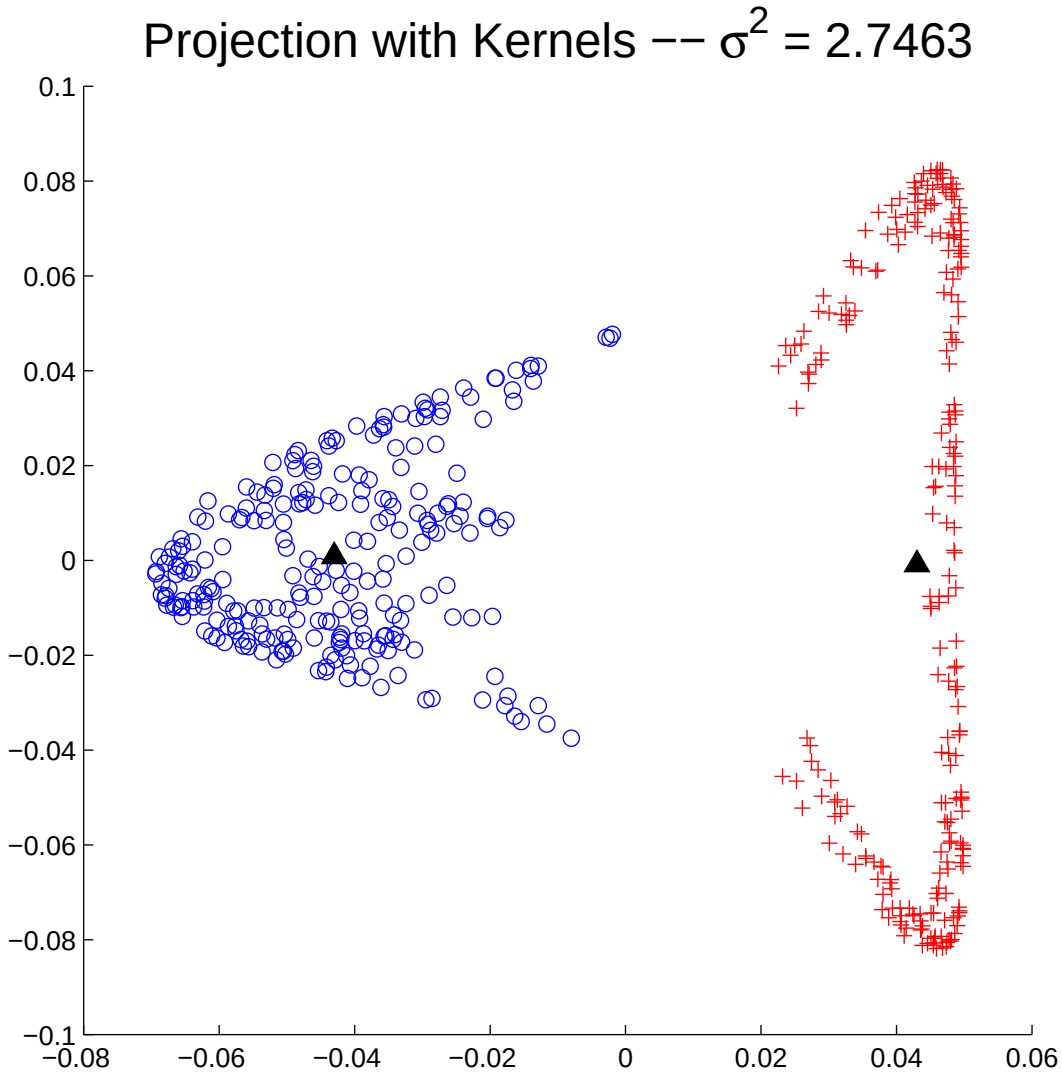


➤ Note: The world in non-linear. Often this is combined with Kernels – amounts to changing the inner product

A harder case:



- Use kernels to transform

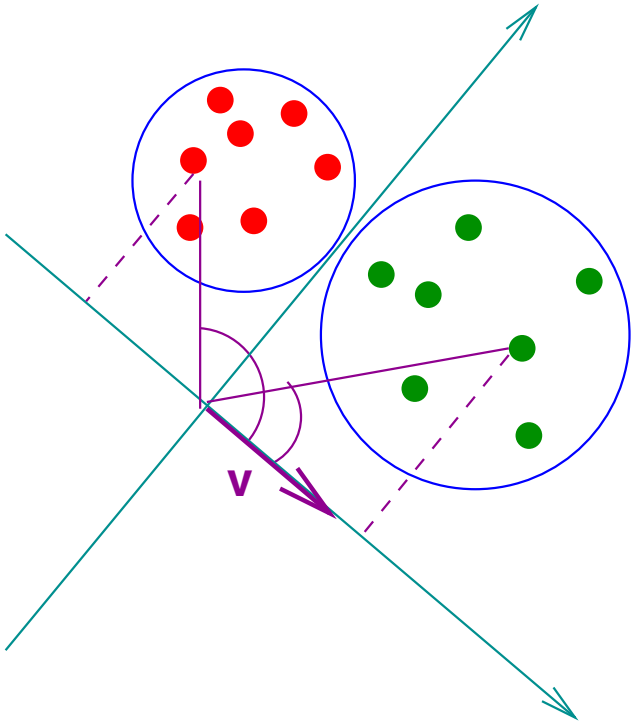


Transformed data with a Gaussian Kernel

Simple linear classifiers

- Let $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$ be the data matrix.
- and $\boldsymbol{L} = [\boldsymbol{l}_1, \cdots, \boldsymbol{l}_n]$ the labels either +1 or -1.

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



[For clarity: principal axis $\boldsymbol{u}$ drawn below where it should be]

Fisher's Linear Discriminant Analysis (LDA)

Principle: Use label information to build a good projector, i.e., one that can 'discriminate' well between 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

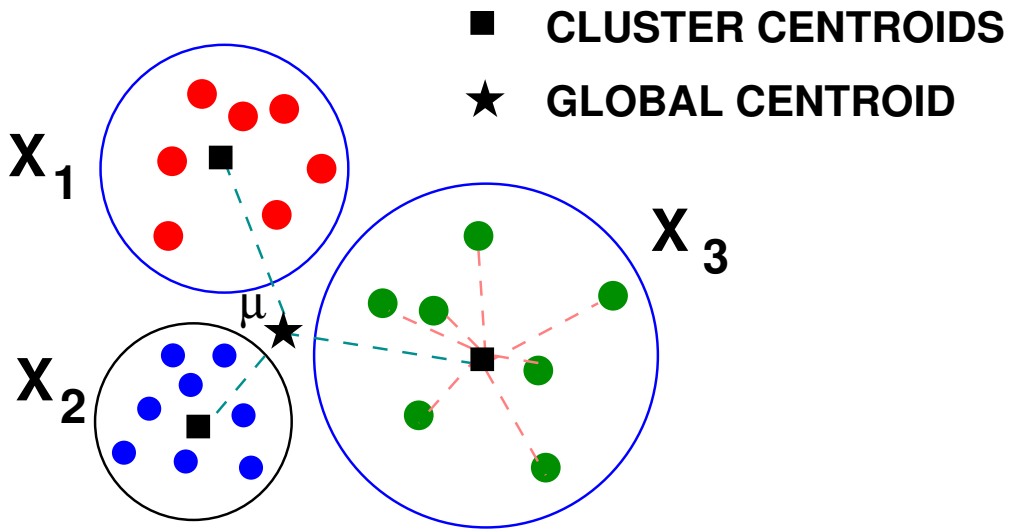
Define:

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

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

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:

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

➤ 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.

➤ Careful when defining $V(\rho)$: define the eigenvectors so the mapping $V(\rho)$ is differentiable.

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

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

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

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

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

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

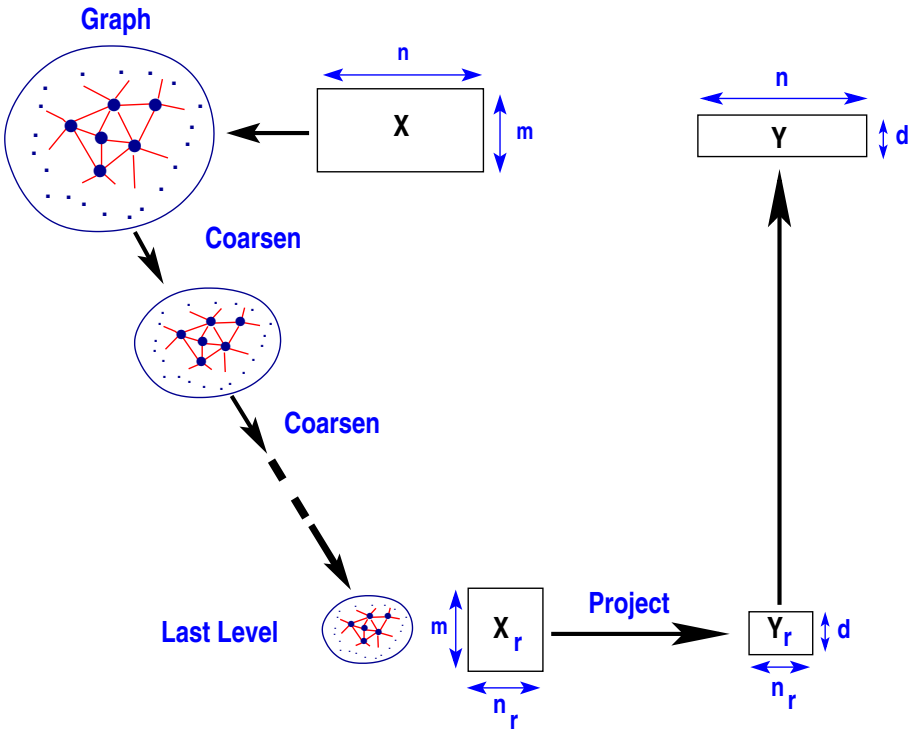
GRAPH-BASED TECHNIQUES

Multilevel techniques in brief

- Divide and conquer paradigms as well as multilevel methods in the sense of 'domain decomposition'
- Main principle: very costly to do an SVD [or Lanczos] on the whole set. Why not find a smaller set on which to do the analysis – without too much loss?
- Tools used: graph coarsening, divide and conquer –
- For text data we use hypergraphs

Multilevel Dimension Reduction

Main Idea: coarsen for a few levels. Use the resulting data set $\hat{X}$ to find a projector P from $\mathbb{R}^m$ to $\mathbb{R}^d$. P can be used to project original data or new data

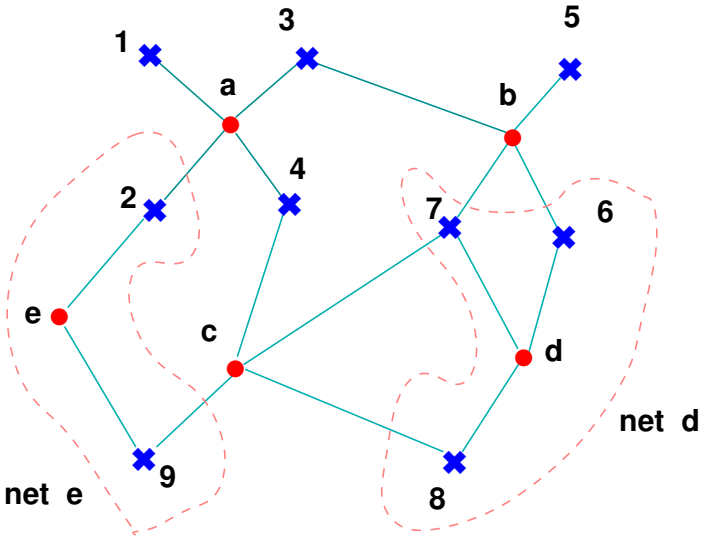


➤ Gain: Dimension reduction is done with a much smaller set. Hope: not much loss compared to using whole data

Making it work: Use of Hypergraphs for sparse data

A =

	1	2	3	4	5	6	7	8	9	
	*	*	*	*						a
			*		*	*	*			b
				*			*	*	*	c
						*	*	*		d
		*							*	e



Left: a (sparse) data set of n entries in $\mathbb{R}^m$ represented by a matrix $A \in \mathbb{R}^{m \times n}$

Right: corresponding hypergraph $H = (V, E)$ with vertex set V representing to the columns of A.

➤ Hypergraph Coarsening uses *column matching* – similar to a common one used in graph partitioning

➤ Compute the non-zero inner product $\langle a^{(i)}, a^{(j)} \rangle$ between two vertices i and j, i.e., the ith and jth columns of A.

➤ Note: $\langle a^{(i)}, a^{(j)} \rangle = \|a^{(i)}\| \|a^{(j)}\| \cos \theta_{ij}$.

Modif. 1: Parameter: $0 < \epsilon < 1$. Match two vertices, i.e., columns, only if angle between the vertices satisfies:

$$\tan \theta_{ij} \leq \epsilon$$

Modif. 2: Scale coarsened columns. If i and j matched and if $\|a^{(i)}\|_0 \geq \|a^{(j)}\|_0$ replace $a^{(i)}$ and $a^{(j)}$ by

$$c^{(\ell)} = \left(\sqrt{1 + \cos^2 \theta_{ij}} \right) a^{(i)}$$

➤ Call C the coarsened matrix obtained from A using the approach just described

Lemma: Let $C \in \mathbb{R}^{m \times c}$ be the coarsened matrix of A obtained by one level of coarsening of $A \in \mathbb{R}^{m \times n}$, with columns $a^{(i)}$ and $a^{(j)}$ matched if $\tan \theta_i \leq \epsilon$. Then

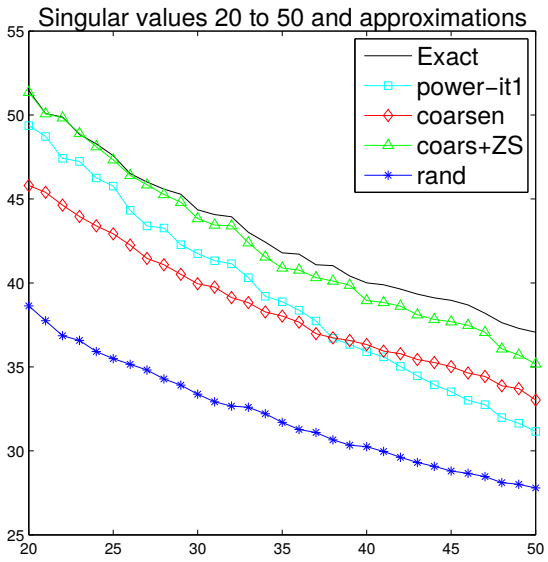
$$|x^T A A^T x - x^T C C^T x| \leq 3\epsilon \|A\|_F^2,$$

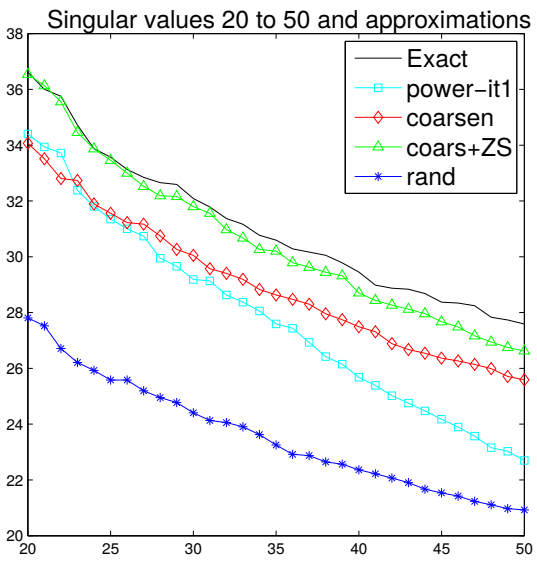
for any $x \in \mathbb{R}^m$ with $\|x\|_2 = 1$.

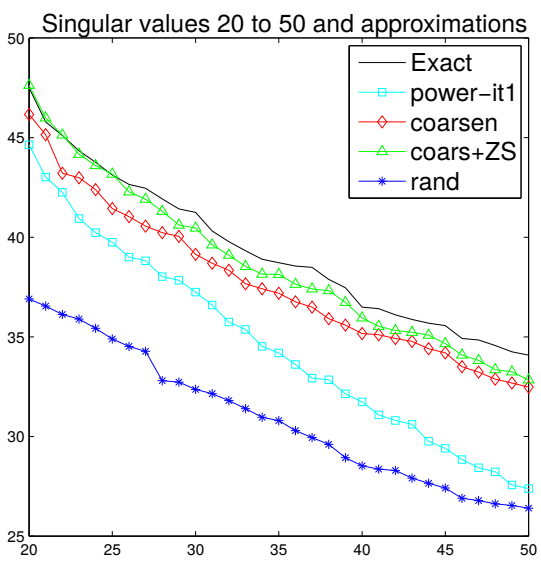
➤ Very simple bound for Rayleigh quotients for any x.

➤ Implies some bounds on singular values and norms - skipped.

Tests: Comparing singular values







Results for the datasets CRANFIELD (left), MEDLINE (middle), and TIME (right).

Low rank approximation: Coarsening, random sampling, and rand+coarsening. Err1 $= \|A - H_k H_k^T A\|_F$*; Err2*$= \frac{1}{k}\sum_k \frac{|\hat{\sigma}_i - \sigma_i|}{\sigma_i}$

Dataset	n	k	c	Coarsen		Rand Sampl	
				Err1	Err2	Err1	Err2
Kohonen	4470	50	1256	86.26	0.366	93.07	0.434
aft01	8205	50	1040	913.3	0.299	1006.2	0.614
FA	10617	30	1504	27.79	0.131	28.63	0.410
chipcool0	20082	30	2533	6.091	0.313	6.199	0.360
brainpc2	27607	30	865	2357.5	0.579	2825.0	0.603
scfxm1-2b	33047	25	2567	2326.1	–	2328.8	–
thermomechTC	102158	30	6286	2063.2	–	2079.7	–
Webbase-1M	1000005	25	15625	–	–	3564.5	–

LINEAR ALGEBRA METHODS: EXAMPLES

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

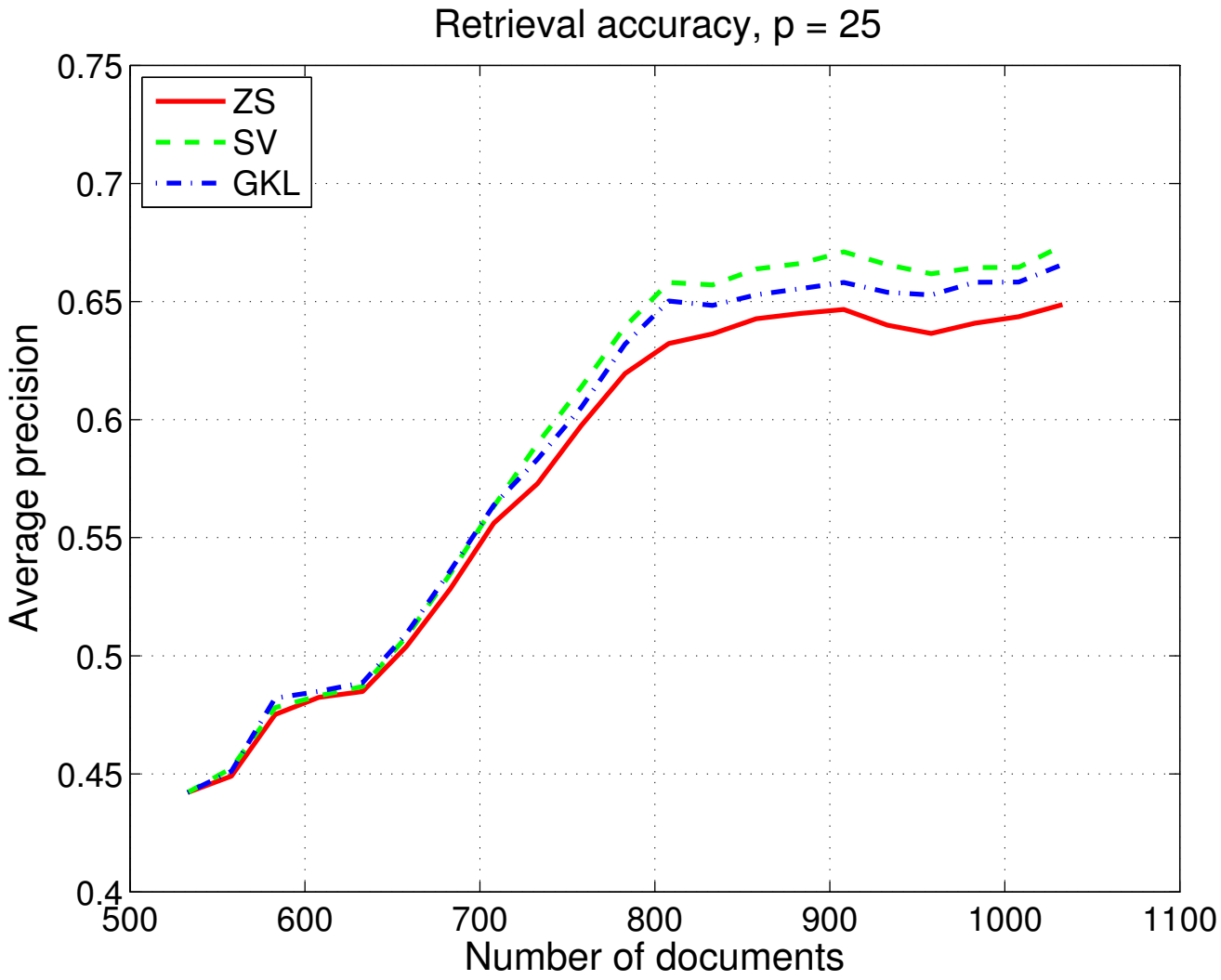
- Zha–Simon ('99): Compute the SVD of H_D & get approximate SVD from above equation
- It turns out this is a Rayleigh-Ritz projection method for the SVD [E. Vecharynski & YS 2013]
- Can show optimality properties as a result

Updating the SVD

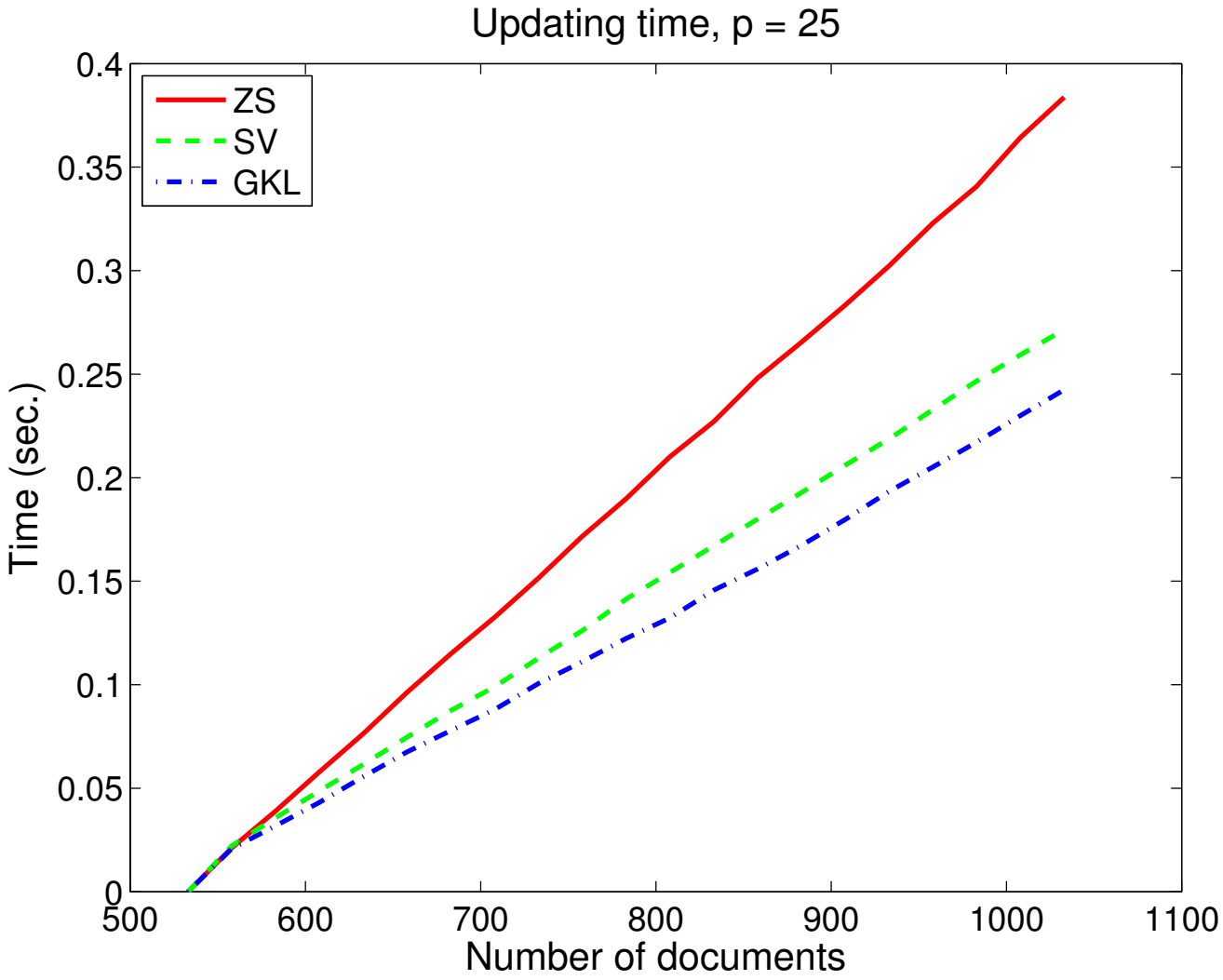
- When the number of updates is large this becomes costly.
- Idea: Replace $\hat{U}_p$ by a low dimensional approximation:
- Use $\bar{U}$ of the form $\bar{U} = [U_k, Z_l]$ instead of $\bar{U} = [U_k, \hat{U}_p]$
- Z_l must capture the range of $D_k = (I - U_k U_k^T) D$
- Simplest idea : best rank–l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

Conclusion

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

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

Alexander Graham Bell (1847-1922)

- In the words of Lao Tzu:

If you do not change directions, you may end-up where you are heading

Thank you !

- Visit my web-site at `www.cs.umn.edu/~saad`