A few applications of the SVD

Many methods require to approximate the original data (matrix) by a low rank matrix before attempting to solve the original problem

- Regularization methods require the solution of a least-squares linear system $Ax = b$ approximately in the dominant singular space of A

- The Latent Semantic Indexing (LSI) method in information retrieval, performs the "query" in the dominant singular space of A

- Methods utilizing Principal Component Analysis, e.g. Face Recognition.

Commonality: Approximate A (or $A^{\dagger}$) by a lower rank approximation A_k (using dominant singular space) before solving original problem.

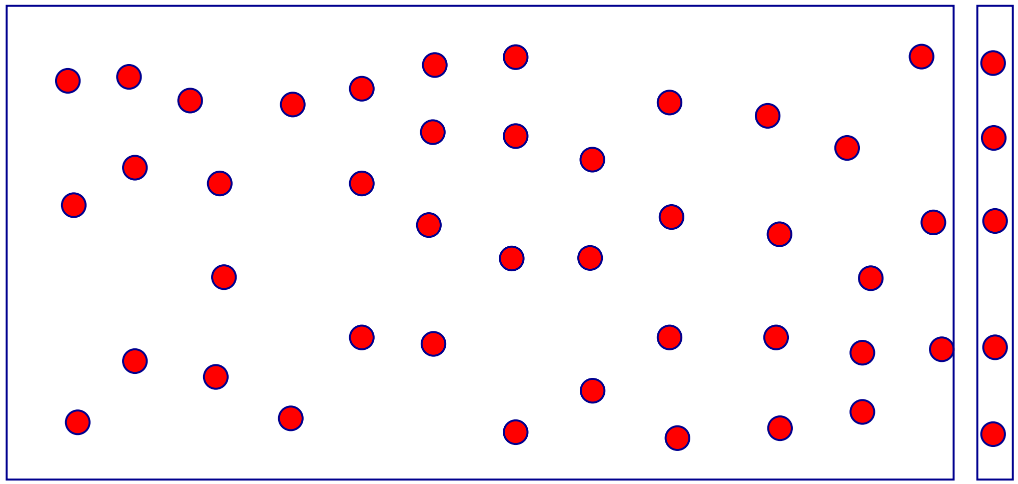
➤ This approximation captures the main features of the data while getting rid of noise and redundancy

Note: Common misconception: 'we need to reduce dimension in order to reduce computational cost'. In reality: using less information often yields better results. This is the problem of overfitting.

➤ Good illustration: Information Retrieval (IR)

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 $\boldsymbol{A}$ that best matches $\boldsymbol{q}$
- Similarity metric: angle between the column and $\boldsymbol{q}$ - Use cosines:

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

- To rank all documents we need to compute

$$\boxed{\boldsymbol{s} = \boldsymbol{A}^T\boldsymbol{q}}$$

- $\boldsymbol{s}$ = similarity vector.
- Literal matching – not very effective.

Use of the SVD

- Many problems with literal matching: *polysemy*, *synonymy*, ...
- Need to extract intrinsic information – or underlying “semantic” information –
- Solution (LSI): replace matrix $\boldsymbol{A}$ by a low rank approximation using the Singular Value Decomposition (SVD)

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

- $\boldsymbol{U_k}$: term space, $\boldsymbol{V_k}$: document space.
- Refer to this as Truncated SVD (TSVD) approach

New similarity vector:

$$s_k = A_k^T q = V_k \Sigma_k U_k^T q$$

Issues:

- Problem 1: How to select k?
- Problem 2: computational cost (memory + computation)
- Problem 3: updates [e.g. google data changes all the time]
- Not practical for very large sets

LSI : an example

```
%% D1   :  INFANT & TODDLER first aid
%% D2   :  BABIES & CHILDREN's room for your HOME
%% D3   :  CHILD SAFETY at HOME
%% D4   :  Your BABY's HEALTH and SAFETY
%%      :  From INFANT to TODDLER
%% D5   :  BABY PROOFING basics
%% D6   :  Your GUIDE to easy rust PROOFING
%% D7   :  Beanie BABIES collector's GUIDE
%% D8   :  SAFETY GUIDE for CHILD PROOFING your HOME
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% TERMS: 1:BABY  2:CHILD 3:GUIDE 4:HEALTH 5:HOME
%%        6:INFANT 7:PROOFING  8:SAFETY 9:TODDLER
%% Source: Berry and Browne, SIAM., '99
```

- Number of documents: 8
- Number of terms: 9

➤ Raw matrix (before scaling).

$$A = \begin{array}{|cccccccc|l} d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & \\ \hline & 1 & & 1 & 1 & & 1 & & bab \\ & 1 & 1 & & & & & 1 & chi \\ & & & & & 1 & 1 & 1 & gui \\ & & & 1 & & & & & hea \\ & 1 & 1 & & & & & 1 & hom \\ 1 & & & 1 & & & & & inf \\ & & & & 1 & 1 & & 1 & pro \\ & & 1 & 1 & & & & 1 & saf \\ 1 & & & 1 & & & & & tod \\ \hline \end{array}$$

Get the anwser to the query `Child Safety`, so

$$q = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0]$$

using cosines and then using LSI with $k = 3$.

Dimension reduction

Dimensionality Reduction (DR) techniques pervasive to many applications

➤ Often main goal of dimension reduction is not to reduce computational cost. Instead:

- Dimension reduction used to reduce noise and redundancy in data
- Dimension reduction used to discover patterns (e.g., supervised learning)

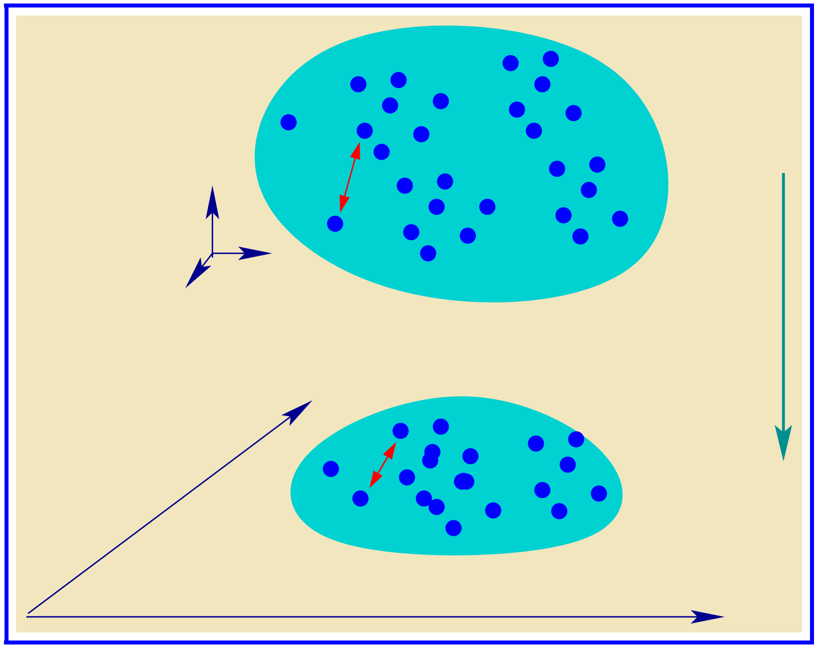
➤ Techniques depend on desirable features or application: Preserve angles? Preserve distances? Maximize variance? ..

The problem

➤ Given $d \ll m$ find a mapping $\Phi : x \in \mathbb{R}^m \longrightarrow y \in \mathbb{R}^d$

➤ Mapping may be explicit (e.g., $y = V^T x$)

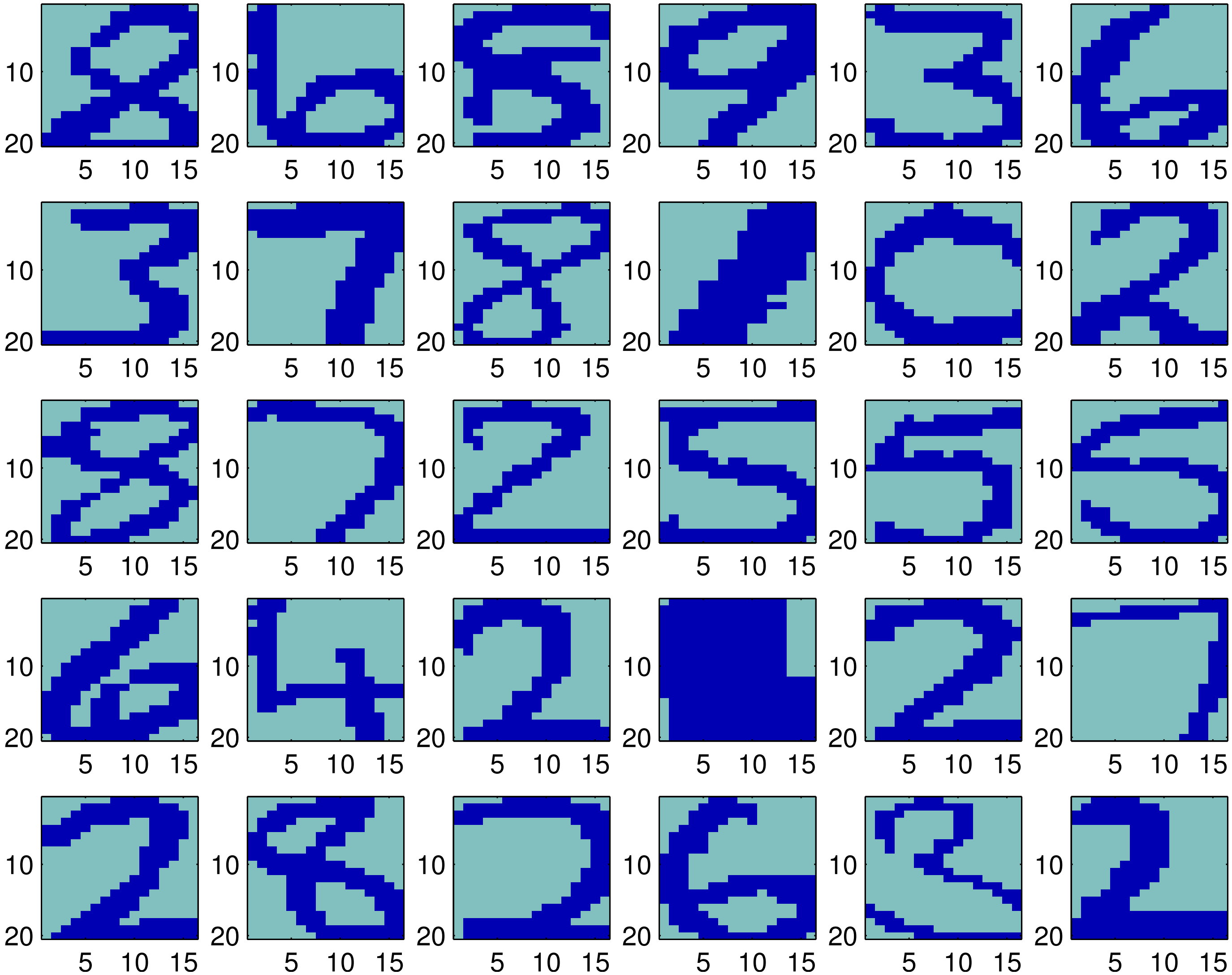
➤ Or implicit (nonlinear)



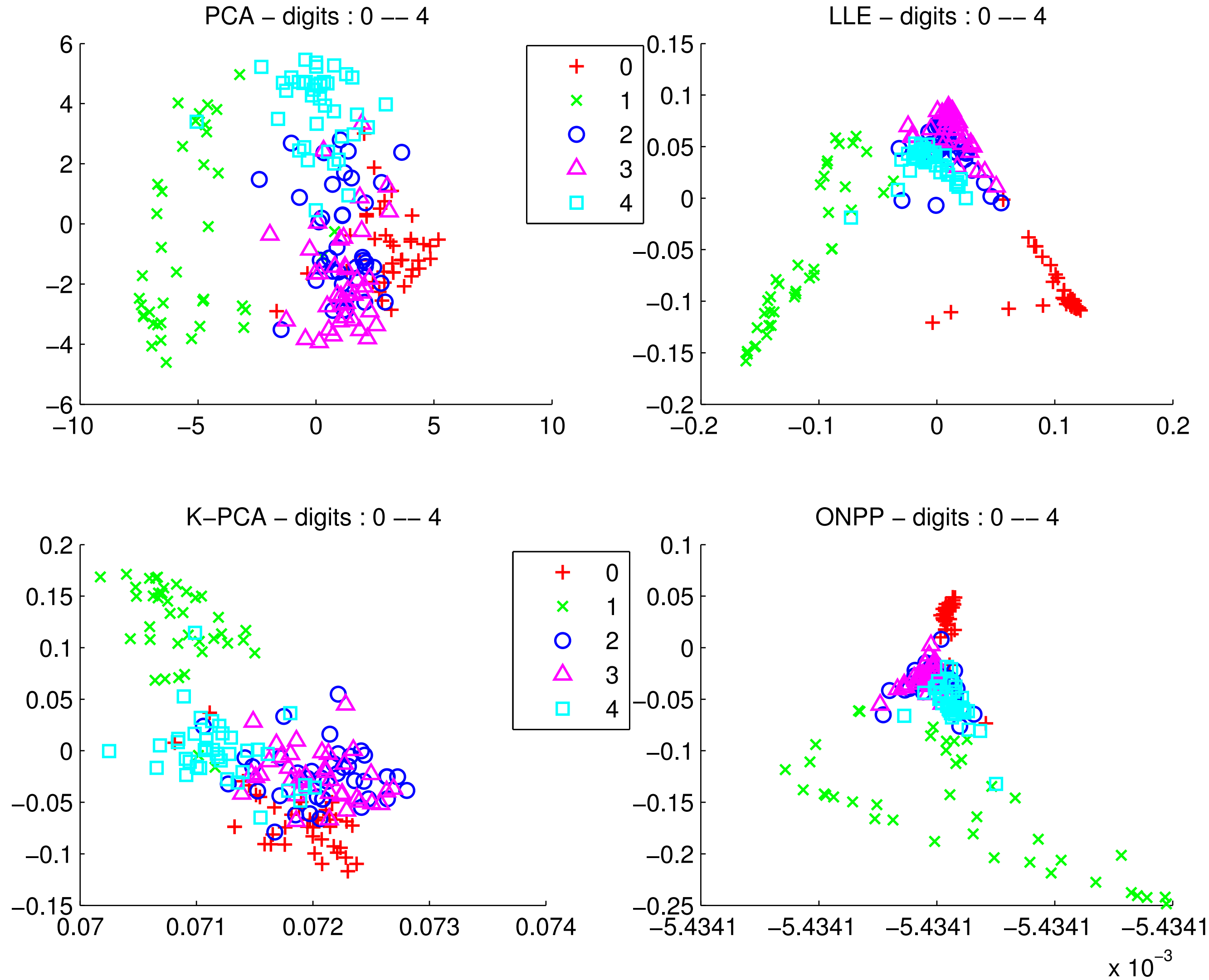
Practically: Find a low-dimensional representation $Y \in \mathbb{R}^{d \times n}$ of $X \in \mathbb{R}^{m \times n}$.

➤ Two classes of methods: (1) projection techniques and (2) nonlinear implicit methods.

Example: Digit images (a sample of 30)



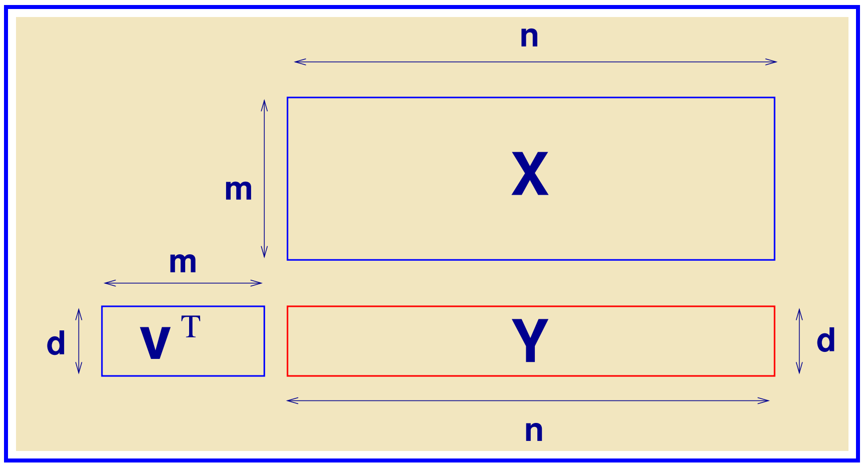
A few 2-D 'reductions':



Projection-based Dimensionality Reduction

Given: a data set $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$, and $\boldsymbol{d}$ the dimension of the desired reduced space $\boldsymbol{Y}$.

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



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

- $\boldsymbol{m}$-dimens. objects ($\boldsymbol{x}_i$) 'flattened' to $\boldsymbol{d}$-dimens. space ($\boldsymbol{y}_i$)

Problem: Find the best such mapping (optimization) given that the $\boldsymbol{y}_i$'s must satisfy certain constraints

Principal Component Analysis (PCA)

- PCA: find $\boldsymbol{V}$ (orthogonal) so that projected data $\boldsymbol{Y} = \boldsymbol{V}^T\boldsymbol{X}$ has maximum variance

- Maximize over all orthogonal $\boldsymbol{m} \times \boldsymbol{d}$ matrices $\boldsymbol{V}$:

$$\sum_i \|\boldsymbol{y}_i - \frac{1}{n}\sum_j \boldsymbol{y}_j\|_2^2 = \cdots = \mathrm{Tr}\left[\boldsymbol{V}^\top \bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^\top \boldsymbol{V}\right]$$

Where: $\bar{\boldsymbol{X}} = [\bar{\boldsymbol{x}}_1, \cdots, \bar{\boldsymbol{x}}_n]$ with $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{\mu}$, $\boldsymbol{\mu}$ = mean.

Solution:

$\boldsymbol{V} = \{$ dominant eigenvectors $\}$ of the covariance matrix

- i.e., Optimal $\boldsymbol{V}$ = Set of left singular vectors of $\bar{\boldsymbol{X}}$ associated with $\boldsymbol{d}$ largest singular values.

Show that $\bar{X} = X(I - \frac{1}{n}ee^T)$ (here e = vector of all ones). What does the projector $(I - \frac{1}{n}ee^T)$ do?

Show that solution V also minimizes 'reconstruction error' ..

$$\sum_i \|\bar{x}_i - VV^T\bar{x}_i\|^2 = \sum_i \|\bar{x}_i - V\bar{y}_i\|^2$$

.. and that it also maximizes $\sum_{i,j} \|y_i - y_j\|^2$

Matrix Completion Problem

Consider a table of movie ratings. You want to predict missing ratings by assuming commonality (low rank matrix).

given data				predictions		
movie	Paul	Jane	Ann	Paul	Jane	Ann
Title-1	–1	3	–1	–1.2	1.7	–0.7
Title-2	4	x	3	2.8	–1.2	2.5
Title-3	–3	1	–4	–2.7	1.0	–2.5
Title-4	x	–1	–1	–0.5	–0.3	–0.6
Title-5	3	–2	1	1.8	–1.4	1.4
Title-6	–2	3	x	–1.6	1.8	–1.2
	$\boldsymbol{A}$			$\boldsymbol{X}$		

➤ Minimize $\|(\boldsymbol{X} - \boldsymbol{A})_{\text{mask}}\|_{\boldsymbol{F}}^{2} + 4\|\boldsymbol{X}\|_{*}$

"minimize sum-of-squares of deviations from known ratings plus sum of singular values of solution (to reduce the rank)."