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 \( A^\top \)) 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 \( a_{ij} = L_{ij} G_i N_j \).
- 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:

\[
\frac{|c^\top q|}{\|c\|_2 \|q\|_2}
\]

- To rank all documents we need to compute

\[
s = A^\top q
\]

- \( s \) = 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 $A$ by a low rank approximation using the Singular Value Decomposition (SVD)
  
  $A = U\Sigma V^T \rightarrow A_k = U_k \Sigma_k V_k^T$

- $U_k$: term space, $V_k$: 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 & TODLER first aid |
| D2 | BABIES & CHILDREN’s room for your HOME |
| D3 | CHILD SAFETY at HOME |
| D4 | Your BABY’s HEALTH and SAFETY |
| D5 | From INFANT to TODDLER |
| D6 | BABY PROOFING basics |
| D7 | Your GUIDE to easy rust PROOFING |
| D8 | Beanie BABIES collector’s GUIDE |
| D9 | SAFETY GUIDE for CHILD PROOFING your HOME |


Source: Berry and Browne, SIAM., ’99

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

Raw matrix (before scaling).

\[
A = \begin{bmatrix}
1 & 1 & 1 & 1 & bab \\
1 & 1 & 1 & 1 & chi \\
1 & 1 & 1 & 1 & gui \\
1 & 1 & 1 & 1 & hea \\
1 & 1 & 1 & 1 & hom \\
1 & 1 & 1 & 1 & inf \\
1 & 1 & 1 & 1 & pro \\
1 & 1 & 1 & 1 & saf \\
1 & 1 & 1 & 1 & tod \\
\end{bmatrix}
\]

Get the answer to the query Child Safety, so

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

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

\[
\begin{align*}
\text{Using cosines:} & \quad s_1 = A_1^T q \\
\text{Using LSI:} & \quad s_3 = A_3^T q
\end{align*}
\]
**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 \( \Phi : x \in \mathbb{R}^m \rightarrow y \in \mathbb{R}^d \)
- 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)**

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```

```
Example: Digit images (a sample of 30)
```

```
A few 2-D 'reductions':
```

```
The problem
```

```
Dimension reduction
```
**Projection-based Dimensionality Reduction**

**Given:** a data set \( X = [x_1, x_2, \ldots, x_n] \), and \( d \) the dimension of the desired reduced space \( Y \).

**Want:** a linear transformation from \( X \) to \( Y \)

\[
X \in \mathbb{R}^{m \times n} \\
V \in \mathbb{R}^{m \times d} \\
Y = V^T X \\
\rightarrow Y \in \mathbb{R}^{d \times n}
\]

- \( m \)-dimens. objects \((x_i)\) ‘flattened’ to \( d \)-dimens. space \((y_i)\)

**Problem:** Find the best such mapping (optimization) given that the \( y_i \)'s must satisfy certain constraints

**Principal Component Analysis (PCA)**

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

\[
\max_{V} \sum_i \|y_i - \frac{1}{n} \sum_j y_j\|_2^2 = \cdots = \text{Tr} [V^T \bar{X} \bar{X}^T V]
\]

Where: \( \bar{X} = [\bar{x}_1, \ldots, \bar{x}_n] \) with \( \bar{x}_i = x_i - \mu, \mu = \text{mean} \).

**Solution:**

\[
V = \{ \text{dominant eigenvectors} \} \text{ of the covariance matrix}
\]

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

**Matrix Completion Problem**

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

\[
\min \| (X - A)_{\text{mask}}\|_F^2 + 4\|X\|_*
\]

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