COMP30120

# Dimension Reduction

Derek Greene

School of Computer Science and Informatics
Autumn 2015

# Overview

- Feature Transformation v Selection
- Feature Transformation Methods
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Latent Semantic Indexing (LSI)

# Feature Transformation v Selection

## Feature Selection

- Tries to find a minimum subset of the original features that optimises one or more criteria, rather than producing an entirely new set of dimensions for the data.

## Feature Transformation (Feature Extraction)

- A popular alternative strategy in data pre-processing is to transform the data into an entirely different format.
- Examples represented by one set of features are transformed to another new set of features.
- Resulting features can be more compact and less noisy, resulting in more accurate predictions.
- Typically involve a linear transformation of the original data.

# Linear Transformations

- In a linear transformation, the new variables are linear functions of the old variables.
- **Simple example:** Bill owes Mary £5 and CHF15 after a holiday.
  - Current exchange rates:

    €:£ → 1.25:1

    €:CHF → 0.8:1
  - Based on rates, Bill owes €(1.25 x 5 + 0.8 x 15) = €18.25
  - We can view this as a linear transformation…

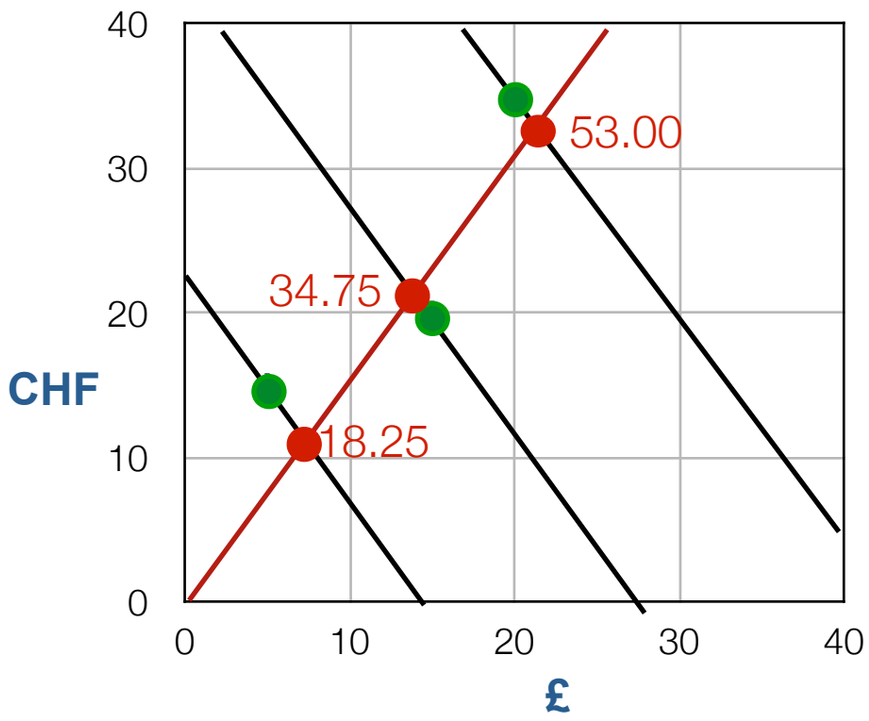
| £ | CHF |
|---|---|
| 5 | 15 |
| 15 | 20 |
| 20 | 35 |

x

| |
|---|
| 1.25 |
| 0.8 |

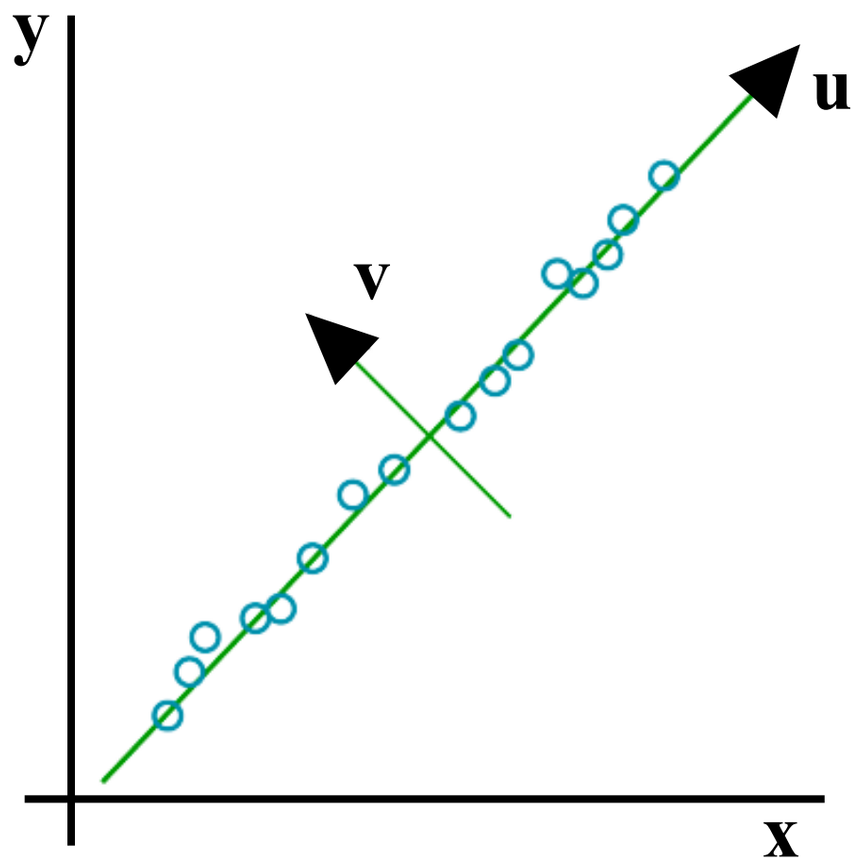
=

| € |
|---|
| 18.25 |
| 34.75 |
| 53.00 |



# Principle Component Analysis (PCA)

- Projection methods find a mapping from the original *d*-dimensional space to a new *(k < d)*-dimensional space, with minimum loss of information.
- "Good" spaces for projections are characterised by preserving most of the useful information in the data.
- PCA: An unsupervised method which performs dimensionality reduction while keeping as much of the variance in the original space as possible. Allows us to find the direction in which the data varies.




Given data in terms of dimensions *(x,y)*, the principal direction in which the data varies is along the *u* axis.

Very little variation in the data in the direction of *v*.

➡ We could represent the data using the *u* dimension, and discard *v*.

# Eigenvectors and Eigenvalues

- Given an input matrix $\mathbf{X}$, an eigenvector of the matrix is a non-zero vector $\mathbf{v}$ that satisfies the equation:

$$\mathbf{X}v = \lambda v$$

  where the corresponding number $\lambda$ is called an eigenvalue.

- Eigendecomposition is the factorization of a matrix into its eigenvalues and eigenvectors.

$$\mathbf{X} = \begin{pmatrix} 1.0000 & 0.5000 & 0.3330 & 0.2500 \\ 0.5000 & 1.0000 & 0.6667 & 0.5000 \\ 0.3333 & 0.6667 & 1.0000 & 0.7500 \\ 0.2500 & 0.5000 & 0.7500 & 1.0000 \end{pmatrix}$$

4 x 4 matrix

$$\mathbf{\Lambda} = \begin{pmatrix} 2.5361 & 0 & 0 & 0 \\ 0 & 0.8483 & 0 & 0 \\ 0 & 0 & 0.4078 & 0 \\ 0 & 0 & 0 & 0.2077 \end{pmatrix}$$

4 eigenvalues

$$\mathbf{V} = \left(\begin{array}{c:c:c:c} -0.37775 & -0.81052 & -0.44217 & -0.06988 \\ -0.53223 & -0.18762 & 0.74199 & 0.36221 \\ -0.56139 & 0.30099 & 0.04872 & -0.76926 \\ -0.50881 & 0.46611 & -0.50155 & 0.52169 \end{array}\right)$$

4 eigenvectors

# Principle Component Analysis (PCA)

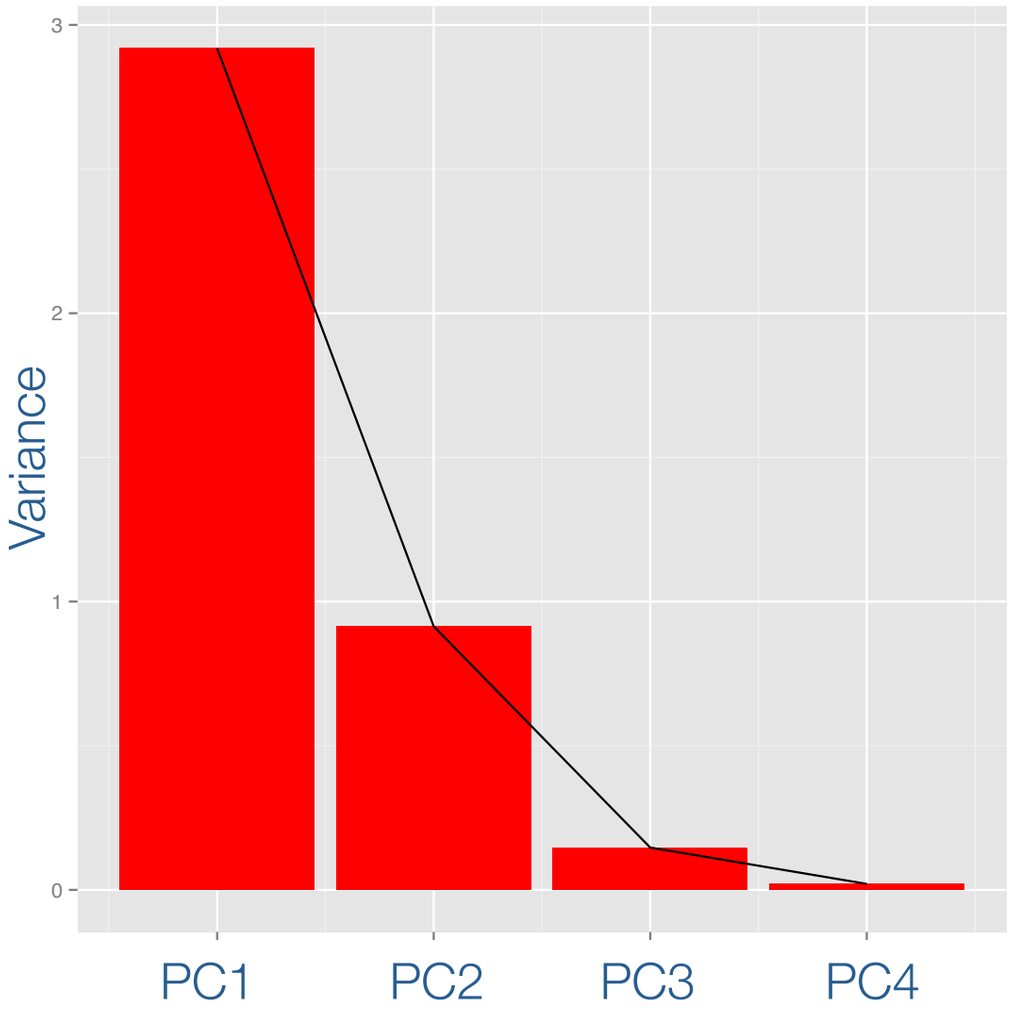
- Principal Components (PCs): New features constructed as linear combinations of the original features, which are uncorrelated with one another.
- First PC accounts for the most variability in the data. Next PC has the highest variance possible under the constraint that it is orthogonal to (uncorrelated with) the first PC, and so on…
- **Applying PCA:**

  1. For a data matrix $\mathbf{X}$ with $n$ samples, calculate mean of the columns of $\mathbf{X}$.
  2. Subtract the mean from each row of $\mathbf{X}$, to give the "centered" matrix $\mathbf{Y}$.
  3. Calculate the covariance matrix $\mathbf{C} = \mathbf{Y}^{\mathsf{T}}\mathbf{Y}/(n-1)$
  4. Calculate the eigenvectors of the covariance matrix $\mathbf{C}$.
  5. The PCs are given by the eigenvectors of $\mathbf{C}$. The *i*-th PC is given by the eigenvector corresponding to the *i*-th largest eigenvalue of $\mathbf{C}$.

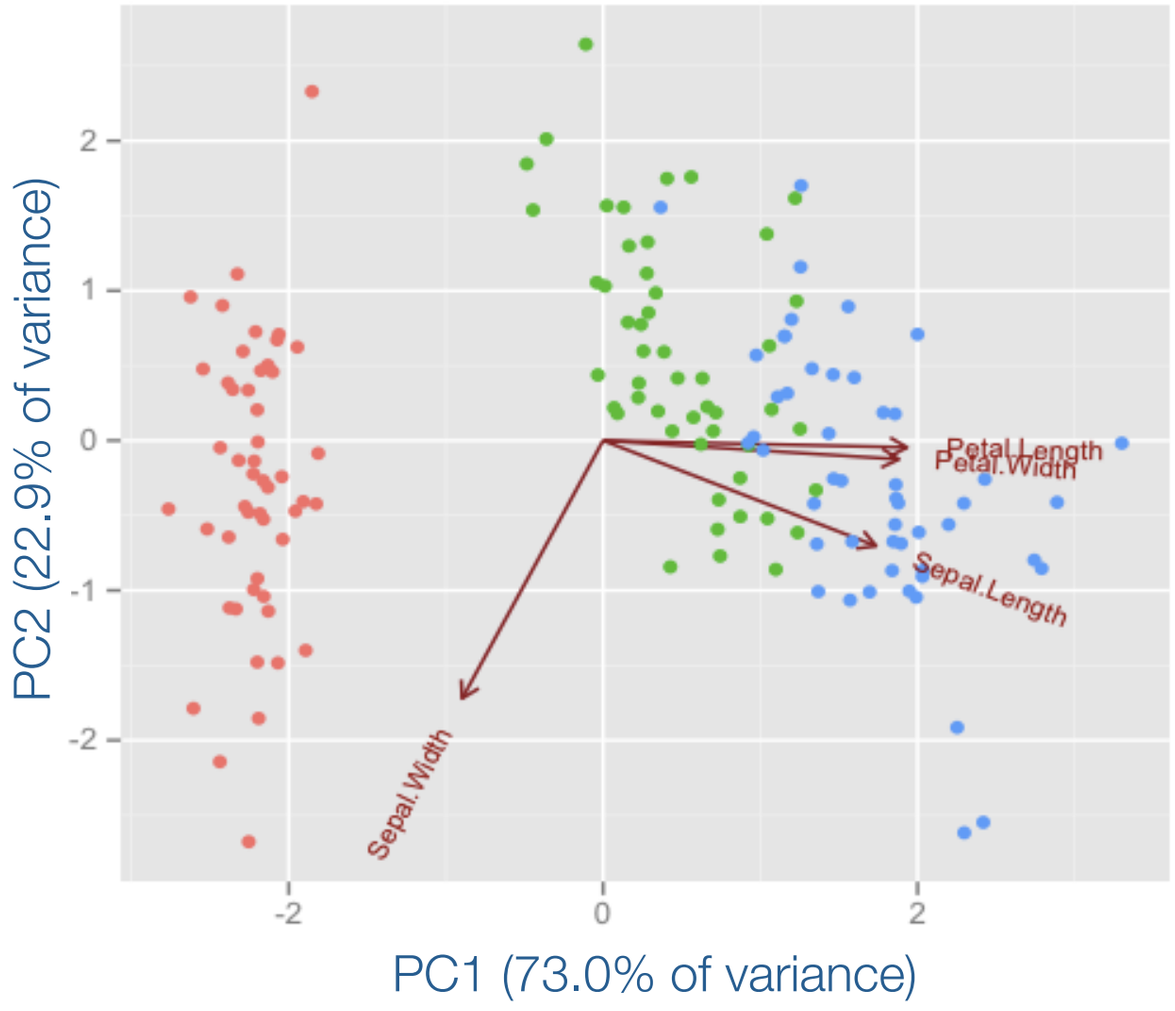
➡ The $k$ PCs with the highest variance provide a new reduced $n$ x $k$ representation of the data.

# Example: PCA

Apply PCA to *Iris* data set, and examine amount of variance in each Principal Component.

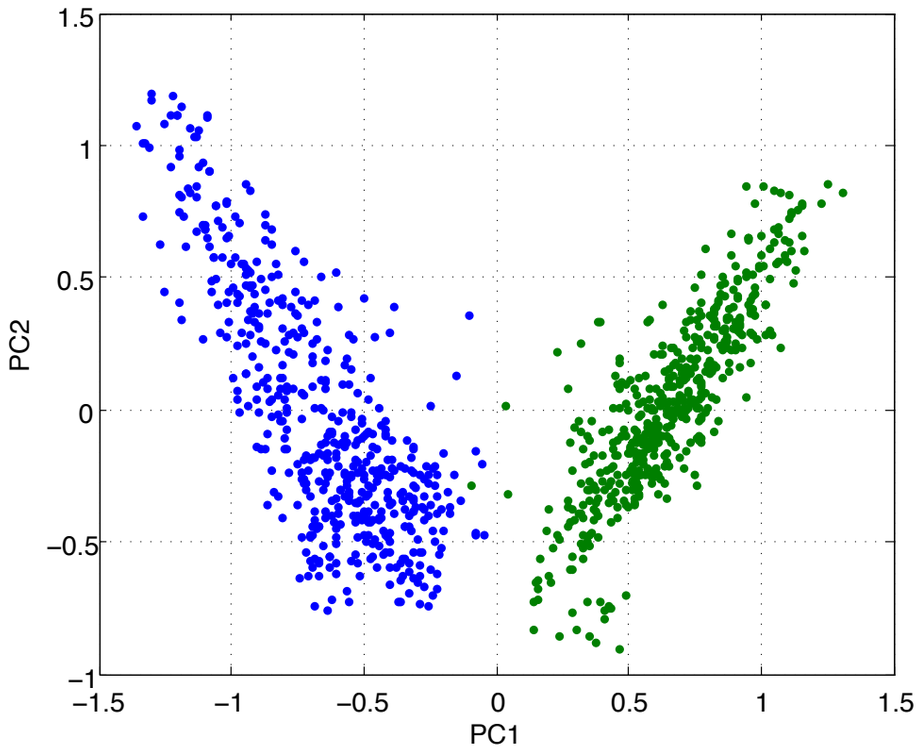



First two Principal Components account for ~96% of the variance in the data.

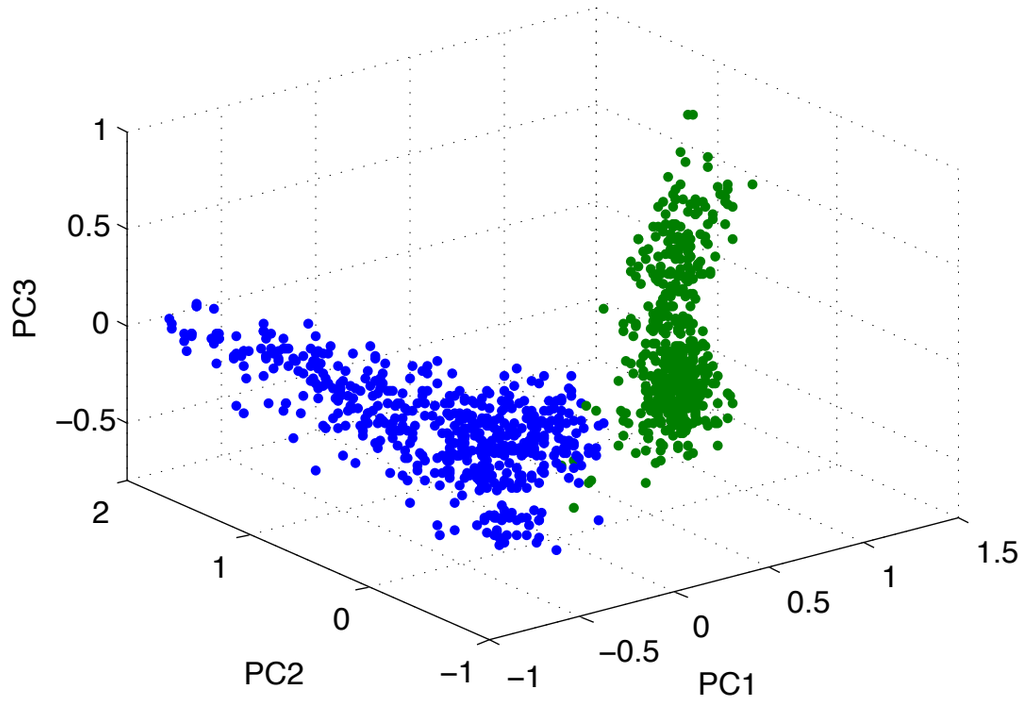


# Example: PCA

- Collection of 1,021 BBC news articles on business + sport, represented by high-dimensional space with 5,570 features (words).
- Applying PCA allows us to visualise the data in a low dimensional space using a small number of PCs.



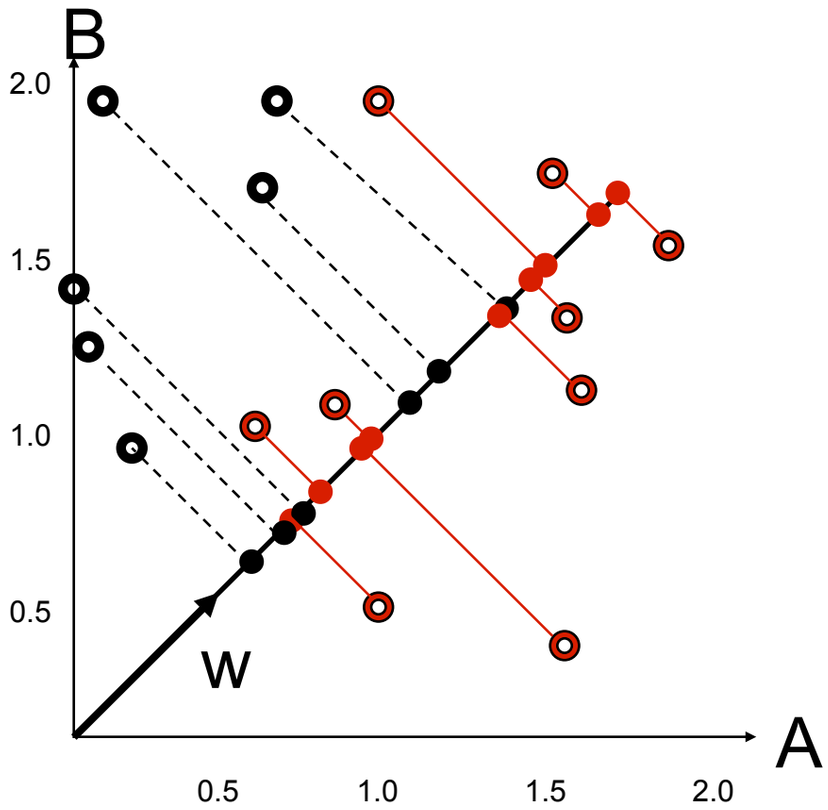

2 leading principal components (PCs)



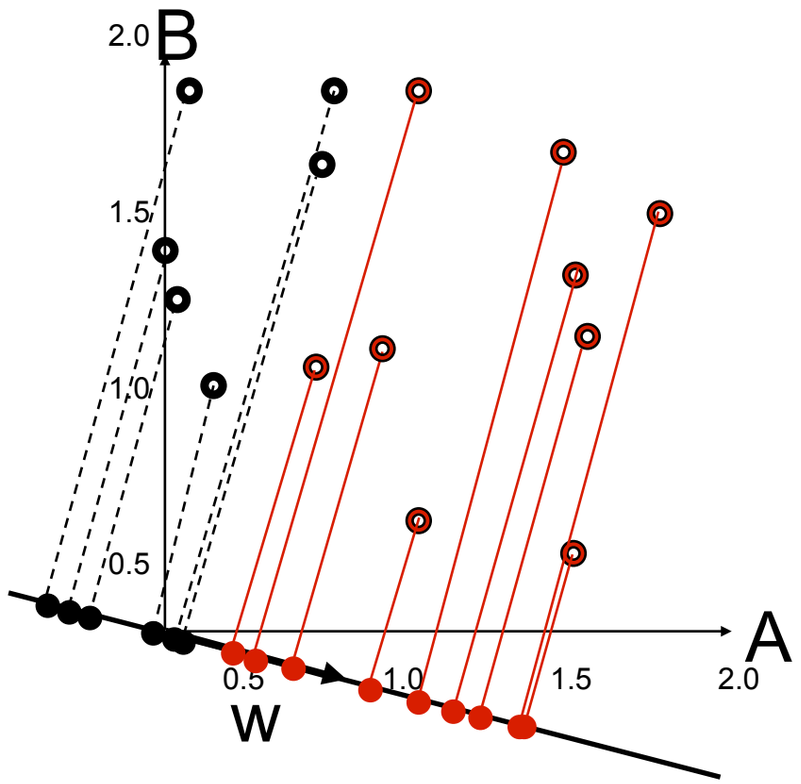

3 leading principal components (PCs)

# Linear Discriminant Analysis (LDA)

- PCA is an unsupervised feature transformation technique. The resulting PCs are not always good for discrimination in classification.
- Linear Discriminant Analysis (LDA): Seeks to create discriminating features, by finding a linear transformation that maximises the between-class variance and minimises the within-class variance.



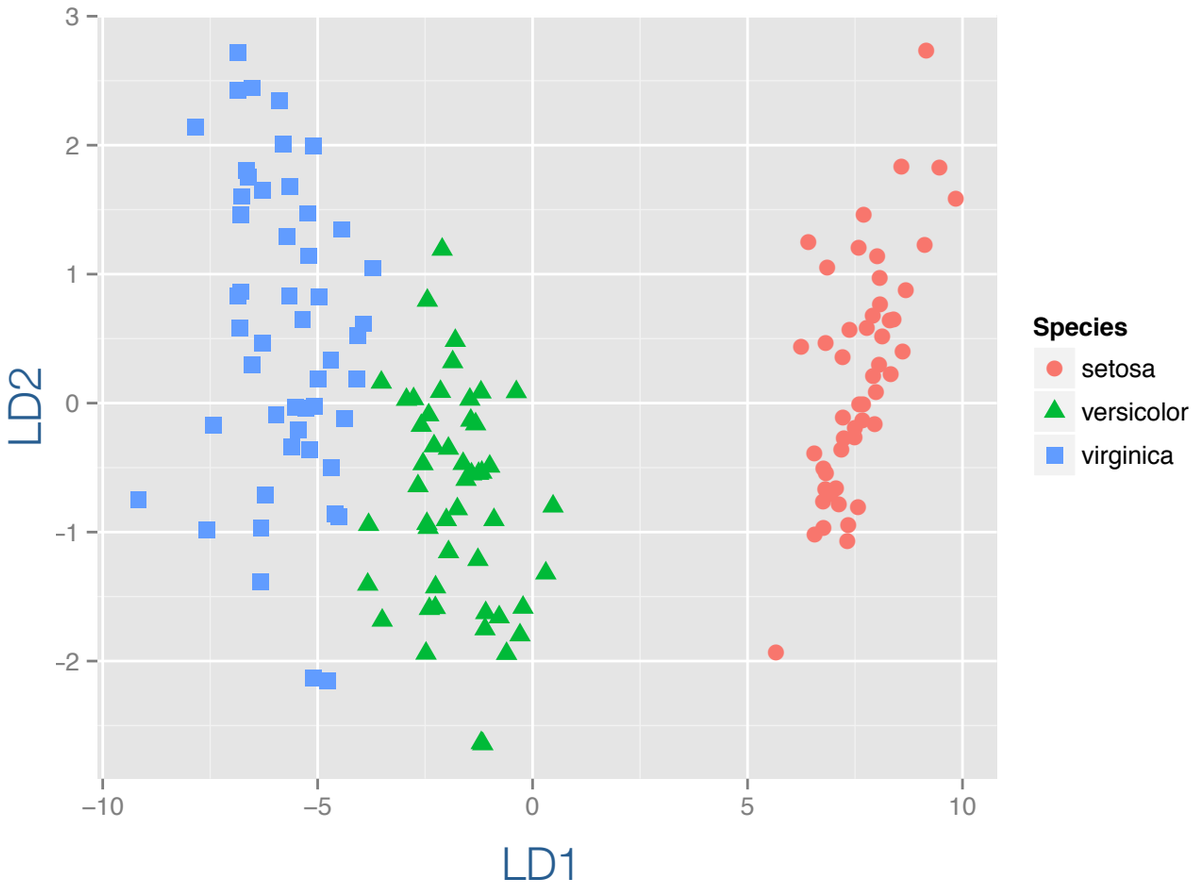

PCA does not separate classes



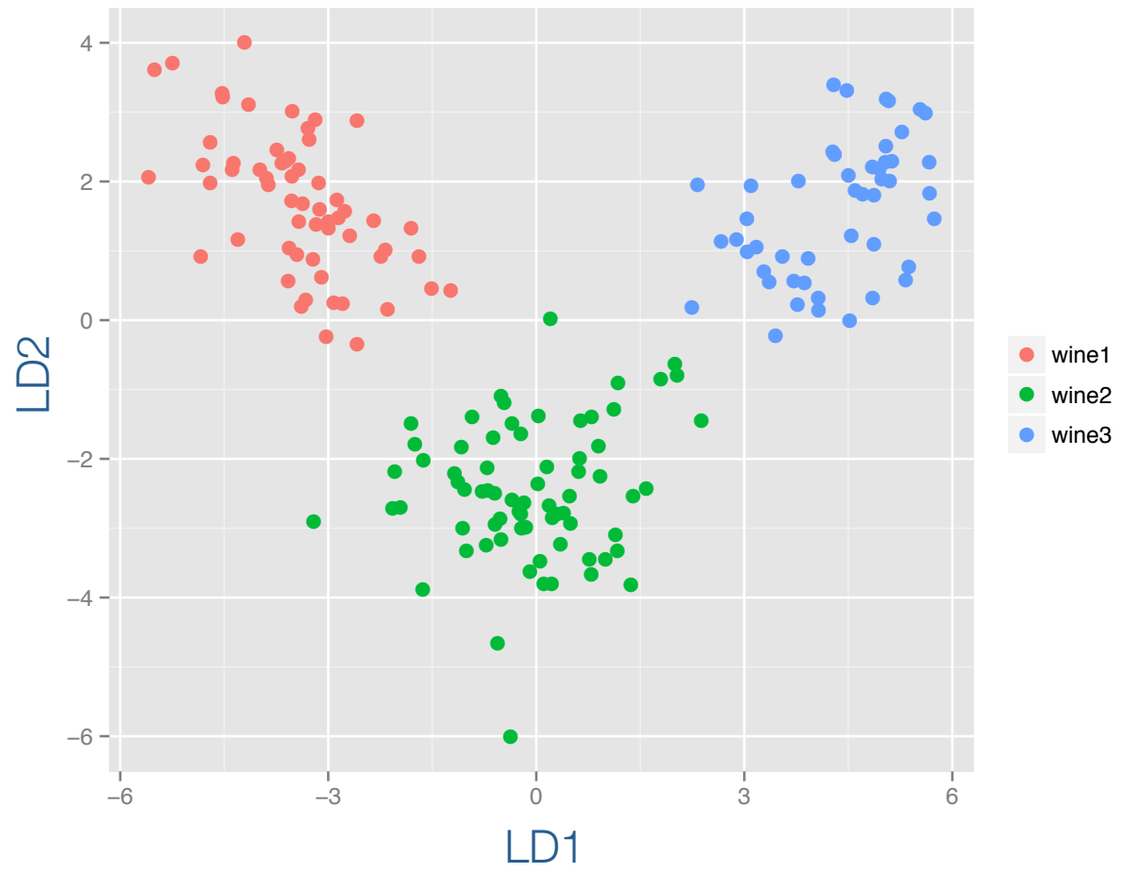

LDA discovers a discriminating projection

# Example: LDA

LDA applied to *Iris* data set. Attempts to maximise the variance between the three labelled classes.




LDA applied to *Wine* data set. Attempts to maximise the variance between the three labelled classes.



# Singular Value Decomposition (SVD)

- Performing SVD on the data is similar to PCA.
- We can rewrite any rectangular $m \times n$ matrix as $\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$

  Left singular vectors $\mathbf{U}$: $n \times n$ matrix, eigenvectors of $\mathbf{X} \cdot \mathbf{X}^T$

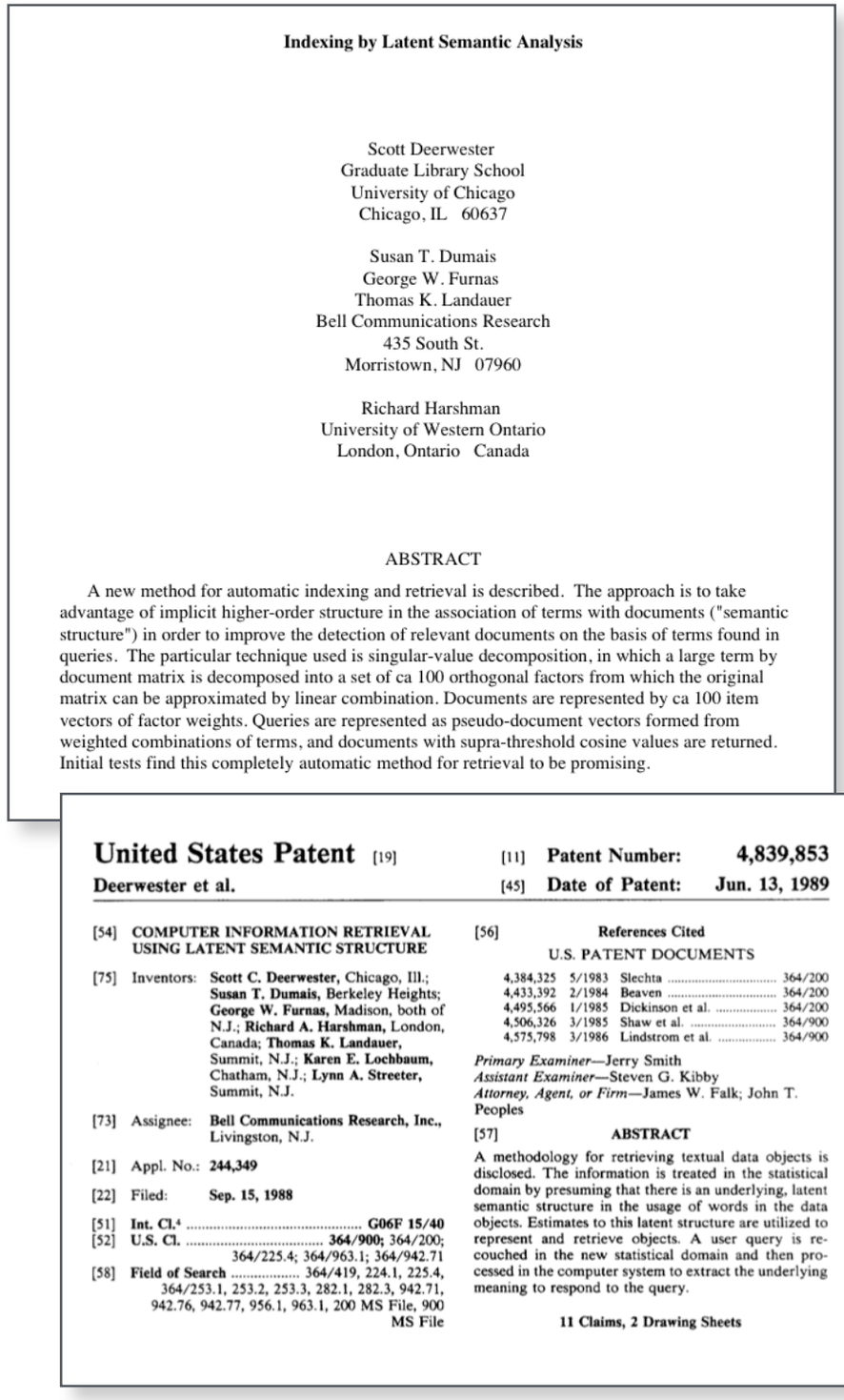
  Right singular vectors $\mathbf{V}$: $m \times m$ matrix, eigenvectors of $\mathbf{X}^T \cdot \mathbf{X}$

  Singular values $\mathbf{S}$: diagonal $n \times m$ matrix

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | = | -0.45 | -0.29 | -0.21 | -0.82 | • | 8.76 | 0 | • | -0.82 | 0.57 |
| 1 | 3 | | -0.29 | -0.26 | -0.80 | 0.46 | | 0 | 7.30 | | -0.57 | -0.82 |
| 0 | 6 | | -0.39 | -0.67 | 0.54 | 0.32 | | 0 | 0 | | | |
| 8 | 0 | | -0.75 | 0.63 | 0.15 | 0.15 | | 0 | 0 | | | |
| $\mathbf{X}$ | | | $\mathbf{U}$ | | | | | $\mathbf{S}$ | | | $\mathbf{V}^T$ | |
| *4 x 2* | | | *4 x 4* | | | | | *4 x 2* | | | *2 x 2* | |

# Latent Semantic Indexing (LSI)

- Latent Semantic Indexing: a method for selecting informative subspaces of feature spaces.
- Developed for information retrieval to reveal semantic information from document co-occurrences.
- Terms that do not appear in a document may still be associated with the document.
- LSI derives uncorrelated dimensions that might be considered as the latent concepts in the data.



**Indexing by Latent Semantic Analysis**


Scott Deerwester
Graduate Library School
University of Chicago
Chicago, IL 60637

Susan T. Dumais
George W. Furnas
Thomas K. Landauer
Bell Communications Research
435 South St.
Morristown, NJ 07960

Richard Harshman
University of Western Ontario
London, Ontario Canada


ABSTRACT


A new method for automatic indexing and retrieval is described. The approach is to take advantage of implicit higher-order structure in the association of terms with documents ("semantic structure") in order to improve the detection of relevant documents on the basis of terms found in queries. The particular technique used is singular-value decomposition, in which a large term by document matrix is decomposed into a set of ca 100 orthogonal factors from which the original matrix can be approximated by linear combination. Documents are represented by ca 100 item vectors of factor weights. Queries are represented as pseudo-document vectors formed from weighted combinations of terms, and documents with supra-threshold cosine values are returned. Initial tests find this completely automatic method for retrieval to be promising.


**United States Patent** [19]

**Deerwester et al.**

[11] **Patent Number: 4,839,853**

[45] **Date of Patent: Jun. 13, 1989**

[54] **COMPUTER INFORMATION RETRIEVAL USING LATENT SEMANTIC STRUCTURE**


[75] Inventors: **Scott C. Deerwester,** Chicago, Ill.; **Susan T. Dumais,** Berkeley Heights; **George W. Furnas,** Madison, both of N.J.; **Richard A. Harshman,** London, Canada; **Thomas K. Landauer,** Summit, N.J.; **Karen E. Lochbaum,** Chatham, N.J.; **Lynn A. Streeter,** Summit, N.J.

[73] Assignee: **Bell Communications Research, Inc.,** Livingston, N.J.



[21] Appl. No.: **244,349**

[22] Filed: **Sep. 15, 1988**

[51] **Int. Cl.**[4] ..... **G06F 15/40**

[52] **U.S. Cl.** ..... **364/900;** 364/200; 364/225.4; 364/963.1; 364/942.71

[58] **Field of Search** ..... 364/419, 224.1, 225.4, 364/253.1, 253.2, 253.3, 282.1, 282.3, 942.71, 942.76, 942.77, 956.1, 963.1, 200 MS File, 900 MS File


[56] **References Cited**

U.S. PATENT DOCUMENTS


| | | | |
|---|---|---|---|
| 4,384,325 | 5/1983 | Slechta | 364/200 |
| 4,433,392 | 2/1984 | Beaven | 364/200 |
| 4,495,566 | 1/1985 | Dickinson et al. | 364/200 |
| 4,506,326 | 3/1985 | Shaw et al. | 364/900 |
| 4,575,798 | 3/1986 | Lindstrom et al. | 364/900 |



*Primary Examiner*—Jerry Smith
*Assistant Examiner*—Steven G. Kibby
*Attorney, Agent, or Firm*—James W. Falk; John T. Peoples


[57] **ABSTRACT**


A methodology for retrieving textual data objects is disclosed. The information is treated in the statistical domain by presuming that there is an underlying, latent semantic structure in the usage of words in the data objects. Estimates to this latent structure are utilized to represent and retrieve objects. A user query is re-couched in the new statistical domain and then processed in the computer system to extract the underlying meaning to respond to the query.


**11 Claims, 2 Drawing Sheets**

# Latent Semantic Indexing (LSI)

- Typically represent documents as high-dimensional vectors, and a complete collection as a $m$ x $n$ term-document matrix.

| | doc1 | doc2 | doc3 | doc4 | doc5 | doc6 |
|---|---|---|---|---|---|---|
| business | 1 | 1 | 1 | | 1 | |
| finance | | | | | | 1 |
| news | | 1 | 1 | | 1 | |
| apple | | 1 | 1 | 1 | | |
| food | | | | 1 | | |
| recipe | | | | 1 | | |
| car | | | | | 1 | |
| automobile | | | | | | 1 |

**Problems**

- Synonymy: different words with the same meaning (e.g "car" v "automobile").
- Polysemy: the same word having multiple meanings (e.g. "apple").

➡ Semantically related documents can appear to have low similarity.

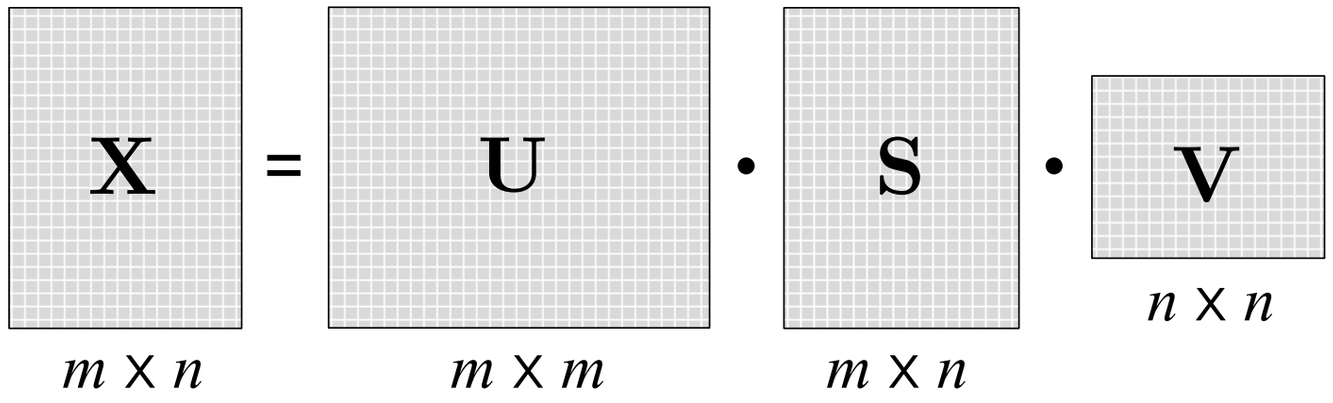
**LSI Solution:**

- Assume there is some underlying latent structure that is hidden by the high-dimensional representation.
- Transform the data to a reduced-dimensional space whose axes are "concepts" that effectively group similar words together.

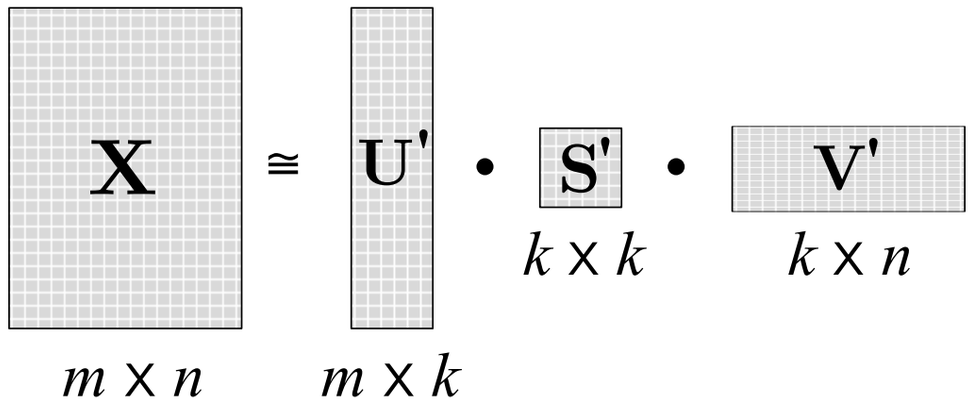
# Latent Semantic Indexing (LSI)

- Compute a $k$-dimensional truncated SVD of the original term-document matrix.

Standard SVD




Truncated SVD




- LSI only uses leading $k$ singular vectors, discards the others, where $k<<m$ and $k<<n$

# Example: LSI

- Collection of CS paper titles (index terms in italics)…

| | |
|---|---|
| c1 | *Human* machine *interface* for Lab ABS *computer* applications |
| c2 | A *survey* of *user* opinion of *computer system response time* |
| c3 | The EPS *user interface* management *system* |
| c4 | *System* and *human system* engineering testing of *EPS* |
| c5 | Relation of *user*-perceived *response time* to error measurement |
| m1 | The generation of random binary unordered *trees* |
| m2 | The intersection *graph* of paths in *trees* |
| m3 | *Graph minors* IV: Widths of *trees* and well quasi ordering |
| m4 | *Graph minors*: A *survey* |

# Example: LSI

- Collection of CS paper titles (index terms in italics)…

| | |
|---|---|
| c1 | *Human* machine *interface* for Lab ABS *computer* applications |
| c2 | A *survey* of *user* opinio |
| c3 | The EPS *user interface* |
| c4 | *System* and *human sy* |
| c5 | Relation of *user*-percei |
| m1 | The generation of rand |
| m2 | The intersection *graph* |
| m3 | *Graph minors* IV: Widt |
| m4 | *Graph minors*: A *surve* |

| | c1 | c2 | c3 | c4 | c5 | m1 | m2 | m3 | m4 |
|---|---|---|---|---|---|---|---|---|---|
| human | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| interface | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| computer | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| user | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| system | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| response | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| time | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| EPS | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| survey | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| trees | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| graph | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| minors | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

- Build 12 x 9 term-document matrix from index terms in titles.

# Example: LSI

- Apply truncated SVD to term-document matrix and extract vectors corresponding to 2 leading singular vectors.

| | |
|---|---|
| -0.22 | -0.11 |
| -0.2 | -0.07 |
| -0.24 | 0.04 |
| -0.4 | 0.06 |
| -0.64 | -0.17 |
| -0.27 | 0.11 |
| -0.27 | 0.11 |
| -0.3 | -0.14 |
| -0.21 | 0.27 |
| -0.01 | 0.49 |
| -0.04 | 0.62 |
| -0.03 | 0.45 |

**U**

*12* x *2*

| | |
|---|---|
| 3.34 | 0 |
| 0 | 2.54 |

**S**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| -0.20 | -0.61 | -0.46 | -0.54 | -0.28 | 0.00 | -0.01 | -0.02 | -0.08 |
| -0.06 | 0.17 | -0.13 | -0.23 | 0.11 | 0.19 | 0.44 | 0.62 | 0.53 |

$\mathbf{V}^T$

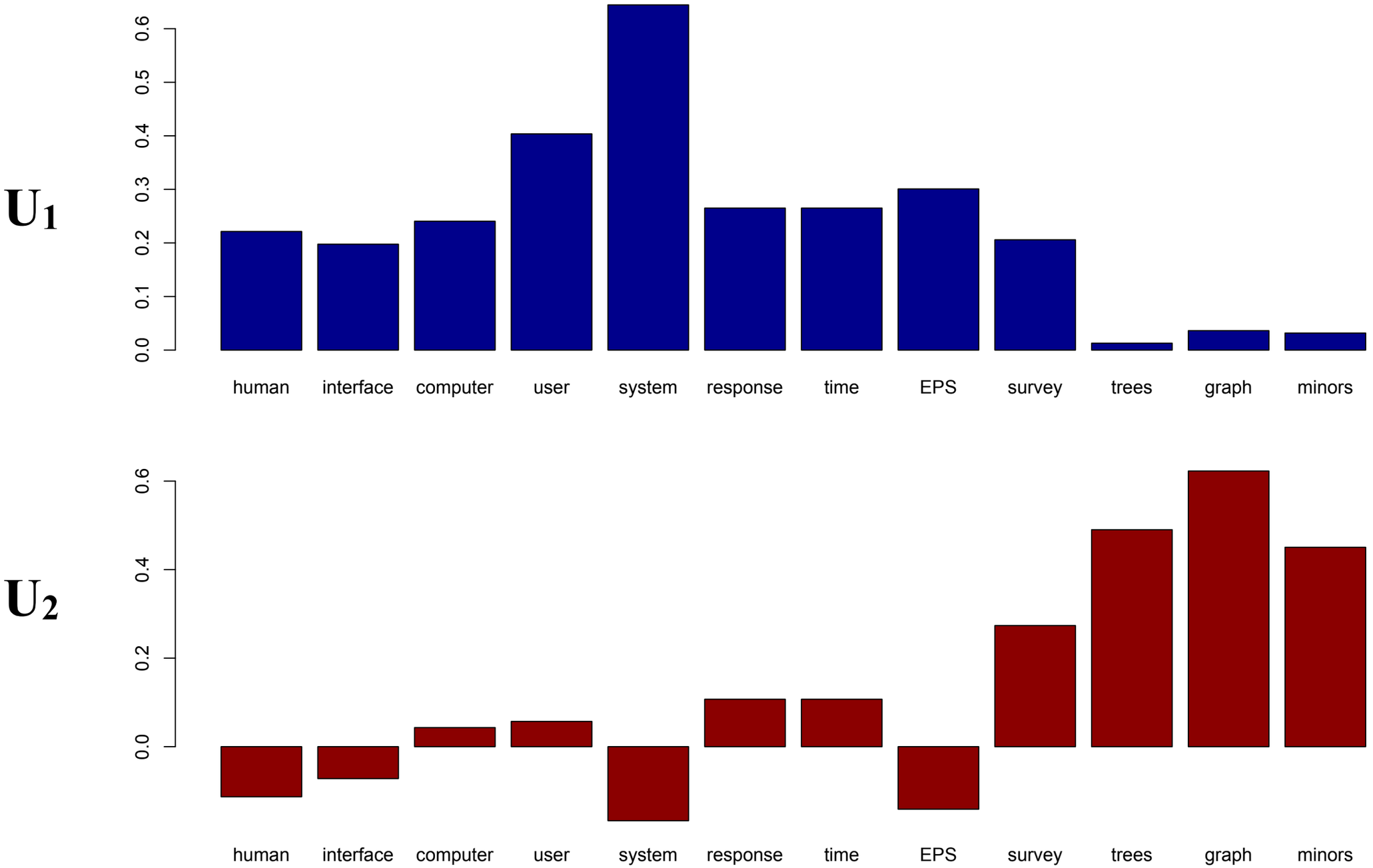
*2* x *9*

Values in $\mathbf{V}$ define coordinates for the original 9 documents in a new latent space.

Values in $\mathbf{U}$ describe the “concepts” in the latent space, and how they relate to the 12 original index terms.

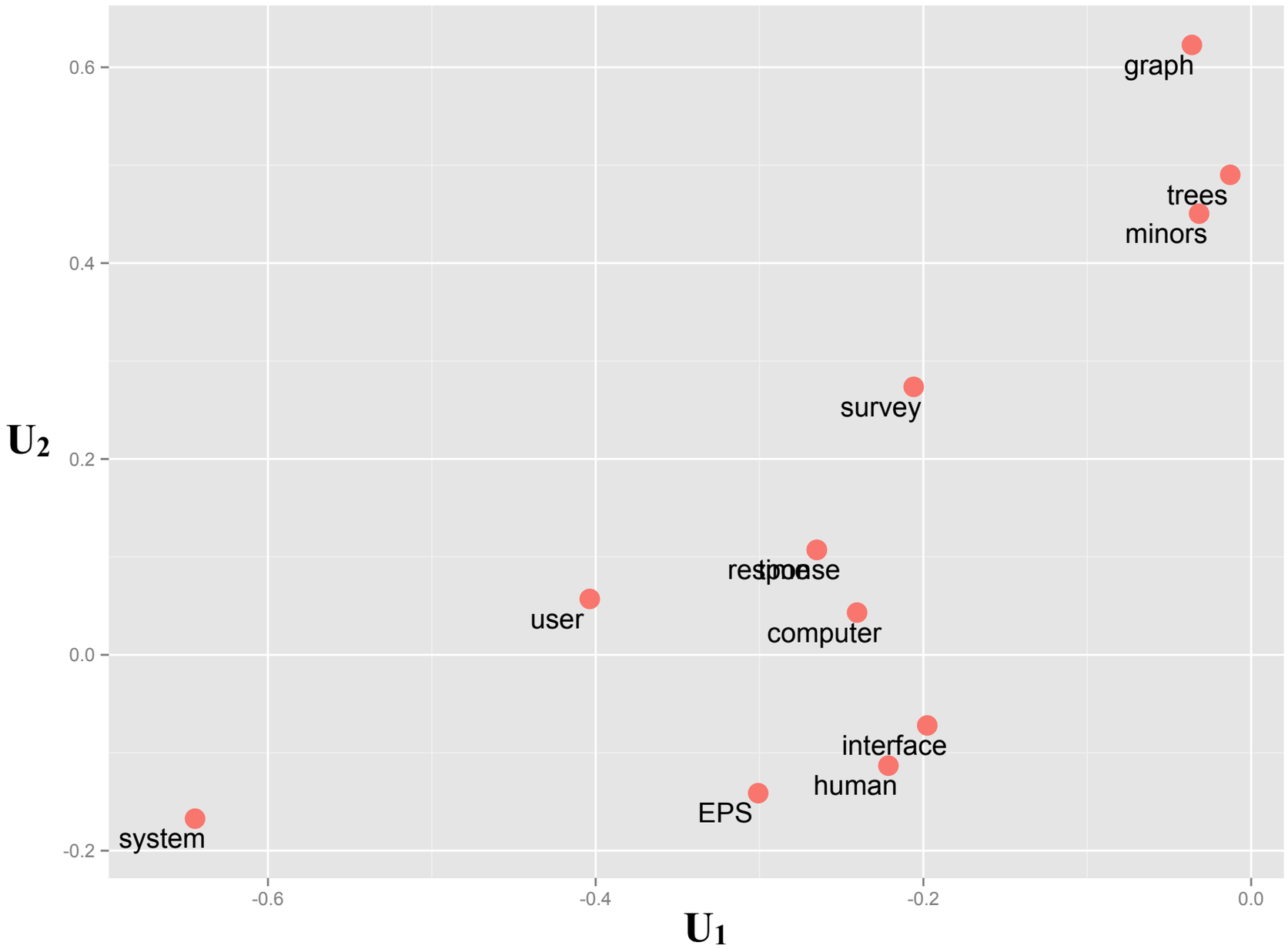
# Example: LSI

- Values in $\mathbf{U}$ describe the "concepts" in the 2D latent space, and how they relate to the 12 original index terms.



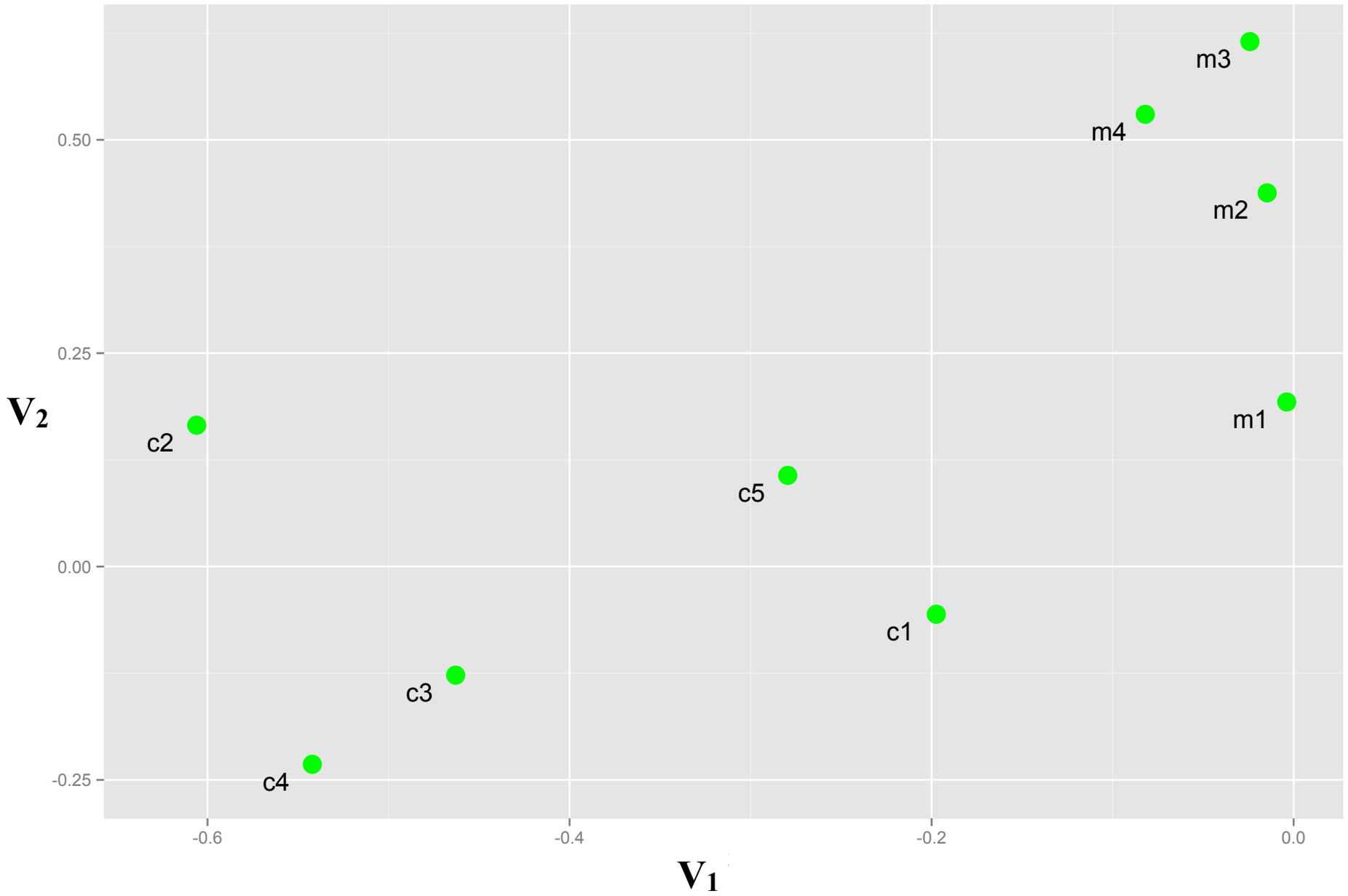
# Example: LSI

- Values in $\mathbf{U}$ describe the "concepts" in the 2D latent space, and how they relate to the 12 original index terms.



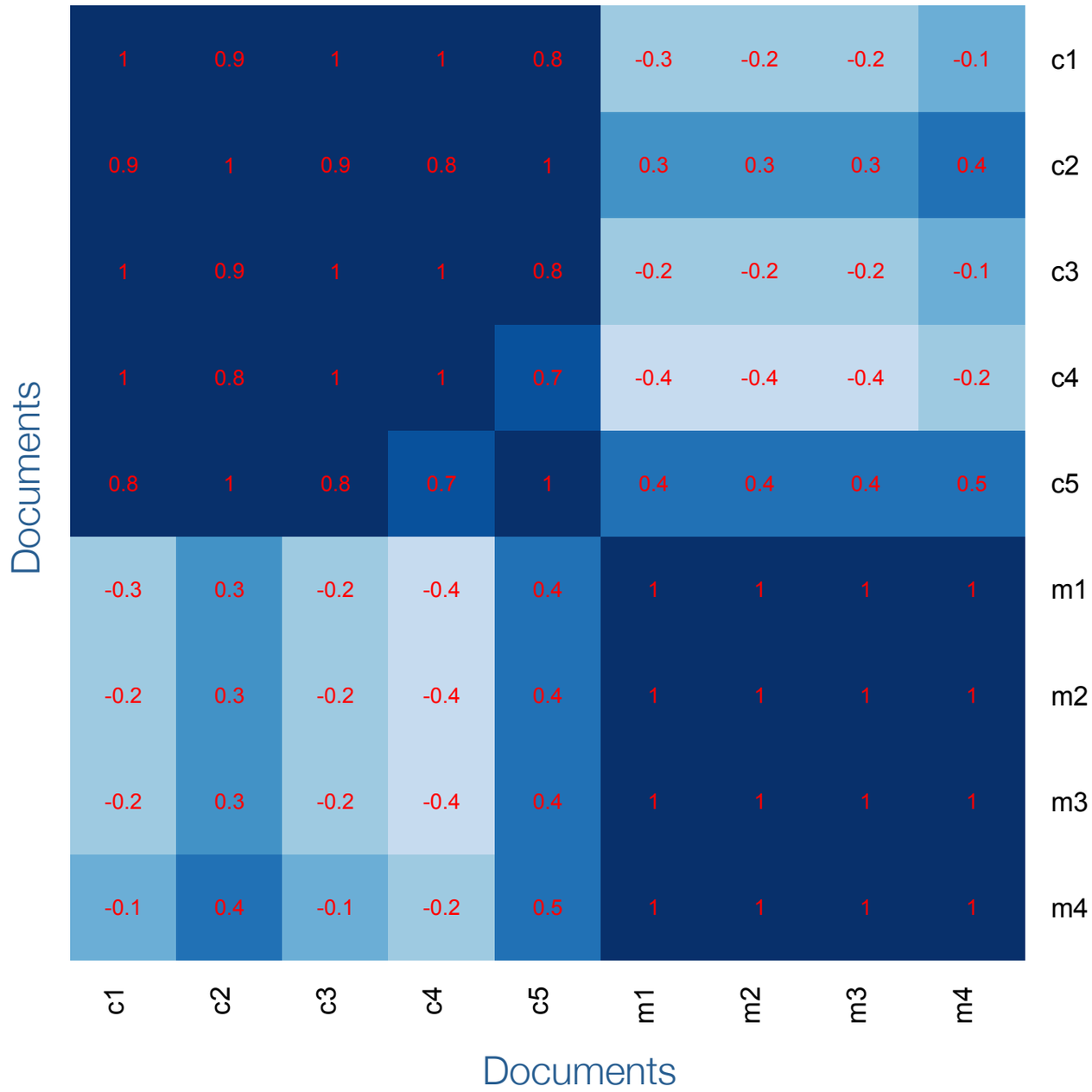
# Example: LSI

- Values in $\mathbf{V}$ define coordinates for the original 9 documents in a new 2D latent space.



# Example: LSI

- We can calculate a Cosine similarity matrix on the 2D latent space vectors in $\mathbf{V}$ to cluster documents.

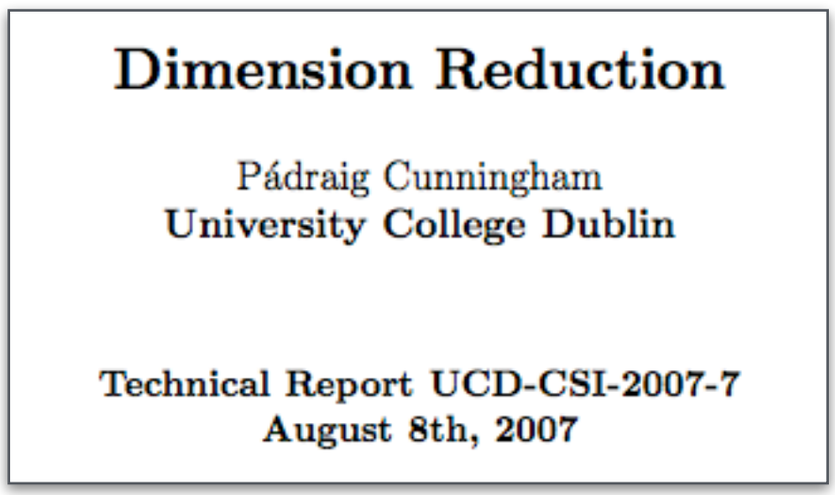



| | |
|---|---|
| c1 | *Human* machine *interface* for Lab ABS *computer* applications |
| c2 | A *survey* of *user* opinion of *computer system response time* |
| c3 | The EPS *user interface* management *system* |
| c4 | *System* and *human system* engineering testing of *EPS* |
| c5 | Relation of *user*-perceived *response time* to error measurement |
| m1 | The generation of random binary unordered *trees* |
| m2 | The intersection *graph* of paths in *trees* |
| m3 | *Graph minors* IV: Widths of *trees* and well quasi ordering |
| m4 | *Graph minors*: A *survey* |

We can find latent similarities between documents containing different terms (e.g. m1 & m4).

# Summary

- Feature Transformation v Selection
- Feature Transformation Methods
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Latent Semantic Indexing (LSI)


**Dimension Reduction**

Pádraig Cunningham
University College Dublin

Technical Report UCD-CSI-2007-7
August 8th, 2007

Technical Report on Moodle page

https://csiweb.ucd.ie/files/UCD-CSI-2007-7.pdf

# References


- E. Alpaydin. "Introduction to Machine Learning", Adaptive Computation and Machine Learning series, MIT press, 2009.
- P. Flach. “Machine Learning: The Art and Science of Algorithms that Make Sense of Data”. Cambridge University Press, 2012.
- S. Deerwester et al, “Improving Information Retrieval with Latent Semantic Indexing”, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 1988.
- P. Cunningham. "Dimension reduction”, UCD CS Technical report UCD-CSI-2007-5, 2007.