Matrix Decomposition Techniques in Machine Learning and Information Retrieval

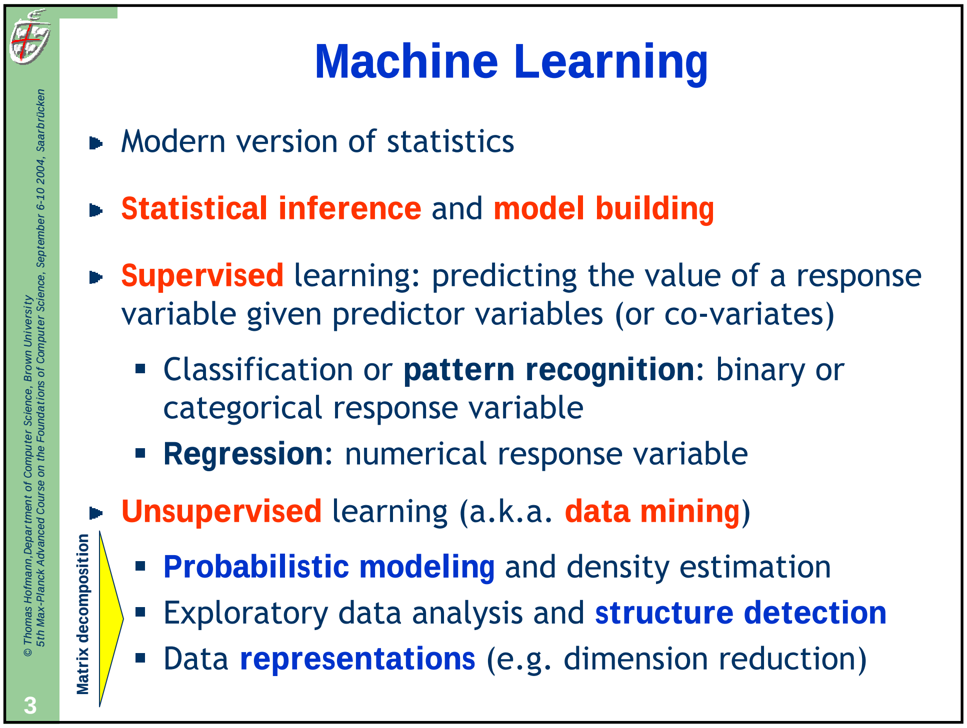
Thomas Hofmann

Associate Professor
Department of Computer Science
Brown University

th@cs.brown.edu
www.cs.brown.edu/~th

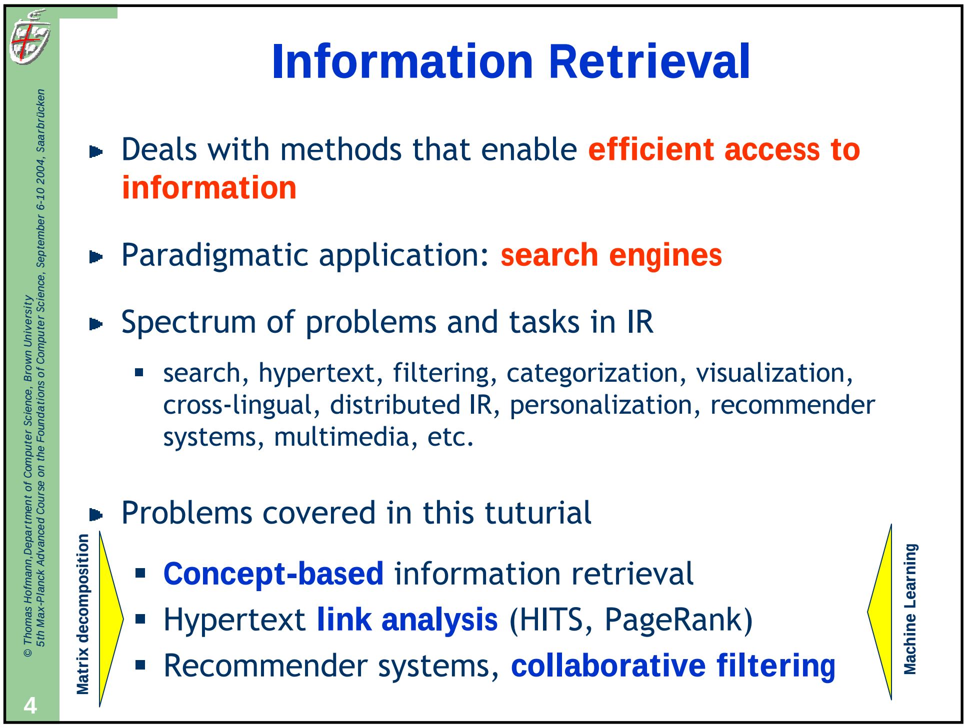
0.

Introduction



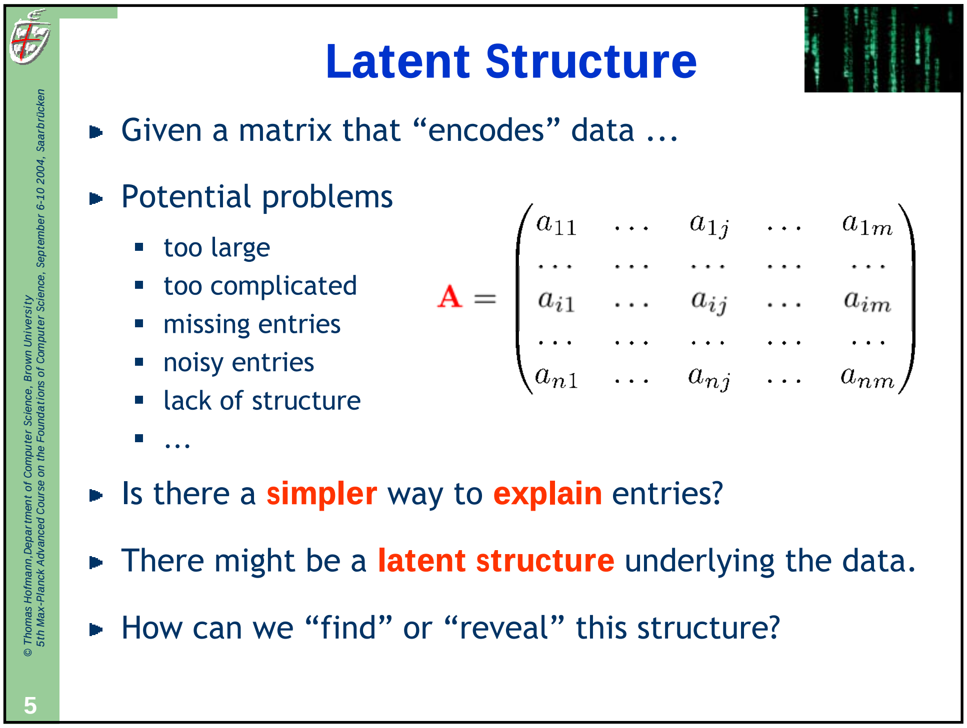
Machine Learning

- Modern version of statistics
- **Statistical inference** and **model building**
- **Supervised** learning: predicting the value of a response variable given predictor variables (or co-variates)
  - Classification or **pattern recognition**: binary or categorical response variable
  - **Regression**: numerical response variable
- **Unsupervised** learning (a.k.a. **data mining**)
  - **Probabilistic modeling** and density estimation
  - Exploratory data analysis and **structure detection**
  - Data **representations** (e.g. dimension reduction)



Information Retrieval

- Deals with methods that enable **efficient access to information**
- Paradigmatic application: **search engines**
- Spectrum of problems and tasks in IR
  - search, hypertext, filtering, categorization, visualization, cross-lingual, distributed IR, personalization, recommender systems, multimedia, etc.
- Problems covered in this tuturial
  - **Concept-based** information retrieval
  - Hypertext **link analysis** (HITS, PageRank)
  - Recommender systems, **collaborative filtering**

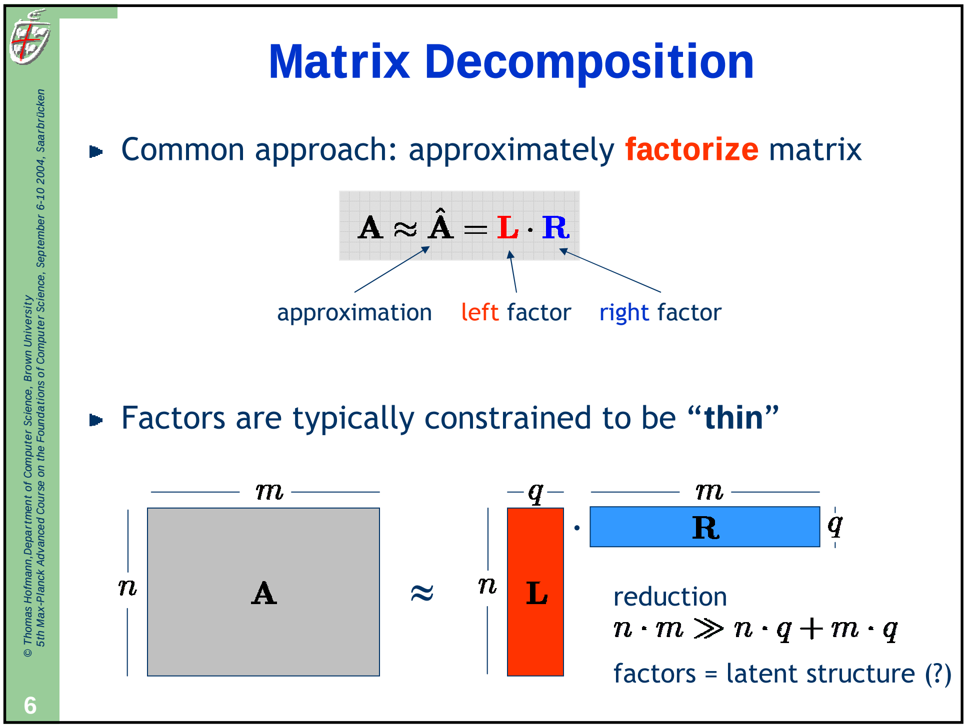


Latent Structure

- Given a matrix that "encodes" data …
- Potential problems
  - too large
  - too complicated
  - missing entries
  - noisy entries
  - lack of structure
  - …

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{im} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nm} \end{pmatrix}$$

- Is there a **simpler** way to **explain** entries?
- There might be a **latent structure** underlying the data.
- How can we "find" or "reveal" this structure?



Matrix Decomposition

- Common approach: approximately **factorize** matrix

$$\mathbf{A} \approx \hat{\mathbf{A}} = \mathbf{L} \cdot \mathbf{R}$$

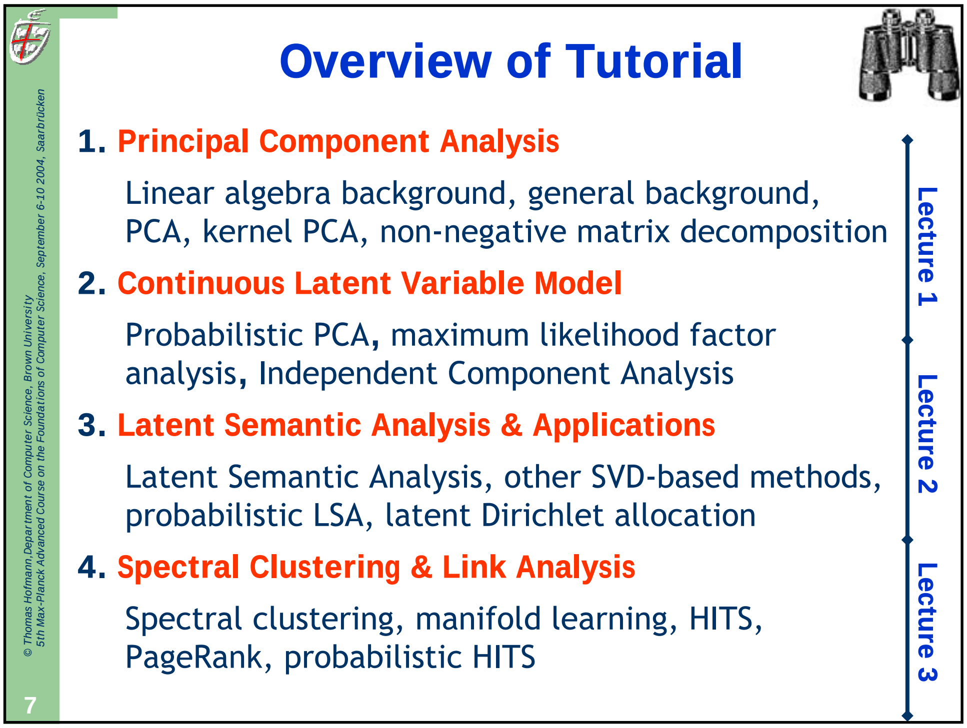
approximation left factor right factor

- Factors are typically constrained to be "**thin**"

reduction

$n \cdot m \gg n \cdot q + m \cdot q$

factors = latent structure (?)



Overview of Tutorial

1. **Principal Component Analysis**

   Linear algebra background, general background, PCA, kernel PCA, non-negative matrix decomposition

2. **Continuous Latent Variable Model**

   Probabilistic PCA, maximum likelihood factor analysis, Independent Component Analysis

3. **Latent Semantic Analysis & Applications**

   Latent Semantic Analysis, other SVD-based methods, probabilistic LSA, latent Dirichlet allocation

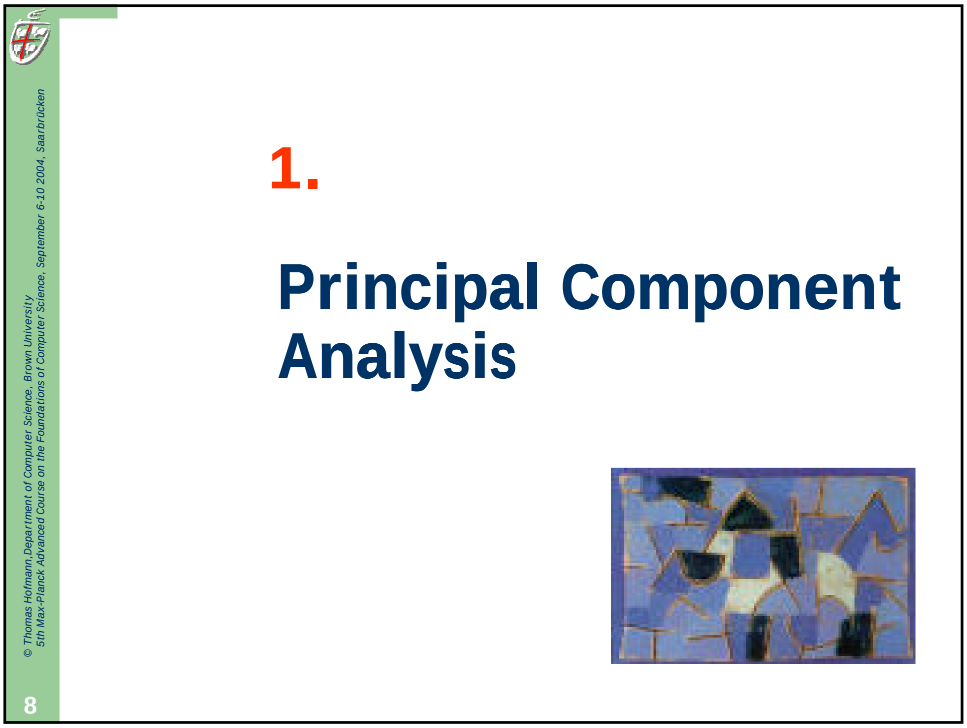
4. **Spectral Clustering & Link Analysis**

   Spectral clustering, manifold learning, HITS, PageRank, probabilistic HITS

Lecture 1

Lecture 2

Lecture 3



1.

Principal Component Analysis

1.1

Linear Algebra Background

Eigenvalues & Eigenvectors

- **Eigenvector equation** (for a square $m \times m$ matrix $\mathbf{S}$)

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

(right) eigenvector
$\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$

eigenvalue
$\lambda \in \mathbb{R}$

Example

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

- How many eigenvalues are there at most?

$$\mathbf{S}\mathbf{v} = \lambda \mathbf{v} \iff (\mathbf{S} - \lambda \mathbf{I})\, \mathbf{v} = \mathbf{0}$$

only has a non-zero solution if $|\mathbf{S} - \lambda \mathbf{I}| = 0$

this is a m-th order equation in λ which can have **at most m distinct solutions** (roots of the characteristic polynomial)

Eigenvalues & Eigenvectors

- For symmetric matrixes, eigenvectors for distinct eigenvalues are **orthogonal**

$$\mathbf{S}\mathbf{v}_{\{1,2\}} = \lambda_{\{1,2\}}\mathbf{v}_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \Rightarrow \langle \mathbf{v}_1, \mathbf{v}_2 \rangle = 0$$

  - proof left as an exercise

- All eigenvalues of a real symmetric matrix are **real**.

$$\text{for } \lambda \in \mathbb{C}, \text{ if } |\mathbf{S} - \lambda\mathbf{I}| = 0 \text{ and } \mathbf{S} = \mathbf{S}' \;\Rightarrow \lambda \in \mathbb{R}$$

  - proof left as an exercise

- All eigenvalues of a positive semidefinite matrix are **non-negative**

$$\overbrace{\mathbf{w}'\mathbf{S}\mathbf{w} \geq 0, \forall \mathbf{w} \in \mathbb{R}^m}, \text{ then if } \mathbf{S}\mathbf{v} = \lambda\mathbf{v} \;\Rightarrow \lambda \geq 0$$

  - proof left as an exercise

Eigen Decomposition

- Let $\mathbf{S} \in \mathbb{R}^{m \times m}$ be a **square** matrix with ***m*** **linearly independent eigenvectors** (a non-defective matrix)

- **Theorem**: Exists a (unique) **eigen decomposition** (cf. matrix diagonalization theorem)

diagonal

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

similarity transform

- Columns of $\mathbf{U}$ are **eigenvectors** of $\mathbf{S}$

- Diagonal elements of $\mathbf{\Lambda}$ are **eigenvalues** of $\mathbf{S}$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

  - proof left as an exercise

Symmetric Eigen Decomposition

- If $\mathbf{S} \in \mathbb{R}^{m \times m}$ is a **symmetric** matrix:
- **Theorem**: Exists a (unique) **eigen decomposition**

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}'$$

  - where $\mathbf{U} \in \mathbb{R}^{m \times m}$ is **orthogonal**

$$\left\{ \begin{array}{l} \mathbf{U}' = \mathbf{U}^{-1} \\ \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \delta_{ij} \end{array} \right.$$

columns are orthogonal and length normalized

Spectral Decomposition

- **Spectral decomposition theorem** (finite dimensional, symmetric case, in general: normal matrices/operators)
- Eigenvalue subspaces

$$\mathcal{U}_\lambda = \{\mathbf{u} : \mathbf{S}\mathbf{u} = \lambda\mathbf{u}\} = \ker(\mathbf{S} - \lambda\mathbf{I})$$

- Direct sum representation

$$\mathbb{R}^m = \bigoplus_{\lambda \in \lambda(\mathbf{S})} \mathcal{U}_\lambda$$

- Projection matrix representation

$$\mathbf{S} = \sum_{\lambda \in \lambda(\mathbf{S})} \mathbf{P}_\lambda \longleftarrow$$

commuting orthogonal projection matrices

Singular Value Decomposition

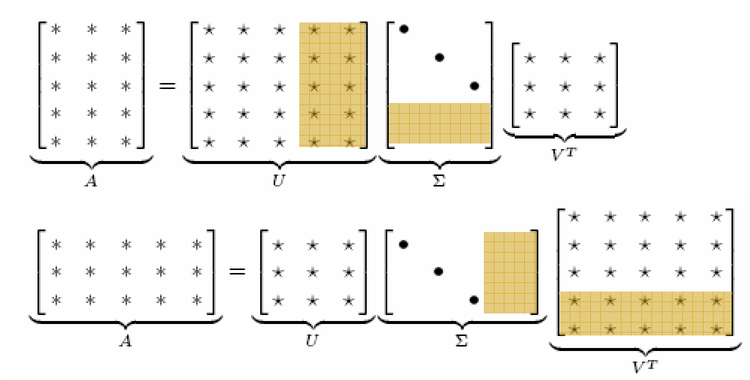
- For an arbitrary matrix $\mathbf{A}$ there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}' \in \mathbb{R}^{n \times m}$$

- Where
  - (i) $\mathbf{U} \in \Re^{n \times k}$ $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ $\mathbf{V} \in \mathbb{R}^{m \times k}$
  - (ii) $\mathbf{U}'\mathbf{U} = \mathbf{I}$ $\mathbf{V}'\mathbf{V} = \mathbf{I}$ *orthonormal columns*
  - (iii) $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_k), \; \sigma_i \geq \sigma_{i+1}$ *singular values (ordered)*
  - (iv) $k = \operatorname{rank}(\mathbf{A})$

Singular Value Decomposition

- Illustration of SVD dimensions and sparseness
- **Full SVD** (padded with zeros) vs. **reduced SVD**



Low-rank Approximation

- SVD can be used to compute optimal **low-rank approximations**.
- Approximation problem:

$$\mathbf{X}^* = \underset{\hat{\mathbf{X}}:\mathbf{rank}(\mathbf{X})=\mathbf{q}}{\operatorname{argmin}} \|\mathbf{X}-\hat{\mathbf{X}}\|_F$$

Frobenius norm

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}.$$

- Solution via SVD

$$\mathbf{X}^* = \mathbf{U}\,\mathrm{diag}(\sigma_1,\ldots,\sigma_q,\underbrace{0,\ldots,0}_{\text{set small singular values to zero}})\,\mathbf{V}'$$

$$\mathbf{X}^* = \sum_{r=1}^{q} \sigma_r \mathbf{u}_r \mathbf{v}_r'$$

column notation: sum of rank 1 matrices

C. Eckart, G. Young, *The approximation of a matrix by another of lower rank.* Psychometrika, 1, 211-218, 1936.

1.2

General Background

Pattern Matrix

- Statistics and machine learning typically starts from data given in the form of **observations**, **feature vectors** or **patterns**
- Feature vectors (in some m-dimensional Euclidean space)

$$\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^m, \quad i = 1, \ldots, n$$

- Patterns can be summarizes into the **pattern matrix**

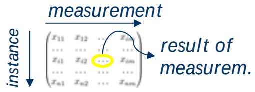
$$\mathbf{X} \in \mathbb{R}^{n \times m}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}'_1 \\ \cdots \\ \mathbf{x}'_i \\ \cdots \\ \mathbf{x}'_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

transposed i-th pattern

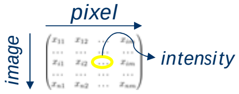
Examples: Pattern Matrices

$\mathbf{X} \in \mathbb{R}^{n \times m}$

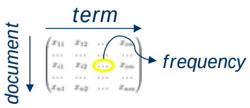
- **Measurement** vectors
  - i: instance number, e.g. a house
  - j: measurement, e.g. the area of a house



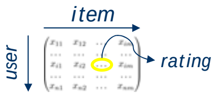
- **Digital images** as gray-scale **vectors**
  - i: image number
  - j: pixel value at location $j=(k,l)$



- **Text documents** in **bag-of-words** representation
  - i: document number
  - j: term (word or phrase) in a vocabulary



- **User rating** data
  - i: user number
  - j: item (book, movie)



Sample Covariance Matrix

- **Mean pattern** and **centered patterns**

$$\bar{\mathbf{x}} \equiv \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i, \quad \tilde{\mathbf{x}}_i \equiv \mathbf{x}_i - \bar{\mathbf{x}}, \quad \tilde{\mathbf{X}} \equiv \begin{pmatrix} \tilde{\mathbf{x}}'_1 \\ \tilde{\mathbf{x}}'_2 \\ \dots \\ \tilde{\mathbf{x}}'_n \end{pmatrix} = \mathbf{X} - \mathbf{1}_n \bar{\mathbf{x}}'$$

- **Sample covariance matrix** measures (empirical) correlations between different features or dimensions

$$\mathbf{S} \in \mathbb{R}^{m \times m}, \quad \mathbf{S} = (S_{rs})_{1 \le r,s \le m}, \quad S_{rs} = \frac{1}{n}\sum_{i=1}^{n} \tilde{x}_{ir}\tilde{x}_{is}$$

in terms of the pattern matrix $\mathbf{S} = \frac{1}{n}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$

1.3

Principal Component Analysis

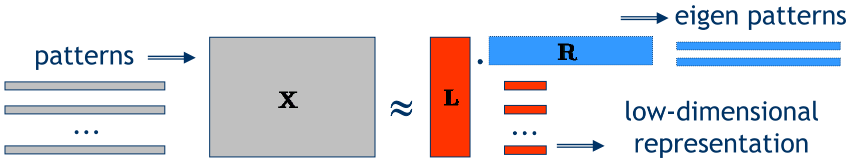
Principal Component Analysis

- The central idea of principal component analysis (PCA) is to **reduce the dimensionality** of a data set consisting of a large number of interrelated variables, while **retaining** as much as possible of the **variation** present in the data set. This is achieved by transforming to a new set of variables, the principal components, which are **uncorrelated** and which are ordered such that the first *few* retain most of the variation present in *all* of the original variables.

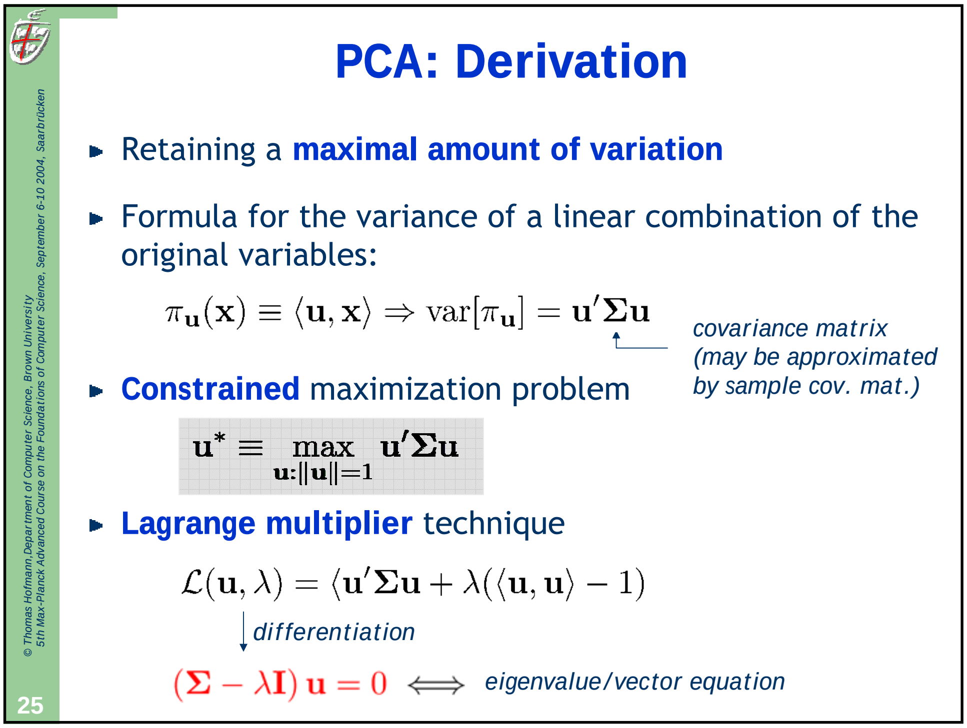
I.T. Jolliffe, Principal Component Analysis

I. T. Jolliffe, *Principal Component Analysis*, Springer, 2nd, 2002.

Principal Component Analysis



- Data matrix: **pattern matrix**
- Latent structure: low-dimensional (affine) **subspace**
- Decomposition: **eigen**-decomposition
- Applications: **workhorse** in machine learning, data mining, signal processing, computer vision, etc.



PCA: Derivation

- Retaining a **maximal amount of variation**
- Formula for the variance of a linear combination of the original variables:

$$\pi_{\mathbf{u}}(\mathbf{x}) \equiv \langle \mathbf{u}, \mathbf{x} \rangle \Rightarrow \mathrm{var}[\pi_{\mathbf{u}}] = \mathbf{u}'\boldsymbol{\Sigma}\mathbf{u}$$

covariance matrix (may be approximated by sample cov. mat.)

- **Constrained** maximization problem

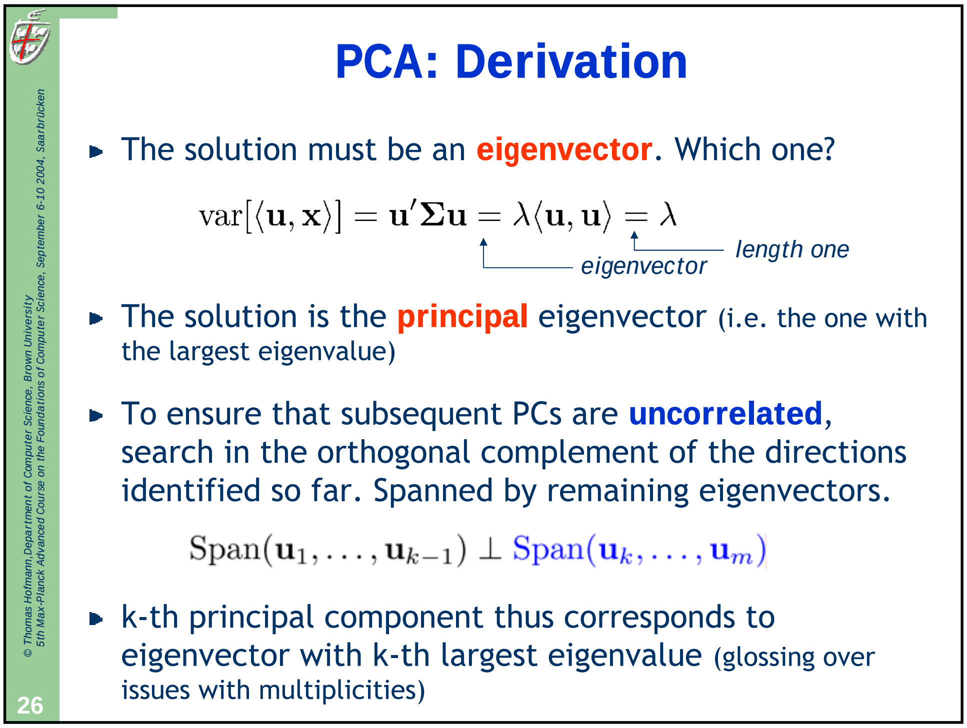
$$\mathbf{u}^* \equiv \max_{\mathbf{u}:\|\mathbf{u}\|=1} \mathbf{u}'\boldsymbol{\Sigma}\mathbf{u}$$

- **Lagrange multiplier** technique

$$\mathcal{L}(\mathbf{u}, \lambda) = \langle \mathbf{u}'\boldsymbol{\Sigma}\mathbf{u} + \lambda(\langle \mathbf{u}, \mathbf{u} \rangle - 1)$$

$\downarrow$ *differentiation*

$$(\boldsymbol{\Sigma} - \lambda \mathbf{I})\,\mathbf{u} = 0 \iff$$ *eigenvalue/vector equation*



PCA: Derivation

- The solution must be an **eigenvector**. Which one?

$$\mathrm{var}[\langle \mathbf{u}, \mathbf{x} \rangle] = \mathbf{u}'\boldsymbol{\Sigma}\mathbf{u} = \lambda \langle \mathbf{u}, \mathbf{u} \rangle = \lambda$$

eigenvector *length one*

- The solution is the **principal** eigenvector (i.e. the one with the largest eigenvalue)
- To ensure that subsequent PCs are **uncorrelated**, search in the orthogonal complement of the directions identified so far. Spanned by remaining eigenvectors.

$$\mathrm{Span}(\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}) \perp \mathrm{Span}(\mathbf{u}_k, \ldots, \mathbf{u}_m)$$

- k-th principal component thus corresponds to eigenvector with k-th largest eigenvalue (glossing over issues with multiplicities)

Dimension Reduction via PCA

- Apply eigen-decomposition to covariance matrix
- **Project data** onto q **principal eigenvectors** (corresponding to largest eigenvalues)
- Idea: **Recover latent low-dimensional structure**

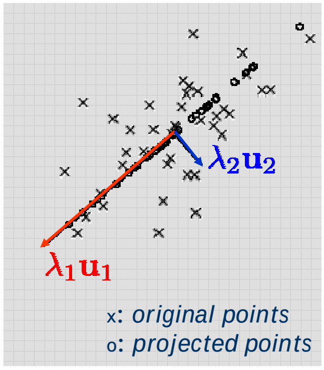
$$\mathbf{S} = \frac{1}{n}\tilde{\mathbf{X}}'\tilde{\mathbf{X}}$$

$$\mathbf{S} = \mathbf{U}'\mathrm{diag}(\lambda_1, \ldots, \lambda_q, \lambda_{q+1}, \ldots, \lambda_m)\mathbf{U}$$
$$\approx \mathbf{U}'\mathrm{diag}(\lambda_1, \ldots, \lambda_q, 0, \ldots, 0)\mathbf{U}$$

low-dimensional representation

$$\hat{\mathbf{x}} = \sum_{j=1}^{q} \langle \mathbf{u}_j, \mathbf{x} \rangle \mathbf{u}_j$$

principal component



PCA & Optimal Reconstruction

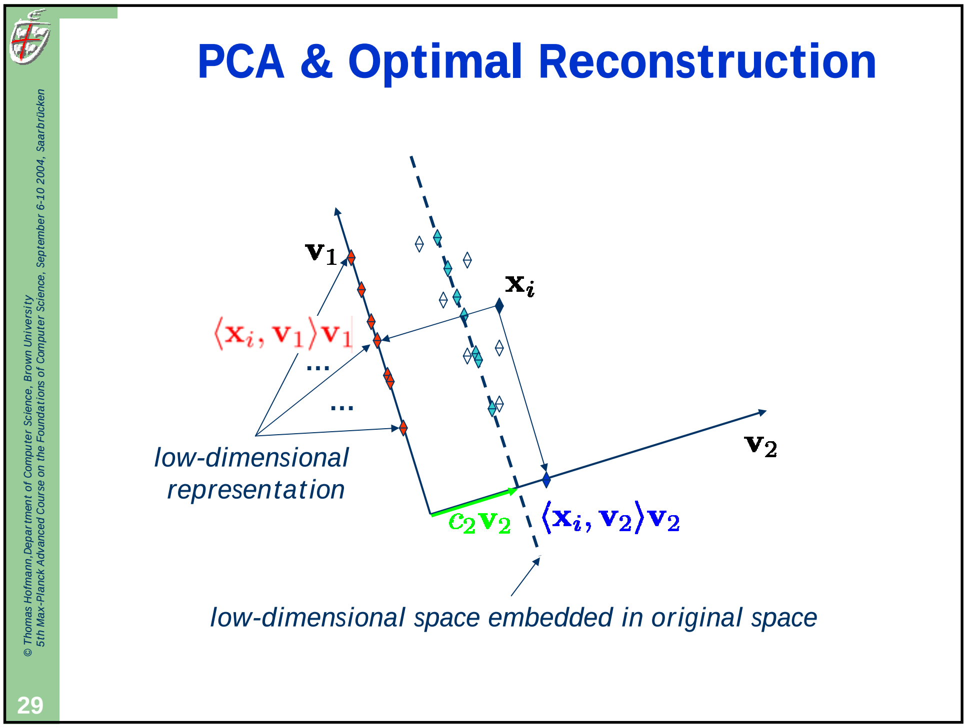
- **Theorem** (Pearson, 1901): PCA = Orthogonal linear projection with **minimal reconstruction error** in the least squares sense
- Express patterns in orthonormal basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$

$$\mathbf{x}_i = \sum_j w_{ij}\mathbf{v}_j, \quad w_{ij} \equiv \langle \mathbf{x}_i, \mathbf{v}_j \rangle$$

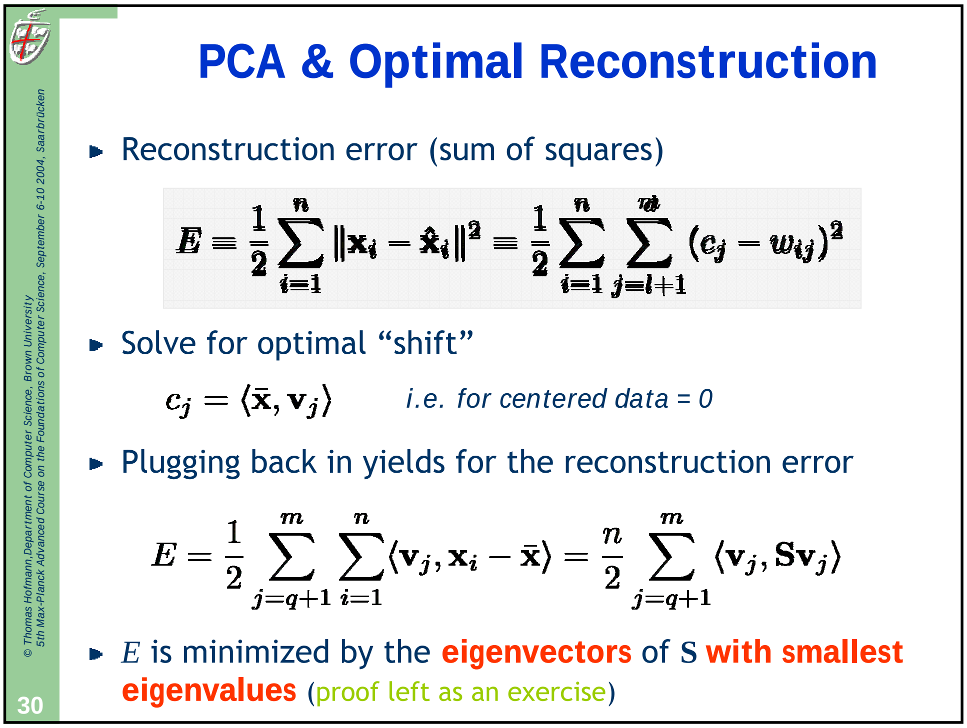
i.e. $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$

- **Low-dimensional approximation** (linear projection)

$$\hat{\mathbf{x}}_i = \underbrace{\sum_{j=1}^{q} w_{ij}\mathbf{v}_j}_{\text{preserved directions}} + \underbrace{\sum_{j=q+1}^{m} c_j\mathbf{v}_j}_{\text{projected away}}, \quad q \leq m$$



PCA & Optimal Reconstruction



PCA & Optimal Reconstruction

- Reconstruction error (sum of squares)

$$E = \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=l+1}^{m} (c_j - w_{ij})^2$$

- Solve for optimal “shift”

$c_j = \langle \bar{\mathbf{x}}, \mathbf{v}_j \rangle$ *i.e. for centered data = 0*

- Plugging back in yields for the reconstruction error

$$E = \frac{1}{2} \sum_{j=q+1}^{m} \sum_{i=1}^{n} \langle \mathbf{v}_j, \mathbf{x}_i - \bar{\mathbf{x}} \rangle = \frac{n}{2} \sum_{j=q+1}^{m} \langle \mathbf{v}_j, \mathbf{S}\mathbf{v}_j \rangle$$

- E is minimized by the **eigenvectors** of S **with smallest eigenvalues** (proof left as an exercise)

PCA & Optimal Reconstruction

- Optimal linear reconstruction (alternative view)
  - orthogonal projection $\pi(\mathbf{x}) = \mathbf{U}'(\mathbf{x} - \bar{\mathbf{x}})$

    columns are orthogonal $\mathbf{U} \in \mathbb{R}^{m \times q}, \ \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \delta_{ij}$
  - formula for optimal reconstruction

    $$\hat{\mathbf{x}} = \mathbf{U}\pi(\mathbf{x}) + \bar{\mathbf{x}}$$
  - proof left as an exercise

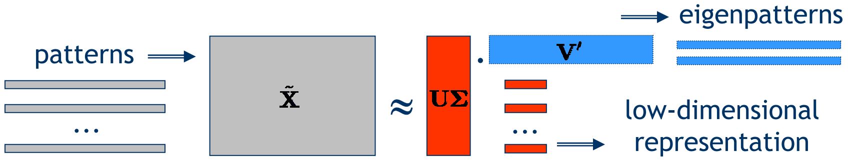
PCA via SVD

- **SVD of the pattern matrix** can be used to compute PCA

$$\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}' \Rightarrow$$

$$\mathbf{S} = \frac{1}{n}\tilde{\mathbf{X}}'\tilde{\mathbf{X}} = \frac{1}{n}(\mathbf{V}\mathbf{\Sigma}\underbrace{\mathbf{U}')(\mathbf{U}}_{=\mathbf{I}}\mathbf{\Sigma}\mathbf{V}') = \frac{1}{n}\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}'$$

- This shows: the rows of $\mathbf{V}$ are the eigenvectors of $\mathbf{S}$
- On the other hand $\tilde{\mathbf{X}}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$ which are just the PC scores (inner products between data and eigenvectors)



Application: Eigenfaces

- Example: application to face images
  - pattern vector encodes image pixel values in row scan
- Eigenfaces

average face

from www.cs.princeton.edu/~cdecoro/eigenfaces

from [Lee & Seung 1999]

B. Moghaddam and A. Pentland. *Face recognition using view-based and modular eigenspaces*. In SPIE, volume 2277, pages 12--21, 1994.

PCA: Applications

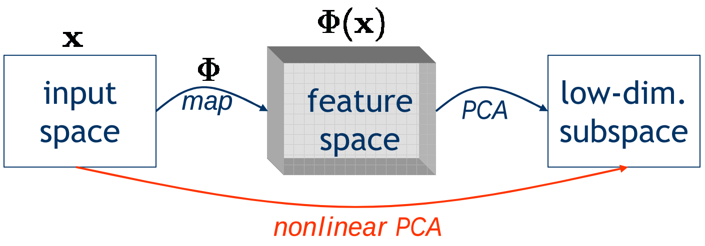
- Applications of PCA:
  - **Dimension reduction** as a preprocessing step for other learning algorithms or analysis steps (e.g. face detection & recognition)
  - **Recovering data manifolds:** finding affine data manifolds
  - **Data visualization** and **exploration** by plotting data in low-dimensional space
  - **Data denoising** and **reconstruction**
- Some Limitations
  - Linearity -> **nonlinear** and **kernel PCA**
  - Uncorrelated is not independent -> **independent CA** (ICA)
  - Probabilistic model/interpretation -> **probabilistic PCA**
  - Least squares approximation may be inappropriate -> **probabilistic Latent Semantic Analysis** (pLSA)
  - Constraints on sign of loadings -> **nonnegative matrix decomposition**

1.4

Kernel Principal Component Analysis

Non-linear PCA

- Sometimes original input features are insufficient or not powerful enough
- Idea: **Non-linear Principal Component Analysis**
  - Re-represent patterns by extracting **non-linear features** (feature space representation)
  - Usually a large (potentially infinite) number of non-linear features will be extracted
  - Use **PCA in feature space** to project down to a smaller number of features



Kernel PCA

- Explicit computation of non-linear features is often prohibitive or even impossible (infinite number of features)
- Idea:
  - Computation of PCA can be performed using **inner products** between feature vectors
  - **Implicit computation** of inner products in feature space using (Mercer) **kernels**
- Kernels
  - higher order features (polynomial kernels)

$$k(\mathbf{x},\mathbf{y}) = (\langle \mathbf{x},\mathbf{y}\rangle + 1)^{p} \;\Rightarrow\; \text{monomials of degree} \leq p$$

  - localized features

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{y}\|^2\right)$$

Kernel PCA

- Assume for simplicity data is centered in feature space
- Sample covariance matrix in feature space

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)'$$

- Eigenvector equation in feature space

$$\frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)\underbrace{\Phi(\mathbf{x}_i)'}_{\text{projects onto span of feature vector sample}}\mathbf{u} = \lambda\mathbf{u}$$

$$\Rightarrow \mathbf{u} = \sum_{i=1}^{n}\alpha_i\Phi(\mathbf{x}_i), \text{ with } \alpha_i \in \mathbb{R}$$

- Equations projected onto feature vectors (sufficient)

$$\langle\Phi(\mathbf{x}_i),\mathbf{S}\mathbf{u}\rangle = \lambda\langle\Phi(\mathbf{x}_i),\mathbf{u}\rangle,\ \forall i = 1,\ldots,n$$

B. Schölkopf, A. Smola, and K.-R. Müller. *Kernel principal component analysis*. In: Advances in Kernel Methods - SV Learning, pages 327-352. MIT Press, Cambridge, MA, 1999.

Kernel PCA

- Introducing the **Gram** or **kernel matrix**

$$\mathbf{K} \in \mathbb{R}^{n\times n},\ K_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle = k(\mathbf{x}_i, \mathbf{x}_j)$$

- One gets ...

$$\underbrace{\frac{1}{n}\left\langle \Phi(\mathbf{x}_i), \sum_{j=1}^{n} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)' \sum_{k=1}^{n} \alpha_k \Phi(\mathbf{x}_k)\right\rangle}_{\frac{1}{n}\mathbf{K}^2\alpha} \overset{\forall \boldsymbol{i}}{=} \lambda \underbrace{\left\langle \Phi(\mathbf{x}_i), \sum_{j=1}^{n} \alpha_j \Phi(\mathbf{x}_j)\right\rangle}_{\lambda\mathbf{K}\alpha}$$

$$\frac{1}{n}\mathbf{K}^2\alpha = \lambda\mathbf{K}\alpha$$

- Relevant solutions can be found by solving

$$\boxed{\mathbf{K}\alpha = n\lambda\alpha} \quad \Rightarrow$$ *eigen decomposition of Gram matrix*

Normalization & Pre-image Problem

- **Normalization** of eigenvectors in feature space

$$\langle \mathbf{u}, \mathbf{u}\rangle = \sum_{i,j} \alpha_i \alpha_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle = \langle \alpha, \mathbf{K}\alpha\rangle = \lambda \underbrace{\langle \alpha, \alpha\rangle}_{=\frac{1}{\lambda}} \overset{!}{=} 1$$

- Computing projections of new **test patterns**

$$\langle \Phi(\mathbf{x}), \mathbf{u}\rangle = \sum_{i} \alpha_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i)\rangle,\ \mathbf{u} = \sum_{i} \alpha_i \Phi(\mathbf{x}_i)$$

- **Reconstruction** in original space leads to **pre-image problem**

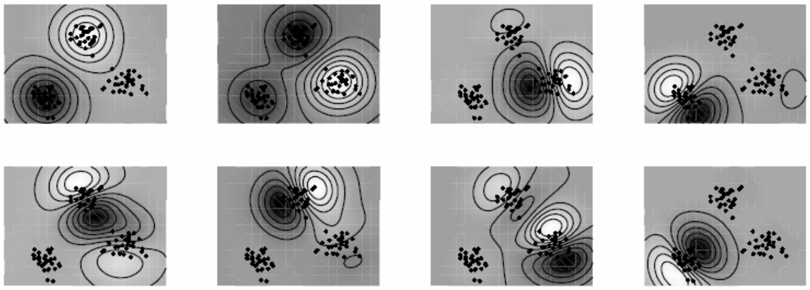
$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{z}} \|\Phi(\mathbf{z}) - \underbrace{\mathbf{P}\Phi(\mathbf{x})}_{\text{PCA projection}}\|^2$$

find pattern z who's feature representation is close to the PCA projection

Example: Kernel PCA

- Example: Kernel PCA on a synthetic data set
- Gaussian kernel

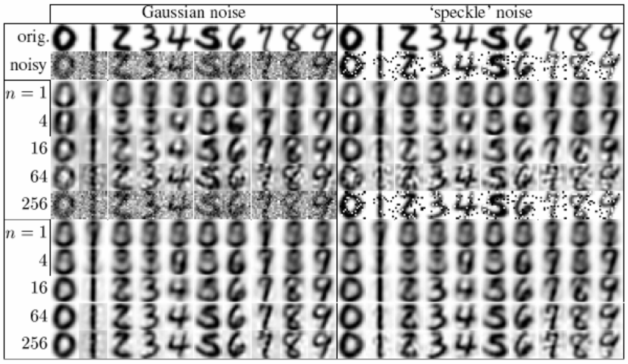
$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{y}\|^2\right)$$



(courtesy of Bernhard Schölkopf)

Example: Kernel PCA

- Application of kernel PCA for de-noising
- Perform non-linear PCA with Gaussian kernel on noisy images of handwritten digits



S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. *Kernel PCA and de-noising in feature spaces*, NIPS 11, pp. 536 - 542, Cambridge, MA, 1999. MIT Press.

1.5

Non-Negative Matrix Decomposition

Non-negative Matrix Decomposition

- Approximate low-rank matrix decomposition by **non-negative factors**

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{LR}, \quad \underbrace{\mathbf{L} \in \mathbb{R}^{n\times q}_{\geq 0}, \; \mathbf{R} \in \mathbb{R}^{q\times m}_{\geq 0}}_{\text{non-negativity constraints}}$$

- Motivation
  - Applied for non-negative matrices (e.g. pattern matrices with non-negative measurements such as counts)
  - Encode **prior knowledge** that latent factors are non-negative
  - Effect of factors is **accumulative** (i.e. no cancellations due to negative contributions)
  - **Probabilistic interpretation** (to come...)

D. D. Lee and H. S. Seung. *Learning the parts of objects by non-negative matrix factorization*. Nature 401, 788-791 (1999).

NMF: Approximation Criterion

- One needs a suitable **approximation criterion** to quantify the approximation error $\hat{\mathbf{X}} = \mathbf{LR}$
- **Squared error** criterion or **Frobenius norm**

$$E_{sq}(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - \hat{x}_{ij})^2 = \|\mathbf{X} - \hat{\mathbf{X}}\|_F$$

- **Divergence** criterion (generalized Kullback-Leibler divergence)

$$E_{div}(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(x_{ij} \log \frac{x_{ij}}{\hat{x}_{ij}} - x_{ij} + \hat{x}_{ij} \right)$$

Reduces to KL divergence, if matrices are normalized $\sum_{i,j} \hat{x}_{ij} = const.$

NMF: Multiplicative Update Rule

- Non-convex optimization problem (Frobenius norm)

$$(\mathbf{L}^*, \mathbf{R}^*) = \underset{\mathbf{L},\mathbf{R} \geq 0}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{LR}\|_F$$

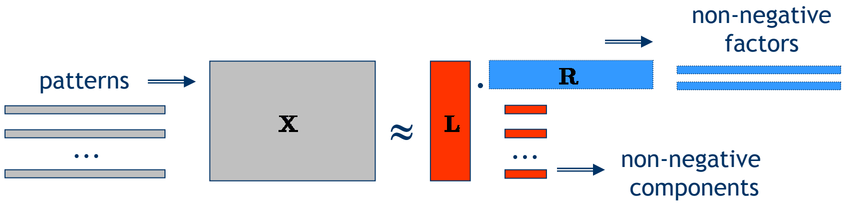
  - Convex in $\mathbf{L}$ given $\mathbf{R}$ and in $\mathbf{R}$ given $\mathbf{L}$, but not convex in both simultaneously. (*Resort to approximation algorithms.*)
- **Multiplicative updating**

$$r_{kj} \leftarrow r_{kj} \frac{(\mathbf{L}^T \mathbf{X})_{kj}}{(\mathbf{L}^T \underbrace{\mathbf{LR}}_{\hat{\mathbf{X}}})_{kj}} \qquad l_{ik} \leftarrow l_{ik} \frac{(\mathbf{X}\mathbf{R}^T)_{ik}}{(\underbrace{\mathbf{LR}}_{\hat{\mathbf{X}}}\mathbf{R}^T)_{ik}}$$

Convergence analysis: Frobenius norm criterion is non-increasing, fixed point corresponds to extremal point of criterion.

D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, NIPS 13, pp. 556-562, 2001.

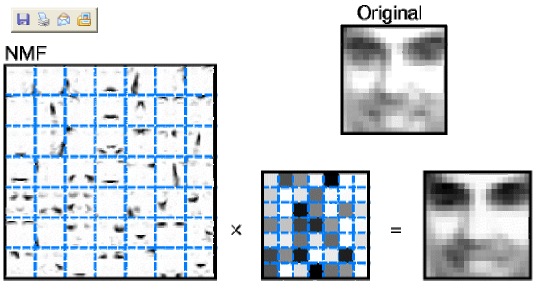
Non-negative Matrix Decomposition



- Data matrix: **pattern matrix**
- Latent structure: low-dimensional (affine) **subspace** spanned by **non-negative basis vectors**
- Decomposition: **non-convex** decomposition problem

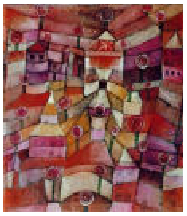
NMF: Application to Face Images

- Prominent application of NMF to automatically detect **parts** in images
- Idea:
  - Digital images can be represented as matrices with non-negative luminance values
  - Without cancellations, intensities will add up, hence factors may correspond to localized parts



2.

Continuous Latent Variable Models

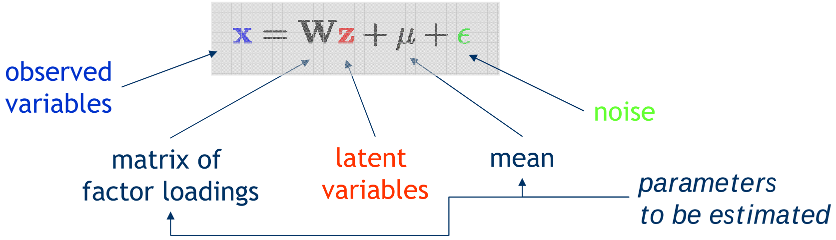


2.1

Probabilistic Principal Component Analysis

Probabilistic PCA

- **Generative probabilistic** model (density model)



$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \epsilon$$

observed variables

matrix of factor loadings

latent variables

mean

noise

parameters to be estimated

- Distributional assumptions (normality)

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\ \mathbf{z} \in \mathbb{R}^q \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}),\ \epsilon \in \mathbb{R}^m$$

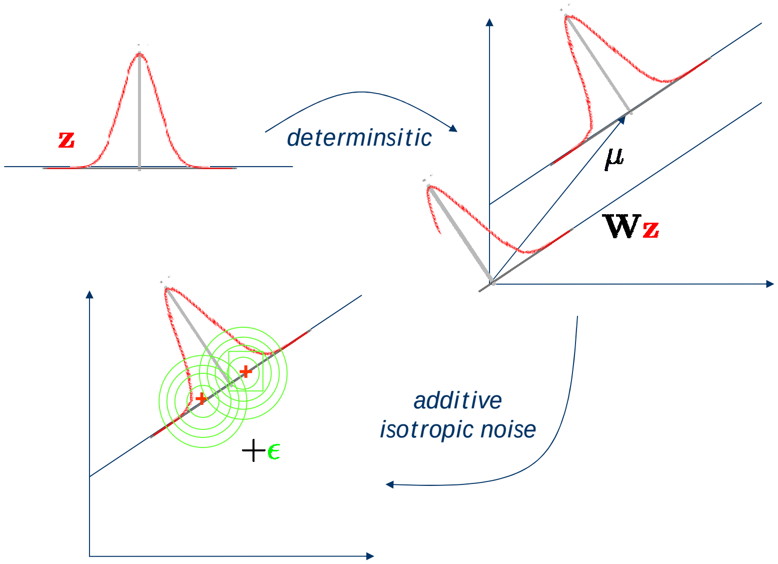
- Induced distribution on observables

$$\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}),\ \mathbf{x} \in \mathbb{R}^m$$

constrained Gaussian model

M. Tipping and C. Bishop, *Probabilistic principal component analysis*, Journal of the Royal Statistical Society, Series B 61(3), pp. 611-622, 1999

Probabilistic PCA: Illustration



Latent Variable Models

- Probabilistic PCA is a special case of a **continuous latent variable model**
- General ideas:

$p(\mathbf{x}, \mathbf{z})$	Define a **joint probability model** for observables and latent variables
$q \ll m$	Latent variables are smaller in number (e.g. low dimensional) or have a reduced state space
$p(\mathbf{x}\|\mathbf{z}) = \prod_j p(x_j\|\mathbf{z})$	Conditional distribution of observables given latent variables is assumed to be simple, typically based on **conditional independence**
$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$	**Integrating out** latent variables yields a probabilistic model for the observables
$p(\mathbf{z}\|\mathbf{x}) = \frac{p(\mathbf{x}\|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$	**Posterior probabilities** recover latent structure

B. S. Everitt, *An introduction to latent variable models.* Chapman & Hall, London, 1982.

Probabilistic PCA: Solution

- **Maximum likelihood estimation**

$$\mathcal{L} = -\frac{n}{2}\left(d \log(2\pi) + \log|\mathbf{C}| + \mathrm{trace}(\mathbf{C}^{-1}\mathbf{S})\right)$$

$$\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}$$

$$\hat{\mathbf{C}} = \mathbf{U}_q \mathbf{\Lambda}_q \mathbf{U}_q'$$

- MLE for **offset** μ is the mean (simple derivation)

$$\hat{\mu} = \frac{1}{n}\sum_i \mathbf{x}_i$$

- MLE for **loadings matrix** $\mathbf{W}$ is given by (involved derivation)

$$\hat{\mathbf{W}} = \mathbf{U}_\mathbf{q}(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{R}$$

q principal eigenvectors/values of $\mathbf{S}$

arbitrary rotation matrix

Probabilistic PCA: Solution

- One can also compute a MLE for the **noise variance**

$$\hat{\sigma}^2 = \frac{1}{m-q} \sum_{r=q+1}^{m} \lambda_r$$

- Simple interpretation: **lost variance** averaged over dimensions

Probabilistic PCA: Discussion

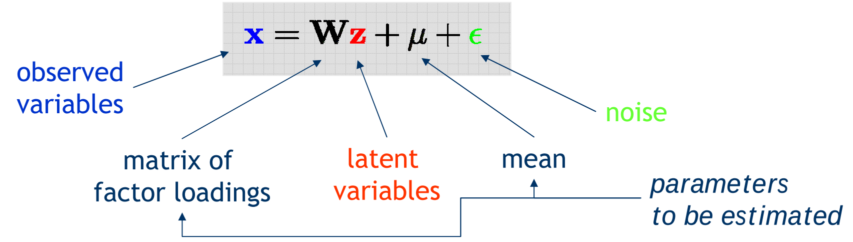
- Advantages of Probabilistic PCA
  - True **generative model** of the data
  - Ability to deal with **missing values** in a principled way
  - Combination with other statistical modeling techniques, e.g. mixture models = **mixture of PCA**
  - Standard **model selection** methods for computing optimal number of retained PCs
  - Extension to **Bayesian PCA**

2.2

Maximum Likelihood Factor Analysis

Factor Analysis

- **Generative probabilistic** model (density model)



$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \epsilon$$

observed variables

matrix of factor loadings

latent variables

mean

noise

parameters to be estimated

- Distributional assumptions (normality)

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Phi}), \; \mathbf{\Phi} = \mathrm{diag}(\phi_1, \ldots, \phi_m)$$

- Induced distribution on observables

$$\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{W}\mathbf{W}^T + \mathbf{\Phi}), \; \mathbf{x} \in \mathbb{R}^m$$

only difference to probabilistic PCA

B. Rubin and D. T. Thayer, *EM algorithms for ML factor analysis*, Psychometrika, vol. 47, no. 1, pp. 69--76, 1982.

Factor Analysis and PPCA

- PPCA is a constrained factor analysis model

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\phi_1, \ldots, \phi_m)) \quad \text{vs.} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

$$\phi_r = \sigma^2, \; \forall r$$

- **Major difference**: Factor analysis models variance of observed variables separately (via Φ), identified factors explain **co-variance** structure
- Other difference:
  - computationally more involved (EM algorithm or quasi-Newton)
  - no nested structure of factors
  - original axis matter in factor analysis, scaling is unimportant

---

2.3

Canonical Correlation Analysis

Canonical Correlation Analysis

- **Canonical correlation analysis:** finding basis vectors for two sets of variables such that the **correlation** between the **projections** of the variables onto these basis vectors are **mutually maximised**

$$\mathbf{x} \in \mathbb{R}^p \qquad\qquad \mathbf{y} \in \mathbb{R}^q$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

$$z_x \equiv \langle \mathbf{w}_x, \mathbf{x} \rangle \qquad\qquad z_y \equiv \langle \mathbf{w}_y, \mathbf{y} \rangle$$

maximize correlation

$$\rho = \frac{\mathbf{w}_x' \mathbf{S}_{xy} \mathbf{w}_y}{\sqrt{(\mathbf{w}_x' \mathbf{S}_{xx} \mathbf{w}_x)(\mathbf{w}_y' \mathbf{S}_{yy} \mathbf{w}_y)}} \quad \Rightarrow$$

(generalized)
eigenvalue problem

2.4

Independent Component Analysis

Non-Gaussian PCA

- Probabilistic model $\mathbf{x} = \mathbf{W}g(\mathbf{z}) + \epsilon$

componentwise nonlinearity

$\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

- Latent variables $g(\mathbf{z})$: **non-Gaussian** prior distribution
- **Independence** of $\mathbf{z}$-components is preserved by componentwise non-linearity
- Classical **Independent Component Analysis** (ICA)

$\mathbf{W} \in \mathbb{R}^{m \times m}, \ \mathrm{rank}(\mathbf{W}) = m$ *invertible case*

$\epsilon \sim \lim_{\sigma \to 0} \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ *noise free case*

A. J. Bell and T. J. Sejnowski, *An information-maximisation approach to blind separation and blind deconvolution*, Neural Computation, 7(6), 1995.

ICA & Blind Source Separation

- ICA is a method to solve the **Blind Source Separation** (BSS) problem

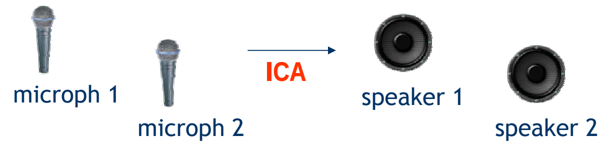
$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)'$$

observed mixed signal

mixing matrix

m independent source components

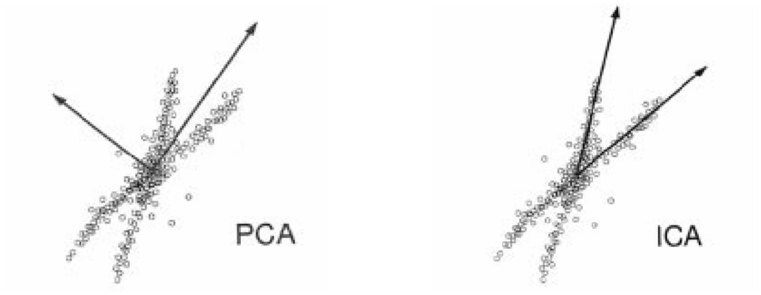
- BSS = "cocktail party problem"
  - m microphones and m speakers
  - Each microphone measures a linear supposition of signals
  - Goal: recover original signals (voices)



(courtesy of Tee-Won Lee)

PCA vs ICA: Example

- Simple synthetic data example:



- PCA aims at **de-correlating** observables (second order statistics)
- ICA aims at **independence** (including higher order moments)

Maximizing Non-Gaussianity

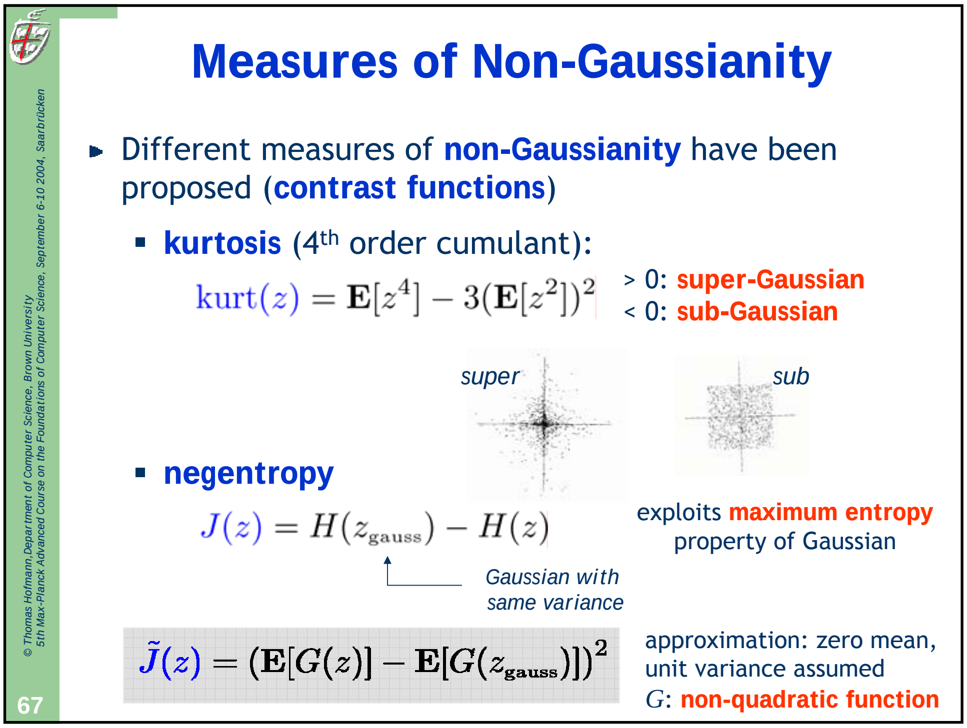
- Linearly mixing independent random variables makes them “more Gaussian” (Central Limit Theorem)
- Linear combination:

$$\langle \mathbf{v}, \mathbf{x} \rangle = \langle \mathbf{v}, \mathbf{W}\mathbf{s} \rangle = \langle \mathbf{W}'\mathbf{v}, \mathbf{s} \rangle = \langle \mathbf{z}, \mathbf{s} \rangle, \; \mathbf{z} = \mathbf{W}'\mathbf{v}$$

combination weights for observables

induced combination weights for independent components

- Find combination weights that make combination appear “**as non-Gaussian as possible**”
- Will recover one of the independent components (up to scale and sign)



Measures of Non-Gaussianity

- Different measures of **non-Gaussianity** have been proposed (**contrast functions**)
  - **kurtosis** (4th order cumulant):

$$\mathrm{kurt}(z) = \mathbf{E}[z^4] - 3(\mathbf{E}[z^2])^2$$

> 0: **super-Gaussian**
< 0: **sub-Gaussian**

  - **negentropy**

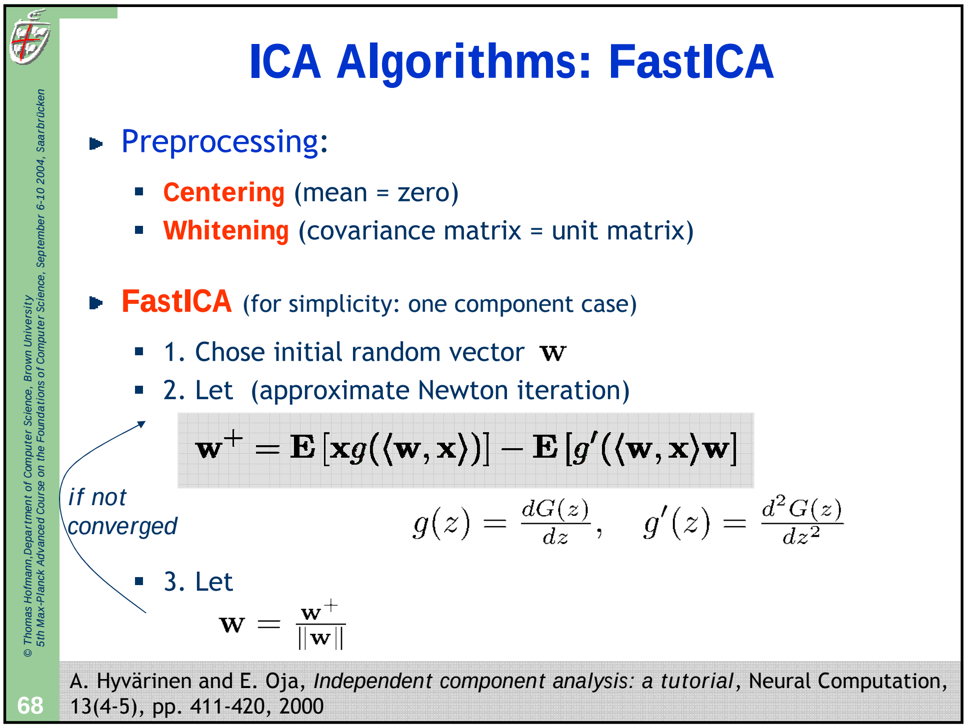
$$J(z) = H(z_{\mathrm{gauss}}) - H(z)$$

Gaussian with same variance

exploits **maximum entropy** property of Gaussian

$$\tilde{J}(z) = (\mathbf{E}[G(z)] - \mathbf{E}[G(z_{\mathbf{gauss}})])^2$$

approximation: zero mean, unit variance assumed
G: **non-quadratic function**



ICA Algorithms: FastICA

- Preprocessing:
  - **Centering** (mean = zero)
  - **Whitening** (covariance matrix = unit matrix)
- **FastICA** (for simplicity: one component case)
  - 1. Chose initial random vector $\mathbf{w}$
  - 2. Let (approximate Newton iteration)

$$\mathbf{w}^+ = \mathbf{E}\left[\mathbf{x}g(\langle\mathbf{w},\mathbf{x}\rangle)\right] - \mathbf{E}\left[g'(\langle\mathbf{w},\mathbf{x}\rangle\mathbf{w}\right]$$

if not converged

$$g(z) = \frac{dG(z)}{dz}, \quad g'(z) = \frac{d^2G(z)}{dz^2}$$

  - 3. Let

$$\mathbf{w} = \frac{\mathbf{w}^+}{||\mathbf{w}||}$$

A. Hyvärinen and E. Oja, *Independent component analysis: a tutorial*, Neural Computation, 13(4-5), pp. 411-420, 2000

Maximum Likelihood ICA

- ICA algorithms can also be based on **maximizing the likelihood** of the generative non-Gaussian factor model
- Noisefree, invertible case:

$$p(\mathbf{x};\mathbf{W}) = \int \overbrace{\prod_j \delta(x_j - \sum_k w_{jk} s_{ik})}^{\mathbf{x} = \mathbf{W}\mathbf{s}} \overbrace{\prod_j p_j(s_j)}^{\text{independent sources}} d\mathbf{s}$$

change of variables

$$\log p(\mathbf{x};\mathbf{W}) = -\log|\mathbf{W}| + \sum_j \log p_j\left(\sum_i w_{ij}^{-1} x_i\right)$$

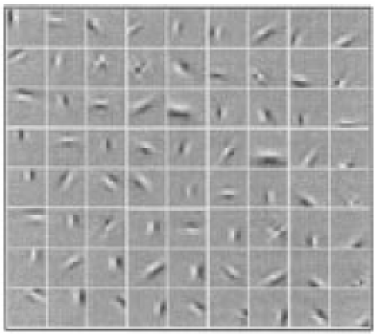
- **Log-likelihood**

$$\mathcal{L}(\mathbf{W}) = -n \log|\mathbf{W}| + \sum_i \sum_j \log p_j\left(\sum_k w_{kj}^{-1} x_{ik}\right)$$

  - optimized with gradient descent procedure

Application of ICA to Images

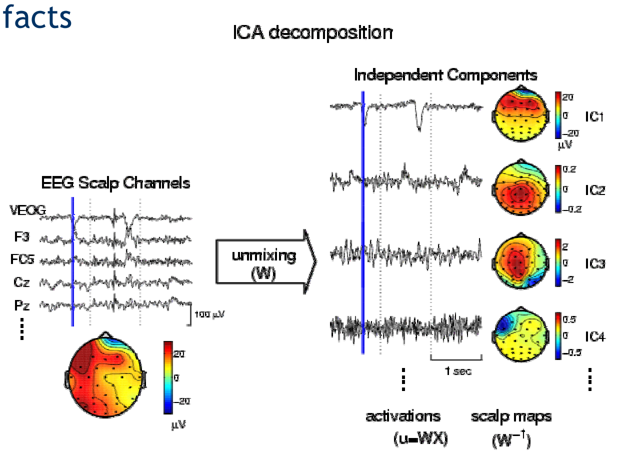
- ICA on **patches of 12-by-12 pixels** from pictures of natural scenes.
- Components are similar to **Gabor filters** (oriented edge detectors)



T.-P. Jung, S. Makeig, M. J. McKeown, A.J.. Bell, T.-W. Lee, and T. J. Sejnowski, *Imaging Brain Dynamics Using Independent Component Analysis,* Proceedings of the IEEE, 89(7):1107-22, 2001.

Application of ICA to Brain Signals

- Unmixing multichannel EEG recordings, e.g. to remove artifacts

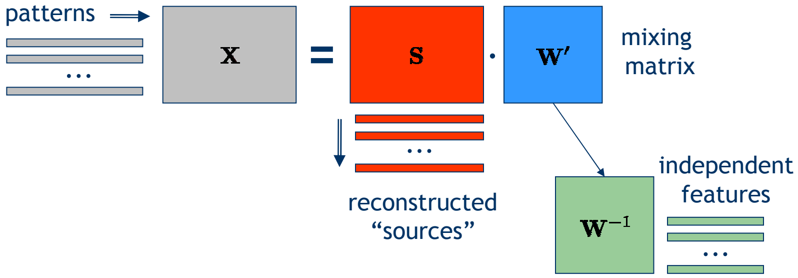


T.-P. Jung, S. Makeig, M. J. McKeown, A.J.. Bell, T.-W. Lee, and T. J. Sejnowski, *Imaging Brain Dynamics Using Independent Component Analysis,* Proceedings of the IEEE, 89(7):1107-22, 2001.

ICA & Matrix Decomposition

- ICA equations can be written in matrix notation (noise-free, invertible case), patterns in rows

$$\mathbf{X} = \mathbf{S}\mathbf{W}'$$



3.

Latent Semantic Analysis

3.1

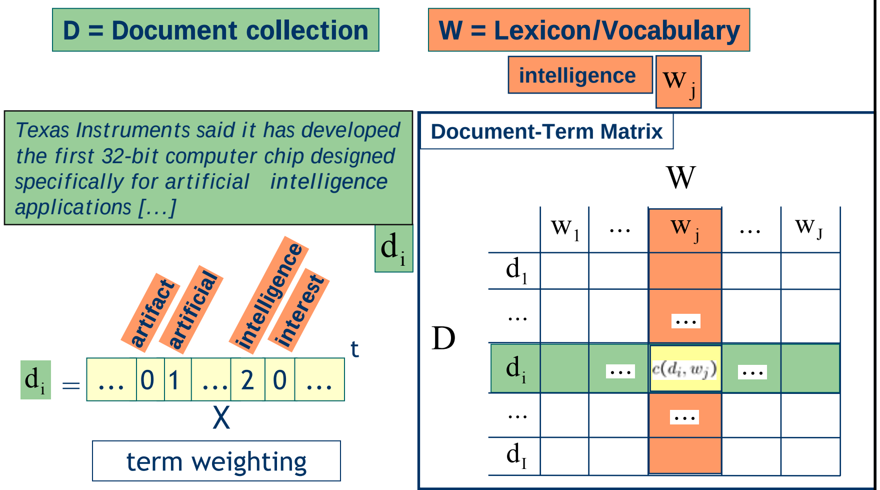
Background: Information Retrieval

Ad Hoc Retrieval

- Search a document collection to find the ones that satisfy an **immediate information need**
- Information need is expressed in terms of a **query**
- *Magic*: Identify relevant documents based on short, ambiguous, and incomplete query

- By far the most popular form of information access:
  - 85% of American Internet users have ever used an online search engine to find information on the Web (Fox, 2002)
  - 29% of Internet users rely on a search engine on a typical day (Fox, 2002)

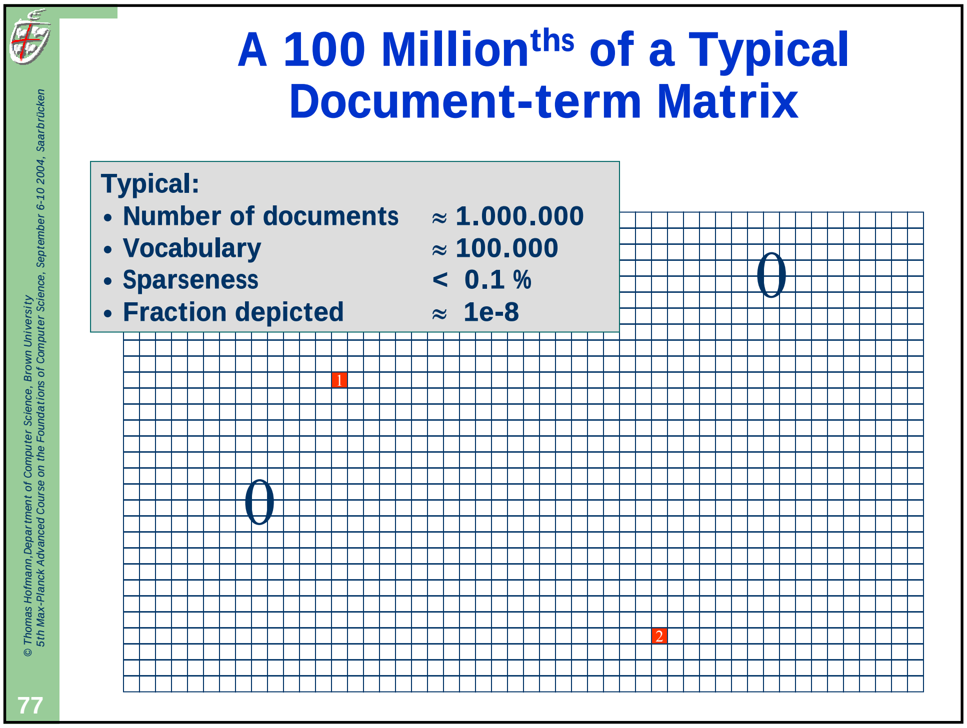
Document-Term Matrix



D = Document collection

W = Lexicon/Vocabulary

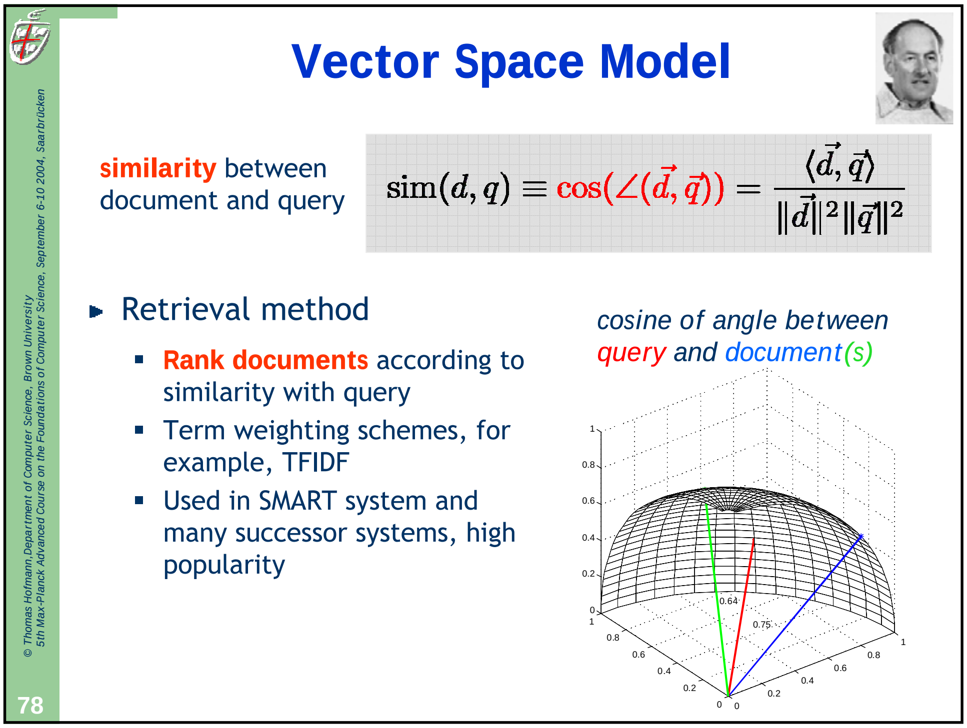
Texas Instruments said it has developed the first 32-bit computer chip designed specifically for artificial intelligence applications [...]



A 100 Millionths of a Typical Document-term Matrix

Typical:

• **Number of documents**	**≈ 1.000.000**
• **Vocabulary**	**≈ 100.000**
• **Sparseness**	**< 0.1 %**
• **Fraction depicted**	**≈ 1e-8**



Vector Space Model

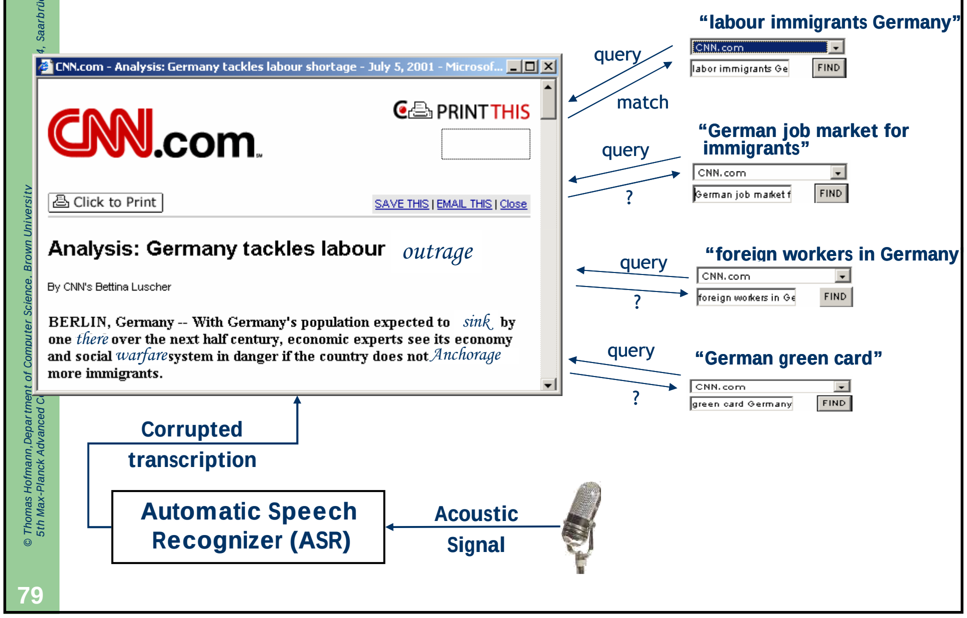
similarity between document and query

$$\mathrm{sim}(d,q) \equiv \cos(\angle(\vec{d},\vec{q})) = \frac{\langle \vec{d},\vec{q} \rangle}{\|\vec{d}\|^2 \|\vec{q}\|^2}$$

- Retrieval method
  - **Rank documents** according to similarity with query
  - Term weighting schemes, for example, TFIDF
  - Used in SMART system and many successor systems, high popularity

cosine of angle between query and document(s)

Vocabulary Mismatch & ~~Robustness~~



CNN.com - Analysis: Germany tackles labour shortage - July 5, 2001 - Microsof...

Analysis: Germany tackles labour *outrage*

By CNN's Bettina Luscher

BERLIN, Germany -- With Germany's population expected to *sink* by one *there* over the next half century, economic experts see its economy and social *warfare*system in danger if the country does not *Anchorage* more immigrants.

Vector Space Model: Pros

- **Automatic** selection of index terms
- **Partial matching** of queries and documents *(dealing with the case where no document contains all search terms)*
- **Ranking** according to **similarity score** *(dealing with large result sets)*
- **Term weighting** schemes *(improves retrieval performance)*
- Various extensions
  - Document clustering
  - Relevance feedback (modifying query vector)
- Geometric foundation

Problems with Lexical Semantics

- Ambiguity and association in natural language
  - **Polysemy**: Words often have a **multitude of meanings** and different types of usage *(more urgent for very heterogeneous collections).*
  - The vector space model is unable to discriminate between different meanings of the same word.

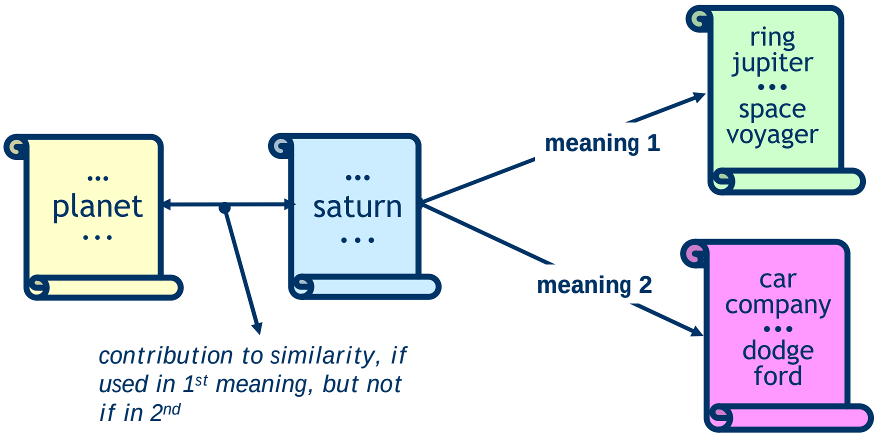
$$\mathrm{sim}_{\mathrm{true}}(d,q) < \cos(\angle(\vec{d},\vec{q}))$$

  - **Synonymy**: Different terms may have an **identical or a similar meaning** (weaker: words indicating the same topic).
  - No associations between words are made in the vector space representation.

$$\mathrm{sim}_{\mathrm{true}}(d,q) > \cos(\angle(\vec{d},\vec{q}))$$

Polysemy and Context

- Document similarity on single word level: polysemy and context



ring
jupiter
•••
space
voyager

meaning 1

…
planet
…

…
saturn
…

meaning 2

car
company
•••
dodge
ford

contribution to similarity, if used in 1st meaning, but not if in 2nd

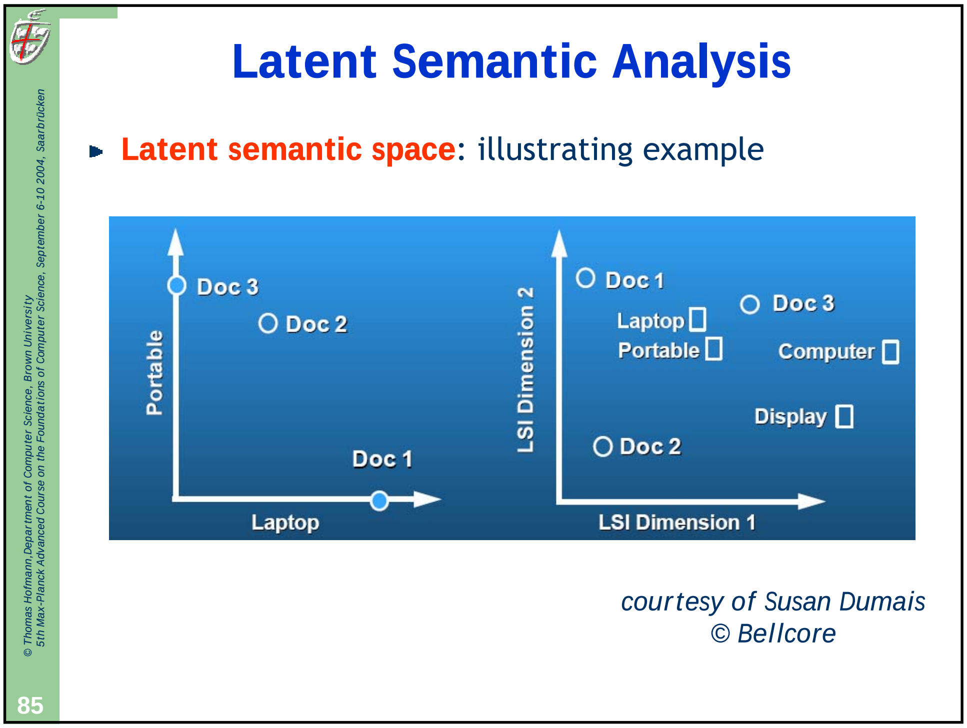
3.2

Latent Semantic Analysis via SVD

Latent Semantic Analysis

- Perform a **low-rank approximation** of **document-term matrix** (typical rank **100-300**)
- General idea
  - Map documents (and terms) to a **low-dimensional** representation.
  - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
  - Compute document similarity based on the **inner product** in the **latent semantic space**
- Goals
  - Similar terms map to similar location in low dimensional space
  - Noise reduction by dimension reduction

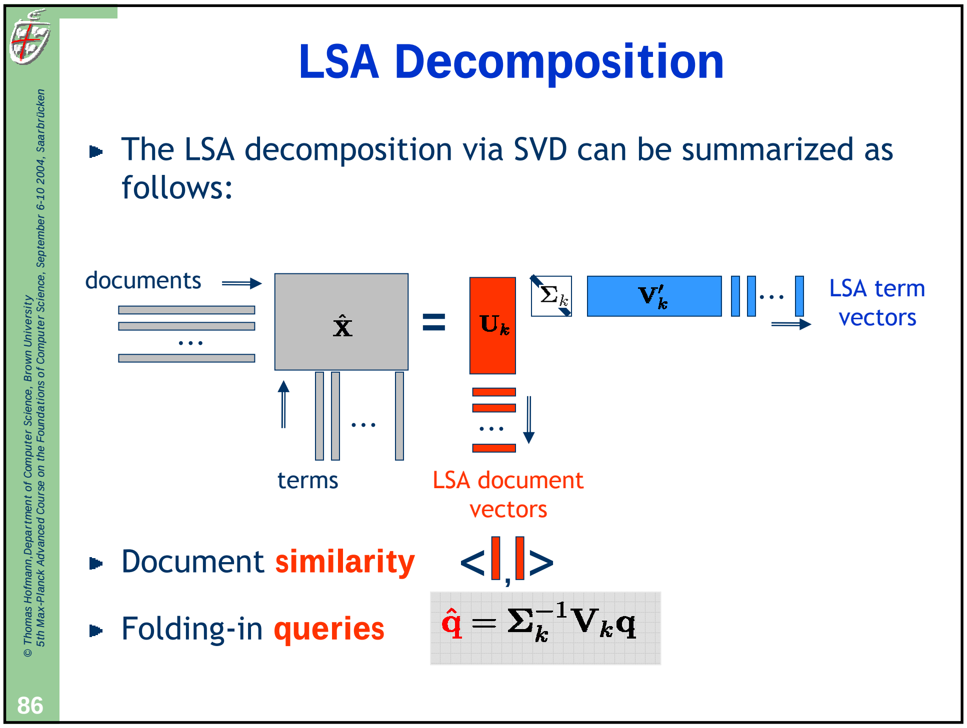
M. Berry, S. Dumais, and G. O'Brien. *Using linear algebra for intelligent information retrieval*. SIAM Review, 37(4):573--595, 1995.



Latent Semantic Analysis

- **Latent semantic space**: illustrating example

courtesy of Susan Dumais
© Bellcore



LSA Decomposition

- The LSA decomposition via SVD can be summarized as follows:

- Document **similarity** <▮,▮>
- Folding-in **queries** $\hat{\mathbf{q}} = \Sigma_k^{-1} \mathbf{V}_k \mathbf{q}$

3.3

Probabilistic Latent Semantic Analysis

Search as Statistical Inference

- Document in bag-of-words representation

China US trade relations

Search

$$\mathbf{P}(\text{'China'}|\text{all other words})$$

$$\mathbf{P}(\text{'trade'}|\text{all other words})$$

How probable is it that terms like "China" or "trade" might occur?

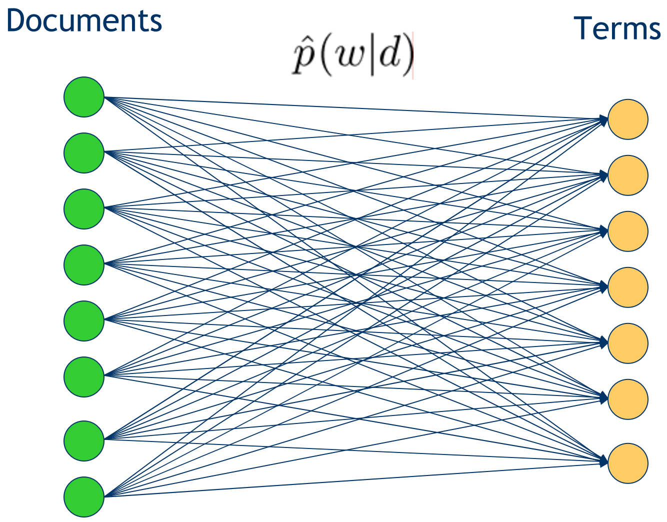
Additional index terms can be added automatically via statistical inference!

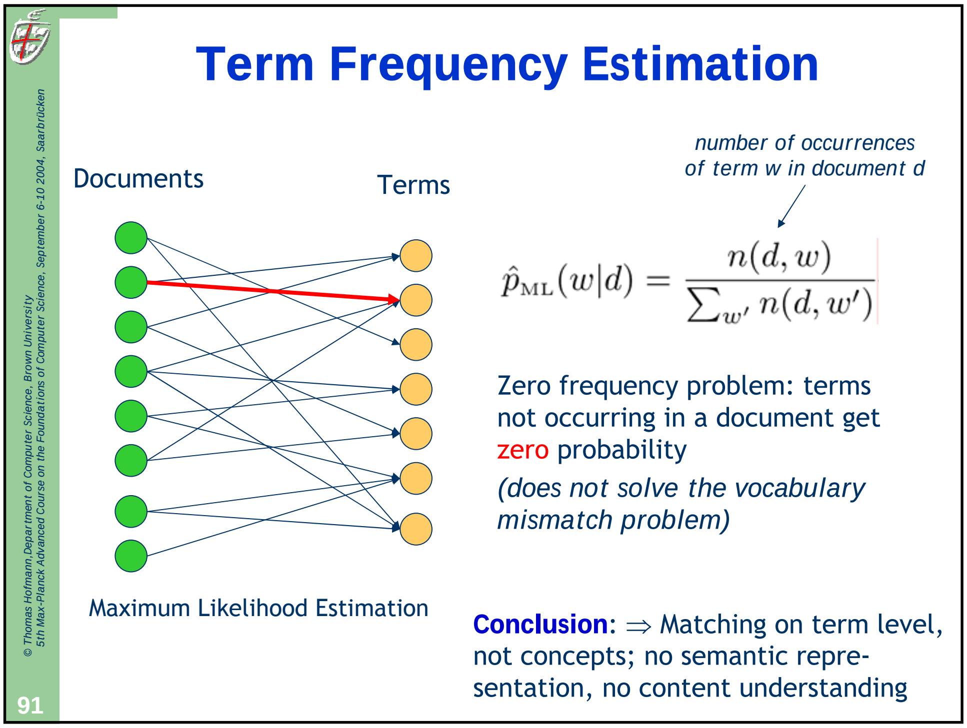
Probabilistic Latent Semantic Analysis

- Concept-based information retrieval: matching based on concepts, not terms/words
  - *E.g. terms like 'Bejing', 'China', 'chinese', or 'Hong Kong' refer to the concept 'CHINA'*
  - *E.g. terms like 'economic' or 'imports' refer to the concept 'TRADE'*
- Design goals of **pLSA**:
  - **Statistical** technique to extract concepts (vs. traditional: utilization of thesauri, semantic networks, ontologies = high manual costs, limited adaptivity)
  - **Domain-specific** extraction of concepts based on given document collection
  - **Quantitative model** for word prediction in documents (concept-based language model)

T. Hofmann. *Probabilistic latent semantic indexing*. In Proceedings 22nd ACM SIGIR, 1999.

Estimation Problem





Term Frequency Estimation

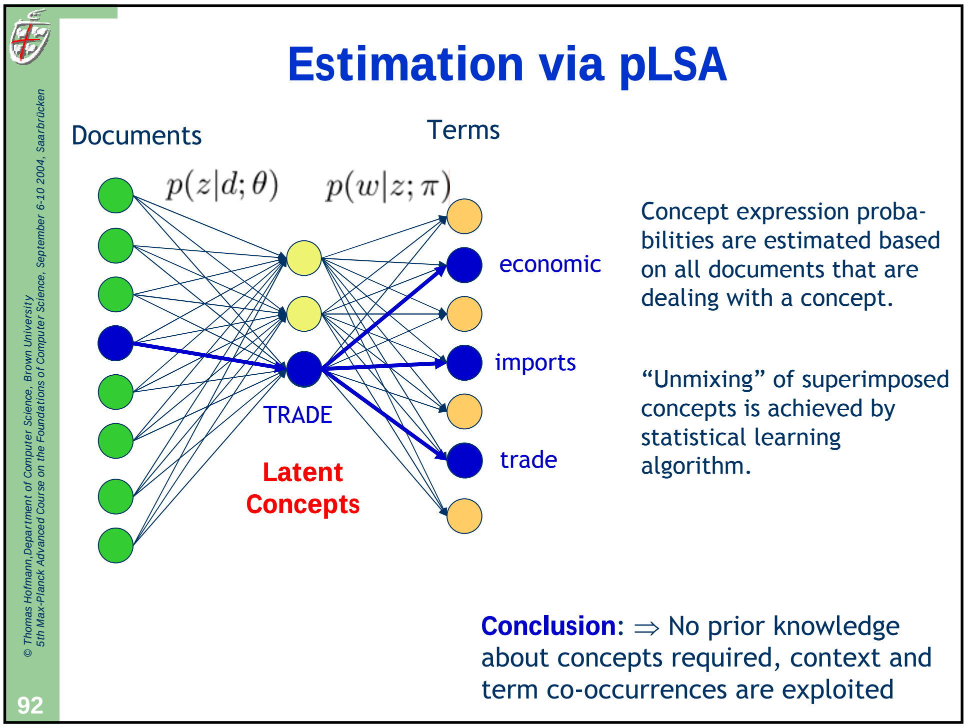
number of occurrences of term w in document d

$$\hat{p}_{\mathrm{ML}}(w|d) = \frac{n(d,w)}{\sum_{w'} n(d,w')}$$

Zero frequency problem: terms not occurring in a document get zero probability

(does not solve the vocabulary mismatch problem)

Conclusion: ⇒ Matching on term level, not concepts; no semantic representation, no content understanding



Estimation via pLSA

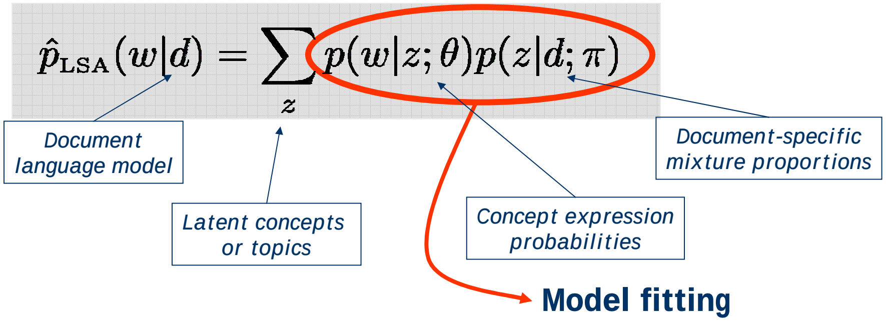
Concept expression probabilities are estimated based on all documents that are dealing with a concept.

“Unmixing” of superimposed concepts is achieved by statistical learning algorithm.

Conclusion: ⇒ No prior knowledge about concepts required, context and term co-occurrences are exploited

pLSA – Latent Variable Model

- Structural modeling assumption (mixture model)



$$\hat{p}_{\text{LSA}}(w|d) = \sum_z p(w|z;\theta)p(z|d;\pi)$$

Document language model

Latent concepts or topics

Concept expression probabilities

Document-specific mixture proportions

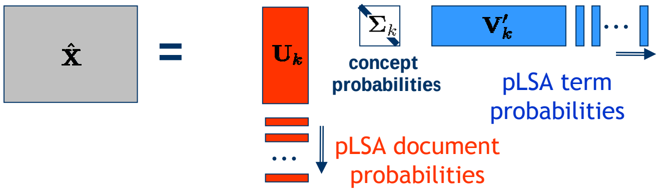
Model fitting

---

pLSA: Matrix Decomposition

- Mixture model can be written as a **matrix factorization**
- Equivalent symmetric (joint) model

$$\hat{p}_{\text{LSA}}(d,w) = \sum_z p(d|z)\,p(z)\,p(w|z)$$



concept probabilities

pLSA term probabilities

pLSA document probabilities

- Contrast to LSA/SVD: **non-negativity** and **normalization**
  (intimate relation to non-negative matrix factorization)

pLSA via Likelihood Maximization

- **Log-Likelihood**

$$L(\theta, \pi; c) = \sum_{d,w} \underbrace{c(d,w)}_{\text{Observed word frequencies}} \log \underbrace{\left[\sum_{z} p(w|z;\theta) p(z|d;\pi) \right]}_{\hat{p}_{\text{LSA}}(w|d)}$$

argmax

$$(\hat{\theta}, \hat{\pi})$$

Predictive probability of pLSA mixture model

- **Goal**: Find model parameters that maximize the log-likelihood, i.e. maximize the average predictive probability for observed word occurrences (**non-convex problem**)

Expectation Maximization Algorithm

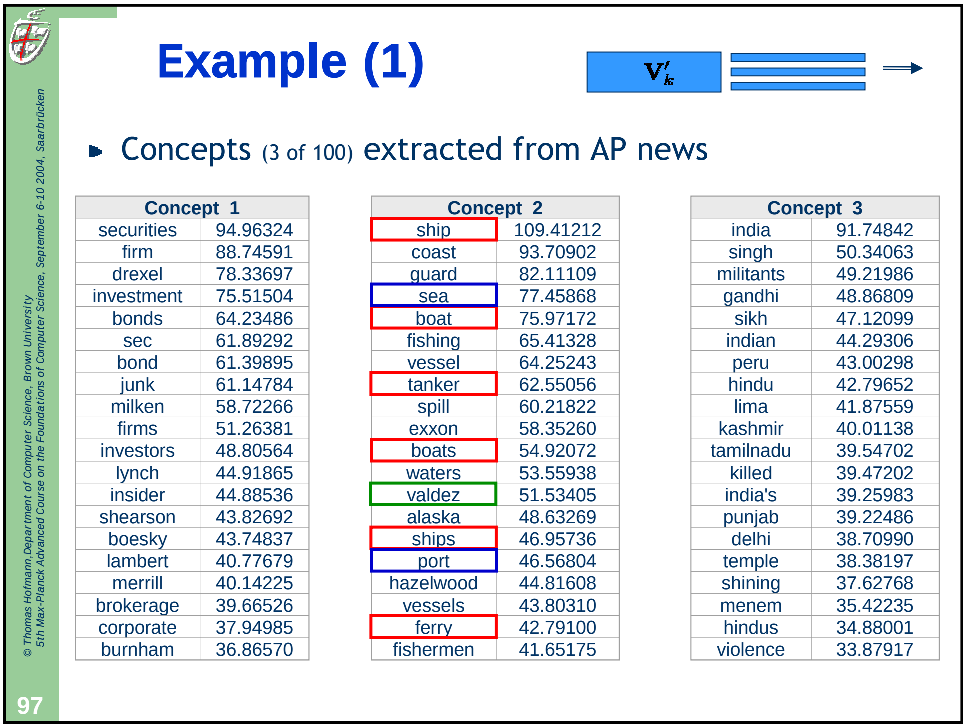
- **E step**: posterior probability of latent variables ("concepts")

$$p(z|d,w) = \frac{p(z|d;\pi)p(w|z;\theta)}{\sum_{z'} p(z'|d;\pi)p(w|z';\theta)}$$

Probability that the occurence of term w in document d can be "explained" by concept z

- **M step**: parameter estimation based on "completed" statistics

$$p(w|z;\theta) \propto \underbrace{\sum_{d} c(d,w)p(z|d,w)}_{\text{how often is term } w \text{ associated with concept } z\text{ ?}}, \quad p(z|d;\pi) \propto \underbrace{\sum_{w} c(d,w)p(z|d,w)}_{\text{how often is document } d \text{ associated with concept } z\text{ ?}}$$

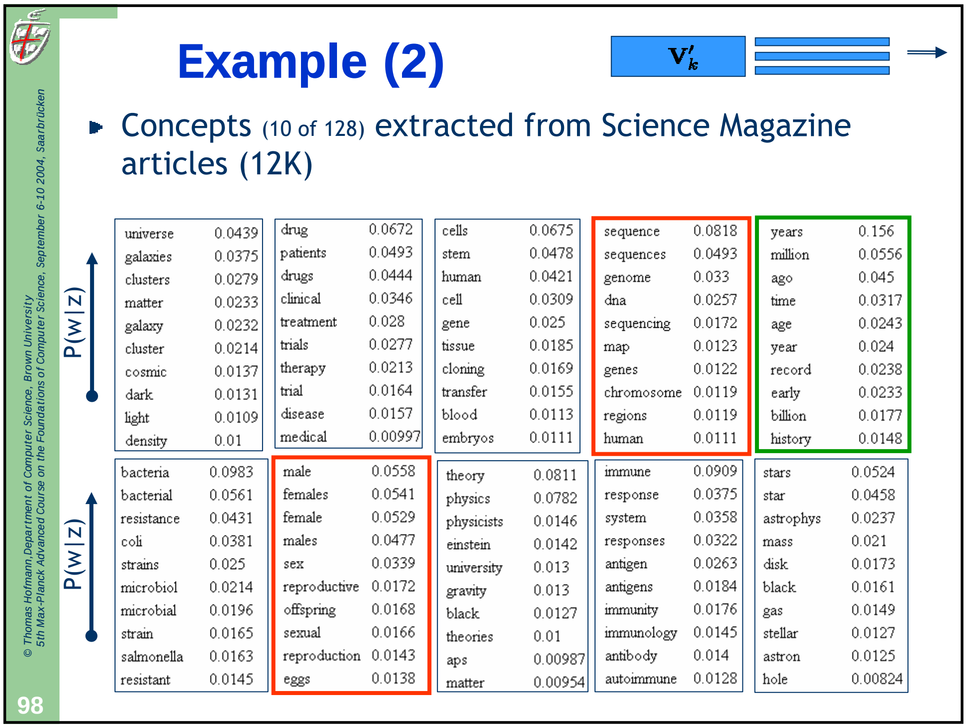


Example (1)

$\mathbf{V}'_k$

- Concepts (3 of 100) extracted from AP news

Concept 1		Concept 2		Concept 3	
securities	94.96324	ship	109.41212	india	91.74842
firm	88.74591	coast	93.70902	singh	50.34063
drexel	78.33697	guard	82.11109	militants	49.21986
investment	75.51504	sea	77.45868	gandhi	48.86809
bonds	64.23486	boat	75.97172	sikh	47.12099
sec	61.89292	fishing	65.41328	indian	44.29306
bond	61.39895	vessel	64.25243	peru	43.00298
junk	61.14784	tanker	62.55056	hindu	42.79652
milken	58.72266	spill	60.21822	lima	41.87559
firms	51.26381	exxon	58.35260	kashmir	40.01138
investors	48.80564	boats	54.92072	tamilnadu	39.54702
lynch	44.91865	waters	53.55938	killed	39.47202
insider	44.88536	valdez	51.53405	india's	39.25983
shearson	43.82692	alaska	48.63269	punjab	39.22486
boesky	43.74837	ships	46.95736	delhi	38.70990
lambert	40.77679	port	46.56804	temple	38.38197
merrill	40.14225	hazelwood	44.81608	shining	37.62768
brokerage	39.66526	vessels	43.80310	menem	35.42235
corporate	37.94985	ferry	42.79100	hindus	34.88001
burnham	36.86570	fishermen	41.65175	violence	33.87917



Example (2)

$\mathbf{V}'_k$

- Concepts (10 of 128) extracted from Science Magazine articles (12K)

P(w|z)

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148

P(w|z)

bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

Live Implementation

Skip navigation

MedlinePlus®
Trusted Health Information for You

A service of the U.S. NATIONAL LIBRARY OF MEDICINE
and the NATIONAL INSTITUTES OF HEALTH

Search MedlinePlus

About MedlinePlus | Site Map | FAQs | Contact Us

español

Health Topics
Start here with over 650 topics on conditions, diseases and wellness

Drug Information
About your prescription and over-the-counter medicines

Medical Encyclopedia
Includes pictures and diagrams

Dictionary
Spellings and definitions of medical words

News
Health News from the past 30 days

Directories
Find doctors, dentists and hospitals

Current Health News
- Radiotherapy Not Needed for Some Breast Cancers
- Soy, Fish Oil May Protect Against Alzheimer's Disease
- Study Finds Immune Therapy for Metastatic Breast Cancer Possible
- More news

Featured Site

September is National Cholesterol Education Month from the National Heart, Lung, and Blood Institute

In the Spotlight

Interactive Tutorials
Over 165 slideshows with sound and pictures

ClinicalTrials.gov
Studies for new drugs and treatments

NIHSeniorHealth
Health information for older adults

NIHSeniorHealth: Health information f

What's new on MedlinePlus?
Sign up now!

http://www.nlm.nih.gov/cgi/medlineplus/leavemedplus.pl?theURL=http://nihseniorhealth.gov/

Internet

3.4

Latent Dirichlet Allocation

Hierarchical Bayesian Model

- **Latent Dirichlet Allocation** (LDA) defines a generative model (for documents) in the following way
  - 1. Choose document length

    $N \sim \text{Poisson}(\xi)$
  - 2. Choose **topic distribution**

    $\theta \sim \text{Dirichlet}(\alpha)$
  - 3. For each of the N words
    - choose a topic $z_i \sim \text{Multinomial}(\theta)$
    - generate a word $w_i \sim P(\cdot|z_i, \beta)$

pLSA

$p(z|d)$

$p(w|z)$

Latent Dirichlet Allocation

- **Joint probabilistic model**

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{i=1}^{n} p(z_i|\theta) p(w_i|z_i; \beta)$$

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1^k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1}$$

Dirichlet density

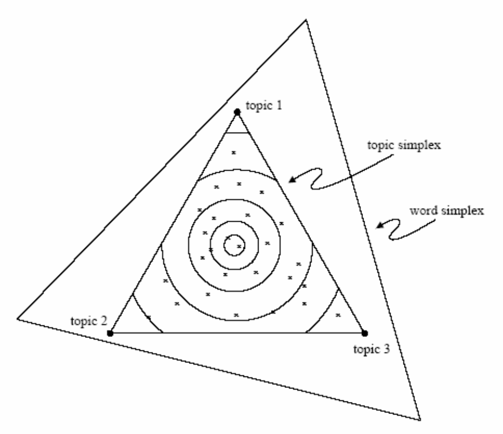
- **Marginal distribution** of a document

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \prod_{i=1}^{n} \sum_{z_i} p(z_i|\theta) p(w_i|z_i; \beta)\, d\theta$$

D. M. Blei and A. Y. Ng and M. I. Jordan, *Latent dirichlet allocation*, J. Mach. Learn. Res., vol 3, 993–1022, 2003.

Topic and Word Simplex

- The geometry of the LDA model (and the pLSA model) can be sketched as follows:



courtesy of David Blei

Variational Approximation

- Computing the marginal distribution is **intractable**, hence exact Maximum Likelihood Estimation is not possible
- Instead: **Convex variational approximation**
- Introduce factorizing variational distribution (**parametrized**)

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{i=1}^{n} q(z_i|\phi_i) \longrightarrow$$

neglects direct couplings between θ, **w**, **z**

- **Variational EM** algorithm: optimize variational parameters and model parameters in an alternating fashion (details beyond the scope of this tutorial)

3.5

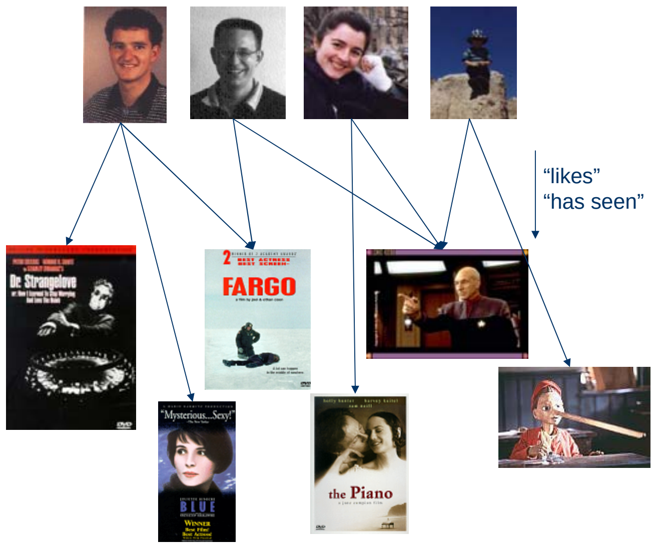
Recommender Systems

Personalized Information Filtering:

Users/ Customers

Judgement/ Selection

Objects



"likes"
"has seen"

Predicting Preferences and Actions

User Profile

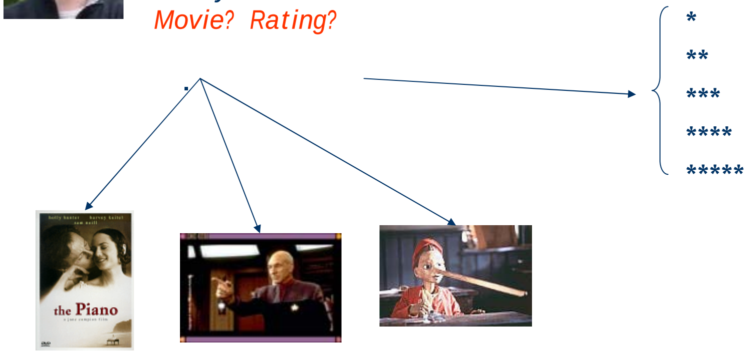
*Dr. Strangeloves ******

*Three Colors: Blue *****

*O Brother, Where Art Thou? ******

*Pretty Woman **

Movie? Rating?



*

**

***

****

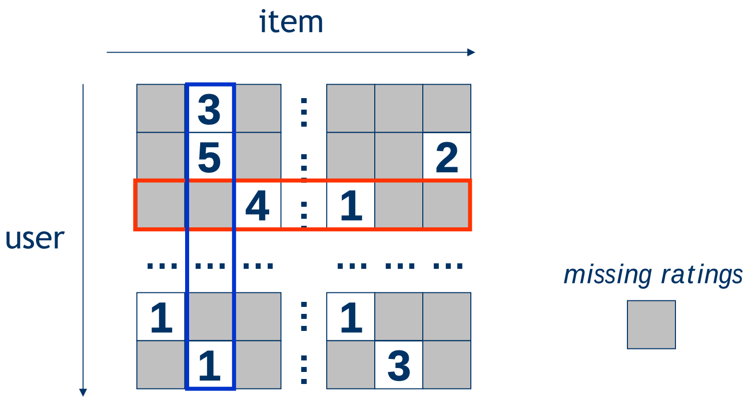
*****

Collaborative & Content-Based Filtering

- **Collaborative/social filtering**
  - Properties of persons or **similarities between persons** are used to improve predictions.
  - Makes use of user profile data
  - Formally: starting point is **sparse matrix with user ratings**
- **Content-based filtering**
  - Properties of objects or similarities between objects are used to improve predictions
  - Problem: predictive attributes may not exist

Rating Matrix

- Rating matrix is typically a large matrix with many (mostly) **missing values**



---

pLSA-like Decomposition

- Generalization of pLSA (additional rating variable)

$$p_{\mathrm{LSA}}(r,y|u) = \sum_z \underbrace{p(r|y,z;\rho)}_{\text{extension to predict ratings}}\underbrace{p(y|z;\theta)p(z|u;\pi)}_{\text{standard pLSA model to explain sparseness pattern}}$$

  - Explicit decomposition of user preferences (each user can have **multiple interests**)
  - Probabilistic model can be used to optimize specific **objectives**
  - Data **compression** and **privacy** preservation
- Details
  - multinomial or Gaussian sampling model for rating variable
  - EM algorithm for (approximate) model fitting

T. Hofmann, *Latent Semantic Models for Collaborative Filtering*, ACM Transactions on Information Systems, 2004, Vol 22(1), pp. 89-115.

Example

- **High** rating factors:

Interest Group 1, *4.8*	Interest Group 2, *4.6*	Interest Group 3, *4.5*	Interest Group 4, *4.4*	Interest Group 5, *4.4*
Twister	Batman (1989)	Trainspotting	Dead Man Walking	The Santa Clause
[*4.6*] [0.064]	[*4.1*] [0.066]	[*5*] [0.038]	[*5*] [0.052]	[*4.5*] [0.014]
Independence Day (...	Apollo 13	Fargo	The Truth about Ca...	Casper
[*4.9*] [0.061]	[*5*] [0.065]	[*5*] [0.033]	[*4.3*] [0.039]	[*4.5*] [0.014]
Toy Story	True Lies	Pulp Fiction	Get Shorty	Robin Hood: Men in...
[*4.9*] [0.057]	[*4.7*] [0.059]	[*5*] [0.028]	[*4.6*] [0.036]	[*4.3*] [0.013]
Broken Arrow	Batman Forever	Clerks	Sense and Sensibil...	Tommy Boy
[*4.4*] [0.054]	[*4.1*] [0.054]	[*4.7*] [0.023]	[*5*] [0.035]	[*4.5*] [0.013]
Interest Group 6, *4.3*	Interest Group 7, *4.3*	Interest Group 8, *4.2*	Interest Group 9, *4*	Interest Group 10, *3.9*
The Remains of the...	The Empire Strikes...	Pretty Woman	Sleepers	A Clockwork Orange...
[*4.5*] [0.047]	[*4.7*] [0.032]	[*4.3*] [0.059]	[*4.2*] [0.015]	[*4.2*] [0.01]
The Piano	Raiders of the Los...	Mrs. Doubtfire	Jerry Maguire	Amadeus (1984)
[*4.7*] [0.043]	[*4.7*] [0.03]	[*4.3*] [0.059]	[*4.6*] [0.013]	[*4.2*] [0.0098]
Like Water For Cho...	Star Wars	Ghost	The First Wives Cl...	Psycho (1960)
[*4.7*] [0.043]	[*4.9*] [0.026]	[*4.4*] [0.057]	[*3.8*] [0.013]	[*4.3*] [0.0098]
Much Ado About Not...	Indiana Jones and ...	Sleepless in Seatt...	William Shakespear...	One Flew Over the ...
[*4.6*] [0.041]	[*4.5*] [0.025]	[*4.4*] [0.055]	[*4.5*] [0.011]	[*4.5*] [0.0095]

Example

- **Low** rating factors:

Interest Group 31, *2.2*	Interest Group 32, *2*	Interest Group 33, *1.8*	Interest Group 34, *1.8*	Interest Group 35, *1.7*
E.T.: The Extrater...	Lord of Illusions	Sleepless in Seatt...	Toy Story	Striptease
[*2.6*] [0.01]	[*1.6*] [0.011]	[*1.8*] [0.017]	[*2.4*] [0.05]	[*0.025*] [0.033]
The Sound of Music...	Tales From the Hoo...	The Firm	Mission: Impossibl...	Independence Day (...
[*2.3*] [0.0086]	[*1.6*] [0.0087]	[*1.8*] [0.015]	[*1.8*] [0.049]	[*0.87*] [0.029]
Top Gun (1986)	Mallrats	Pretty Woman	Independence Day (...	The Cable Guy
[*2.3*] [0.0086]	[*2.4*] [0.0083]	[*1.5*] [0.015]	[*2.1*] [0.048]	[*0.16*] [0.028]
Mary Poppins (1964...	Wes Craven's New N...	Dave	Twister	Barb Wire
[*2.3*] [0.0083]	[*2*] [0.0082]	[*2*] [0.015]	[*1.8*] [0.043]	[*4.9e-005*] [0.025]
Interest Group 36, *1.1*	Interest Group 37, *0.68*	Interest Group 38, *0.39*	Interest Group 39, *0.16*	Interest Group 40, *0.16*
Super Mario Bros.	Mighty Morphin Pow...	Dumb and Dumber	Kazaam	Tales From the Hoo...
[*0.11*] [0.017]	[*0.017*] [0.033]	[*0.0025*] [0.038]	[*0.028*] [0.014]	[*0.022*] [0.0075]
The Beverly Hillbi...	The Brady Bunch Mo...	Ace Ventura: Pet D...	Children of the Co...	Vampire in Brookly...
[*0.34*] [0.016]	[*0.28*] [0.024]	[*0.016*] [0.034]	[*0.021*] [0.014]	[*1.3e-005*] [0.007]
Richie Rich	Mortal Kombat	Ace Ventura: When ...	A Very Brady Seque...	The Baby-Sitters C...
[*0.22*] [0.015]	[*0.21*] [0.018]	[*0.00067*] [0.033]	[*0.083*] [0.012]	[*0.0063*] [0.007]
The Next Karate Ki...	The Bridges of Mad...	Waterworld	Halloween: The Cur...	Candyman: Farewell...
[*0.21*] [0.014]	[*0.015*] [0.018]	[*0.034*] [0.028]	[*0.035*] [0.012]	[*0.0039*] [0.0065]

SVD-based Modeling Approach

- Model: matrix entries have been omitted **randomly**

unobserved complete ratings

$$r^*_{ij} = \begin{cases} r_{ij} & \text{with probability } p_{ij} \\ ? & \text{with probability } 1 - p_{ij} \end{cases}$$

omitted ratings

- Two step procedure for predicting missing entries

1

$$p_{ij} = \begin{cases} 0 & \text{if } r^*_{ij} = ? \\ 1 & \text{otherwise} \end{cases}$$

$\Downarrow$

rank k approximation (SVD)

$\Downarrow$

$\hat{\mathbf{P}}_k$

2

$$\hat{r}_{ij} = \begin{cases} r^*_{ij}/p_{ij} & \text{if } r^*_{ij} \neq ? \\ 0 & \text{otherwise} \end{cases}$$

$\Downarrow$

rank q approximation (SVD)

$\Downarrow$

$\hat{\mathbf{R}}_q$

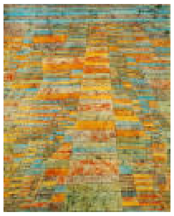
SVD-based Modeling Approach

- Theoretical guarantees for reconstruction accuracy (if omission probabilities are correct)
- Rank of P-approximation:
  - Low rank (e.g. 2): “completely random” omission probabilities
  - High rank: accurate omission model
- Applicable as a more general data mining technique

Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. *Spectral analysis of data*. In Proceedings of the ACM Symposium on Theory of Computing (STOC), 2001

4.

Spectral Clustering & Link Analysis

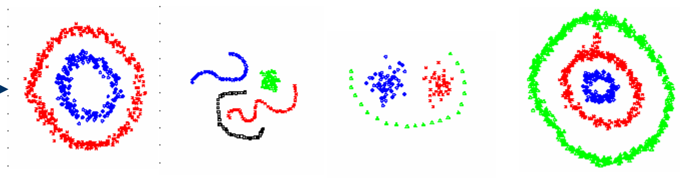


4.1

Spectral Clustering

Data Clustering

- Goal of data clustering is to automatically discover grouping structure (**clusters**)
- Different definition of what a good cluster is exist:
  - compactness (e.g. pairwise distances, distance from center or diameter is small) -> K-means and relatives
  - **denseness** (i.e. clusters are regions of relatively high density)



- Many applications: data mining, document clustering, computer vision, etc.

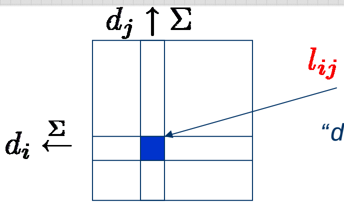
Affinity Matrix

- Assumption: **distance function** (metric) is given
- 1. Compute **affinity matrix**

$$\mathbf{A} \in \mathbb{R}^{n \times n}, \; a_{ij} \equiv \exp\left[-\gamma \|\mathbf{x_i} - \mathbf{x}_j\|^2\right], \; \gamma > 0$$

  - between 0 and 1, exponentially decaying with squared distance
- 2. **Normalization** (differs for different algorithms)

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \; \mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n), \; d_i = \sum_j a_{ij}$$



$$l_{ij} = \frac{a_{ij}}{\sqrt{d_i d_j}}$$

"degree normalization"

Decomposition & Clustering

- 3. Eigen decomposition and **low-rank approximation**

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}' \approx \mathbf{U}_q\mathbf{\Lambda}_q\mathbf{U}_q'$$

- 4. **Row-normalization**

$$\hat{\mathbf{U}} \in \mathbb{R}^{n \times q}, \;\; \hat{u}_{ij} = \frac{u_{ij}}{\sqrt{\sum_k u_{ik}^2}}$$

- 5. **Clustering**: cluster rows of $\hat{\mathbf{U}}$ (e.g. using k-means)

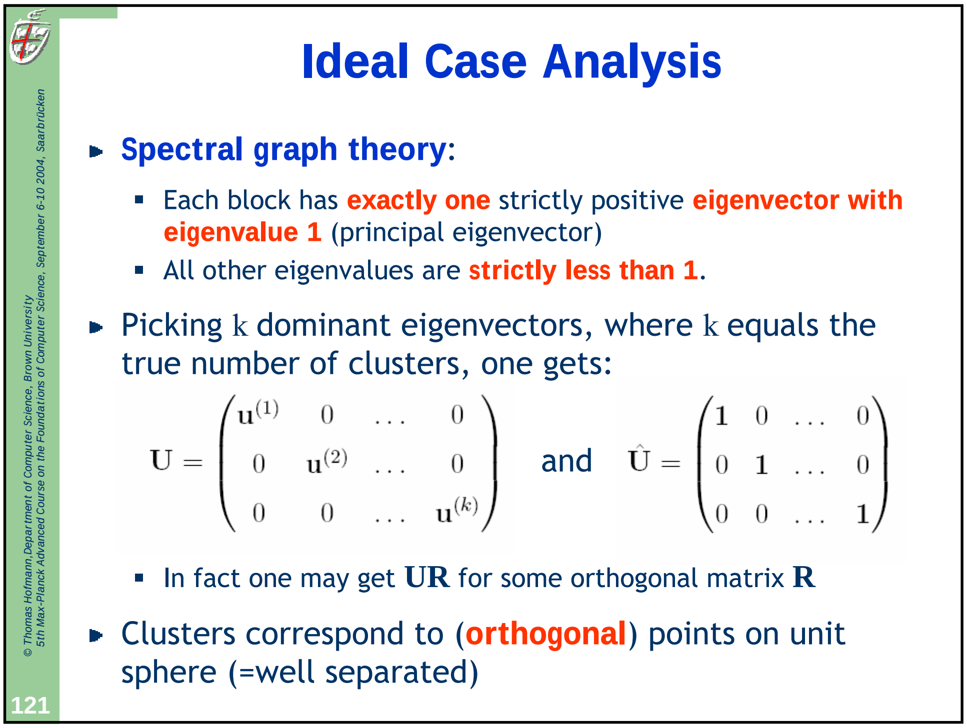
Ideal Case Analysis

- Ideal case: **perfectly separated** cluster, i.e. $a_{ij} = 0$ for data points in different clusters
- **Block diagonal** (normalized) affinity matrix

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}^{(1)} & 0 & \dots & 0 \\ 0 & \mathbf{L}^{(2)} & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & \mathbf{L}^{(k)} \end{pmatrix}$$

- Eigenvectors: union of the zero-padded **eigenvectors of the individual blocks** (clusters)

A. Y. Ng, M. I. Jordan, and Y. Weiss. *On spectral clustering: analysis and an algorithm*. In NIPS 14, 2001.

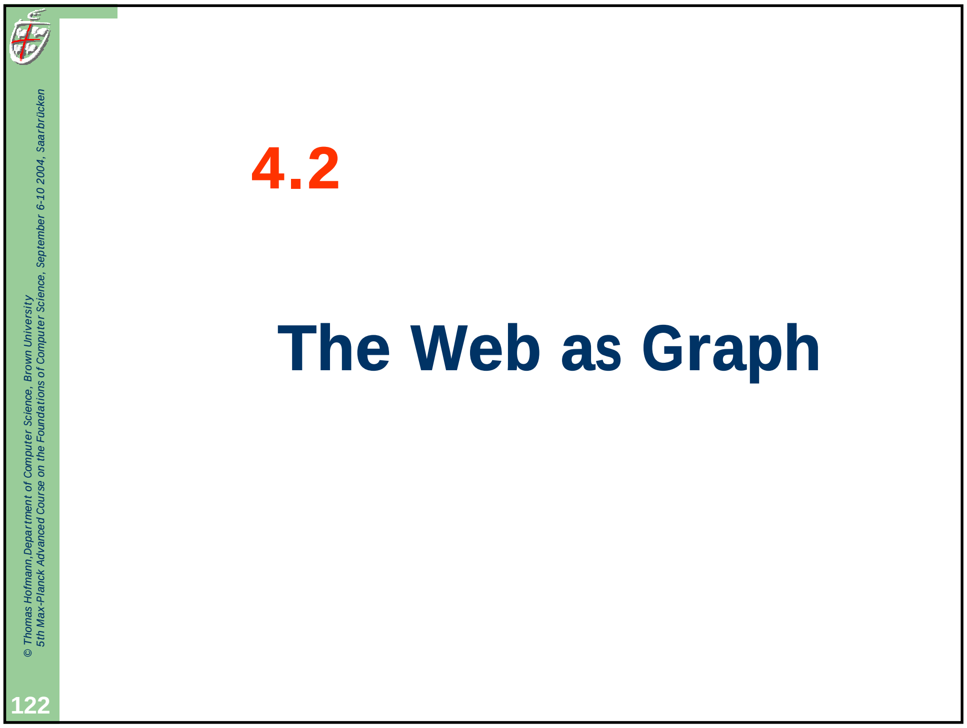


Ideal Case Analysis

- **Spectral graph theory:**
  - Each block has **exactly one** strictly positive **eigenvector with eigenvalue 1** (principal eigenvector)
  - All other eigenvalues are **strictly less than 1**.
- Picking k dominant eigenvectors, where k equals the true number of clusters, one gets:

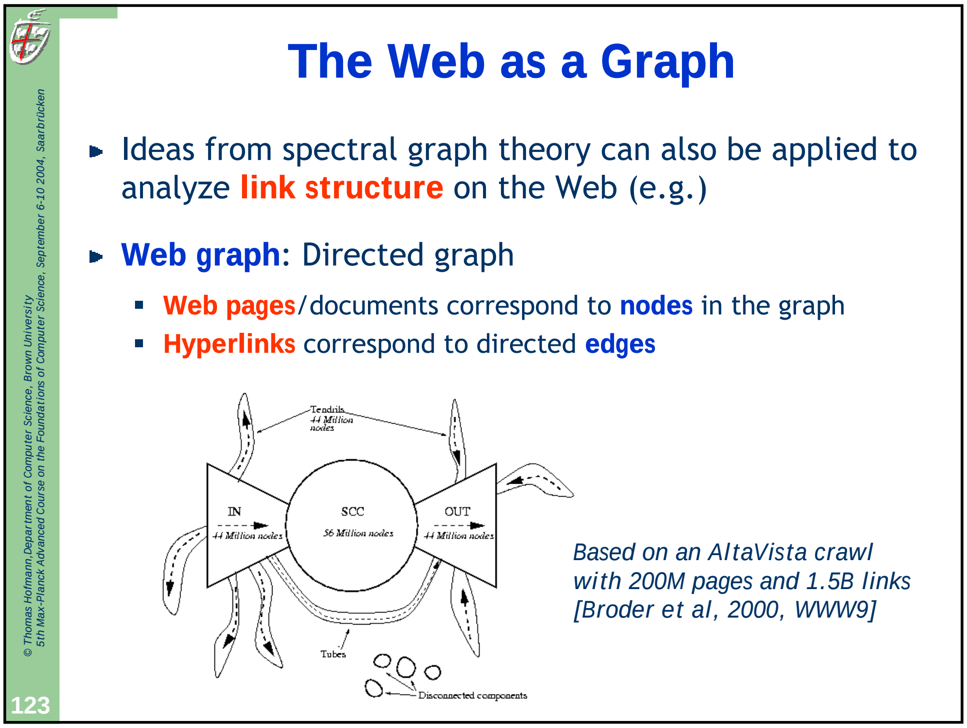
$$\mathbf{U} = \begin{pmatrix} \mathbf{u}^{(1)} & 0 & \dots & 0 \\ 0 & \mathbf{u}^{(2)} & \dots & 0 \\ 0 & 0 & \dots & \mathbf{u}^{(k)} \end{pmatrix} \quad \text{and} \quad \hat{\mathbf{U}} = \begin{pmatrix} \mathbf{1} & 0 & \dots & 0 \\ 0 & \mathbf{1} & \dots & 0 \\ 0 & 0 & \dots & \mathbf{1} \end{pmatrix}$$

  - In fact one may get $\mathbf{UR}$ for some orthogonal matrix $\mathbf{R}$
- Clusters correspond to (**orthogonal**) points on unit sphere (=well separated)



4.2

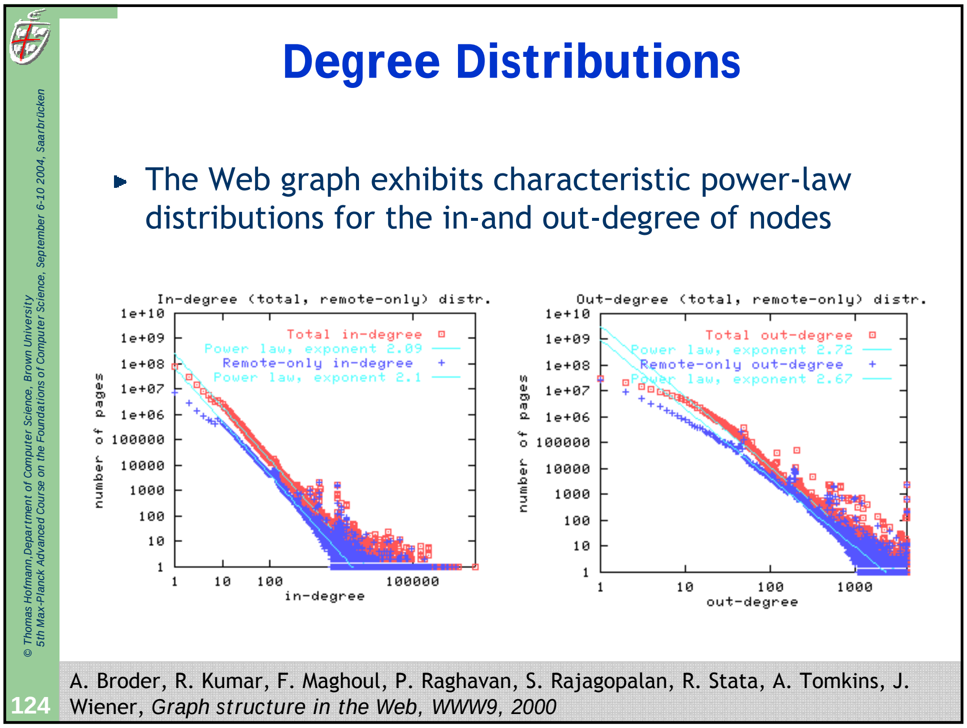
The Web as Graph



The Web as a Graph

- Ideas from spectral graph theory can also be applied to analyze **link structure** on the Web (e.g.)
- **Web graph**: Directed graph
  - **Web pages**/documents correspond to **nodes** in the graph
  - **Hyperlinks** correspond to directed **edges**

Based on an AltaVista crawl with 200M pages and 1.5B links [Broder et al, 2000, WWW9]



Degree Distributions

- The Web graph exhibits characteristic power-law distributions for the in-and out-degree of nodes

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, *Graph structure in the Web, WWW9, 2000*

The Web as a Matrix

- Form adjacency matrix of Web graph
  - Extremely sparse
  - Extremely huge
- Analysis of Web matrix:
  - Determine importance of a Web page: Google (**PageRank**)
  - Find authoritative pages on particular topics: **HITS**
  - Identify Web communities
    - “Bipartite cores”
    - Decomposition

4.3

Hypertext Induced Topic Search

Authority from Hyperlinks

- **Motivation**: different types of queries in IR & search
  - **specific questions**: “in which city lived Kant most of his life?”
  - **broad-topic queries**: “find information on Nietzsche”
  - **similarity queries**: “find pages similar to www.....de/hegel”
- **Abundance problem** for broad-topic queries
  - *“Abundance Problem: The number of pages that could reasonably be returned as relevant is far too large for a human user to digest.”* [Kleinberg 1999]
  - Goal: identify those relevant pages that are the most **authoritative** or definitive ones.
- **Hyperlink structure**
  - Page content is insufficient to define authoritativeness
  - Exploit **hyperlink structure** as source of latent/implicit human judgment to assess and quantify autoritativeness

---

Hubs & Authorities

- Associate two numerical scores with each document in a hyperlinked collection: authority score and hub score
  - **Authorities:** most definitive information sources (on a specific topic)
  - **Hubs**: most useful compilation of links to authoritative documents
- Basic presumptions
  - Creation of links indicates judgment: conferred authority, endorsement
  - Authority is not conferred directly from page to page, but rather mediated through hub nodes: authorities may not be linked directly but through co-citation
  - *Example: major car manufacturer pages will not point to each other, but there may be hub pages that compile links to such pages*

J. Kleinberg. *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998

Hub & Authority Scores

- *"Hubs and authorities exhibit what could be called a mutually reinforcing relationship: a good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs"* [Kleinberg 1999]
- Notation

Directed Graph	$G = (V, E), \quad E \subseteq V \times V$
Authority score of page i	$x_i, \quad i \in V$
Hub score of page i	$y_i, \quad i \in V$

- **Consistency relationship** between two scores

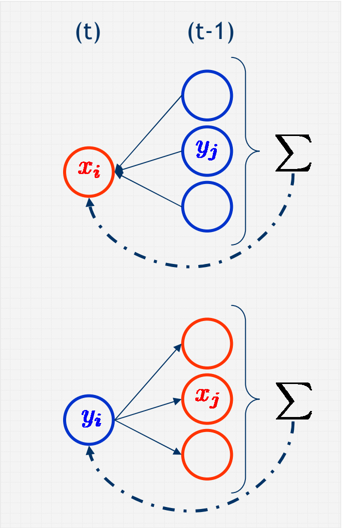
$$x_i \propto \sum_{j:(j,i)\in E} y_j \quad \text{and} \quad y_i \propto \sum_{j:(i,j)\in E} x_j, \quad \forall i \in V$$

Iterative Score Computation (1)

- Translate mutual relationship into **iterative update equations**

$$x_i^{(t)} \propto \sum_{j:(j,i)\in E} y_j^{(t-1)}$$

$$y_i^{(t)} \propto \sum_{j:(i,j)\in E} x_j^{(t-1)}$$



Iterative Score Computation (2)

- Matrix notation

$$\mathbf{x}^{(t)} \propto \mathbf{A}^T \mathbf{y}^{(t-1)}, \qquad \mathbf{y}^{(t)} \propto \mathbf{A}\,\mathbf{x}^{(t-1)}$$

Adjacency matrix

$$\mathbf{A} = (a_{ij}), \quad a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Score vectors

$$\mathbf{x} = (x_1, \ldots, x_{|V|})^T$$
$$\mathbf{y} = (y_1, \ldots, y_{|V|})^T$$

Iterative Score Computation (3)

- Condense into a single update equation (e.g.)

$$\mathbf{x}^{(t)} \propto \left(\mathbf{A}^T \mathbf{A}\right) \mathbf{x}^{(t-1)}$$

- Question of convergence *(ignore absolute scale)*

$$\mathbf{x}^{(1)} \leftarrow \mathbf{A}^T \mathbf{1}, \quad \mathbf{1} \equiv (1, \ldots, 1)^T$$

$$\mathbf{x}^{(\infty)} \equiv \lim_{t\to\infty} \frac{\mathbf{x}^{(t)}}{\|\mathbf{x}^{(t)}\|}$$

→ Existence ?

→ Uniqueness ?

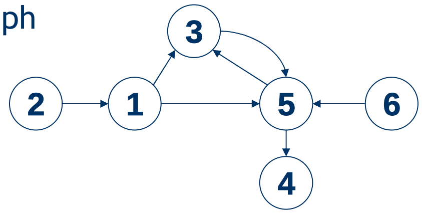
- Notice resemblance with eigenvector equations

$$\mathbf{u} = \lambda \mathbf{L} \mathbf{u}$$

Example

- Simple example graph

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



- Hub & authority matrices

$$\mathbf{A}\mathbf{A}^T = \begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \qquad \mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

- Authority and Hub weights

$$\mathrm{x}^T = (0 \quad 0 \quad .3660 \quad .1340 \quad .5 \quad 0)\,,$$
$$\mathrm{y}^T = (.3660 \quad 0 \quad .2113 \quad 0 \quad .2113 \quad .2113)\,.$$

Convergence

- Notation: enumeration of eigenvalues of $\mathbf{A}^T\mathbf{A}$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0, \quad n = |V|$$

note: symmetric and positive semi-definite

- Pick orthonormal basis of eigenvectors

$$\omega_i = \lambda_i \left(\mathbf{A}^T\mathbf{A}\right) \omega_i, \quad \langle \omega_i, \omega_j \rangle = \delta_{ij}$$

- Technical assumption

$\lambda_1 > \lambda_2$ i.e. largest (abs.) eigenvalue is of multiplicity 1

- **Theorem:** (using the above definitions and assumptions)

$$\mathbf{x}^{(\infty)} = \pm\omega_1$$

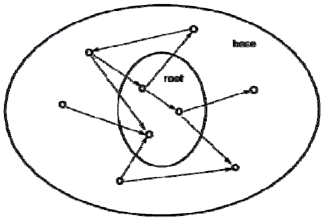
i.e. authority score is dominant eigenvector of $\mathbf{A}^T\mathbf{A}$

Convergence

- Follows from standard linear algebra result (e.g. Golub and Van Loan [1989]) = **power method**
- Requires that $\mathbf{x}^{(1)} = \mathbf{A}^T\mathbf{1}$ is **not** orthogonal to ω_1
- Follows from ...
- **Corollary**: If a matrix $\mathbf{M}$ has only non-negative entries, then $\omega_1(\mathbf{M})$ has only non-negative entries as well.
- If matrix $\mathbf{A}^T\mathbf{A}$ is not irreducible, then solution will depend on initialization, otherwise initialization is basically irrelevant.

Focused Web Graph

- The above analysis applied to a subgraph of the Web graph ⇒ **focused subgraph**
- Subgraph should be determined based on a specific **query**
  - should include most of the authoritative pages
  - use simple key-word matching plus graph expansion
- Use text-based search engine to create a **root set** of matching documents
- Expand root set to form **base set**
  - context graph of depth 1
  - additional heuristics

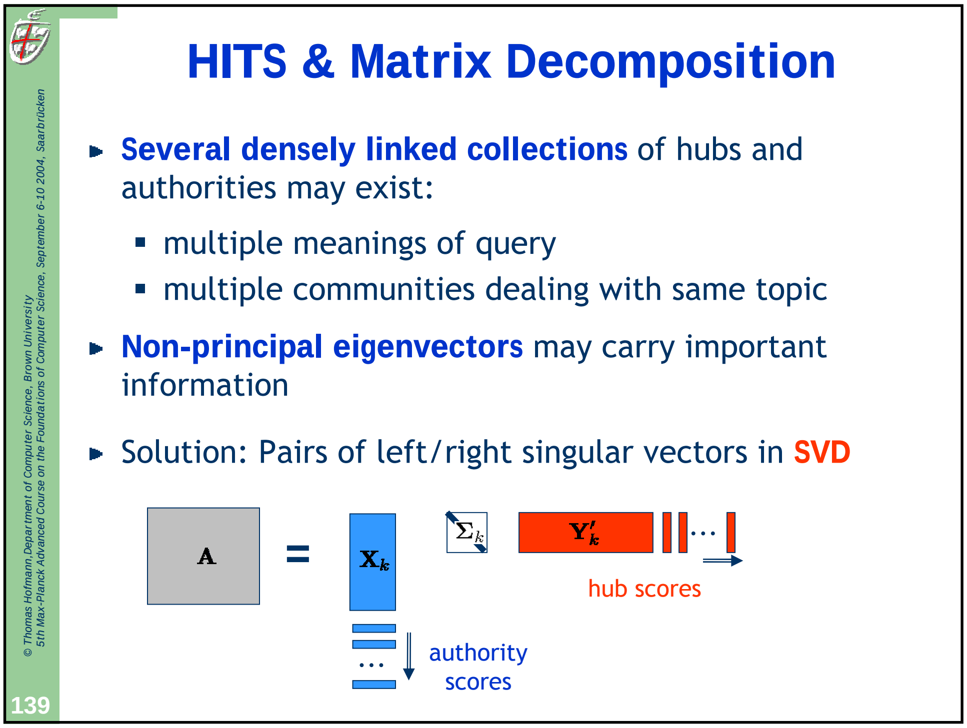


Hypertext Induced Topic Search (HITS)

- **Step 1: Generate focused subgraph G=(V,E)**
  - retrieve top **r** result pages for query and add results to **V**
  - for each result page p: add all pages to **V** to which p points to
  - for each result page p:
    - add all pages to **V** which point to p, if their number is less or equal to **s**
    - otherwise randomly select a set of **s** pages of the pages pointing to p
  - define **E** to be the subset of links within **V**
- **Step 2: Hub-and-Authority Computation**
  - form adjacency matrix **A**
  - compute authority and hub scores **x** and **y** using the iterative power method with **k** iterations
  - return authority and hub result lists with the top **q** pages ranked according to the authority and hub scores, respectively

HITS: Discussion

- Pros
  - Derives **topic-specific** authority scores
  - Returns **list of hubs** in addition to authorities
  - Computational **tractable** (due to focused subgraph)
- Cons
  - Sensitive to **Web spam** (artificially increasing hub and authority weight)
  - **Query dependence** requires expensive context graph building step
  - **Topic drift**: dominant topic in base set may not be the intended one
- *Off-line: Serge Brin and Larry Page are soon-to-become-billionaires, Jon Kleinberg probably not. One reason for this is that HITS is less well-suited as the basis for a Web search engine.*



HITS & Matrix Decomposition

- **Several densely linked collections** of hubs and authorities may exist:
  - multiple meanings of query
  - multiple communities dealing with same topic
- **Non-principal eigenvectors** may carry important information
- Solution: Pairs of left/right singular vectors in **SVD**

4.4

Probabilistic HITS

Probabilistic HITS

- Probabilistic model of link structure
  - Probabilistic graph model, i.e., predictive model for additional links/nodes based on existing ones
  - Centered around the notion of "**Web communities**"
  - Probabilistic version of HITS
  - Enables to predict the existence of hyperlinks: estimate the entropy of the Web graph
- Combining with content
  - Text at every node ...

D. Cohn and T. Hofmann. *The missing link - a probabilistic model of document content and hypertext connectivity*. In NIPS 13, 2001.

Finding Latent Web Communities

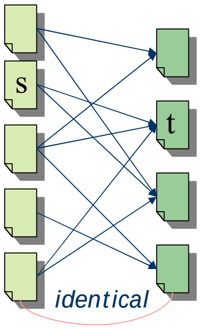
- Web Community: **densely connected bipartite subgraph**
- Probabilistic model **pHITS** (cf. pLSA model)

$$P(s \to t) = \sum_z P(s \to z)P(t \leftarrow z)P(z)$$

Source nodes

$P(s \to z)$

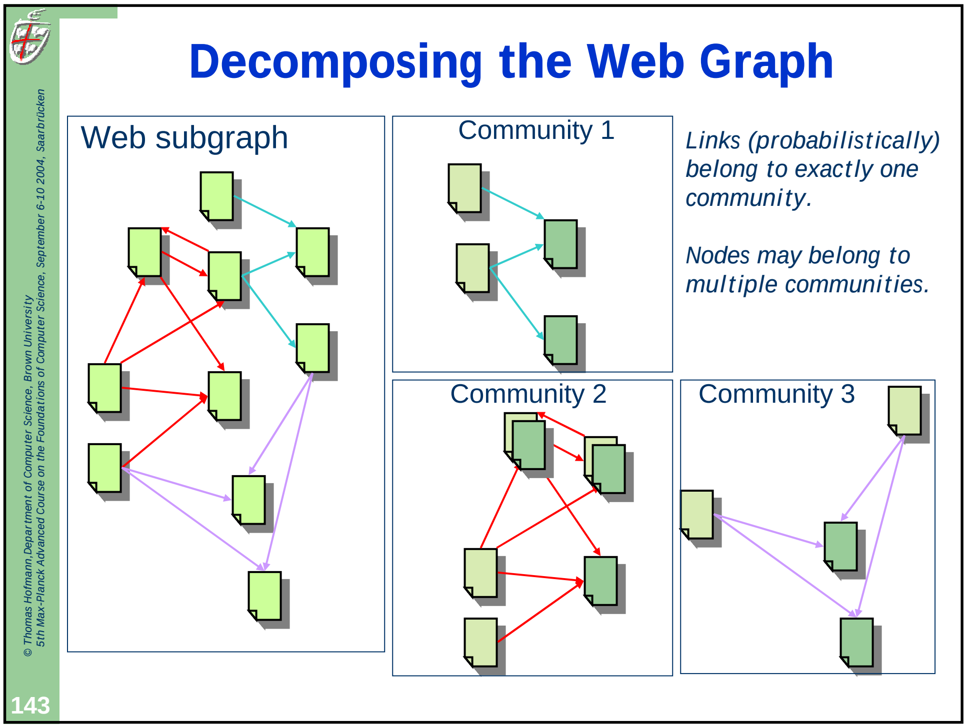
probability that a random out-link from s is part of the community z



Target nodes

$P(t \leftarrow z)$

probability that a random in-link from t is part of the community z



Decomposing the Web Graph

Web subgraph

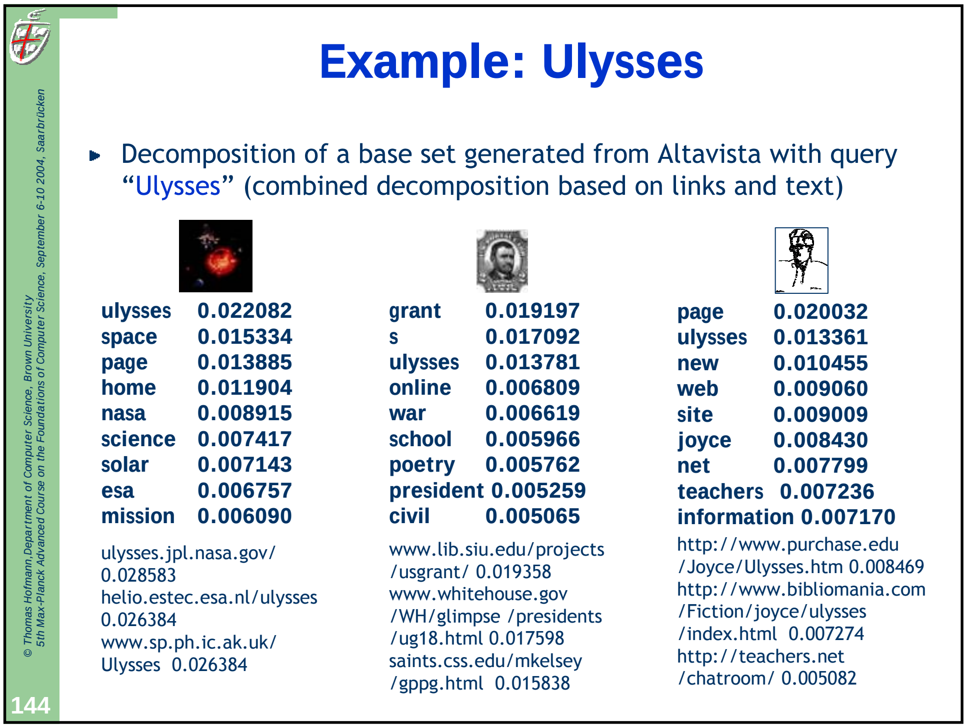
Community 1

Community 2

Community 3

Links (probabilistically) belong to exactly one community.

Nodes may belong to multiple communities.



Example: Ulysses

- Decomposition of a base set generated from Altavista with query "Ulysses" (combined decomposition based on links and text)

ulysses	**0.022082**	**grant**	**0.019197**	**page**	**0.020032**
space	**0.015334**	**s**	**0.017092**	**ulysses**	**0.013361**
page	**0.013885**	**ulysses**	**0.013781**	**new**	**0.010455**
home	**0.011904**	**online**	**0.006809**	**web**	**0.009060**
nasa	**0.008915**	**war**	**0.006619**	**site**	**0.009009**
science	**0.007417**	**school**	**0.005966**	**joyce**	**0.008430**
solar	**0.007143**	**poetry**	**0.005762**	**net**	**0.007799**
esa	**0.006757**	**president**	**0.005259**	**teachers**	**0.007236**
mission	**0.006090**	**civil**	**0.005065**	**information**	**0.007170**

ulysses.jpl.nasa.gov/ 0.028583
helio.estec.esa.nl/ulysses 0.026384
www.sp.ph.ic.ak.uk/ Ulysses 0.026384

www.lib.siu.edu/projects /usgrant/ 0.019358
www.whitehouse.gov /WH/glimpse /presidents /ug18.html 0.017598
saints.css.edu/mkelsey /gppg.html 0.015838

http://www.purchase.edu /Joyce/Ulysses.htm 0.008469
http://www.bibliomania.com /Fiction/joyce/ulysses /index.html 0.007274
http://teachers.net /chatroom/ 0.005082

4.5

PageRank & Google

Google

- Exploit link analysis to derive a **global "importance" score** for each Web page (PageRank)
- Crucial to deal with "document collections" like the Web which exhibit a **high degree of variability in document quality**
- Assumptions:
  - Hyperlinks provide **latent human annotation**
  - Hyperlinks represent an **implicit endorsement** of the page being pointed to
- **In-degree** alone is not sufficient
  - Can be artificially inflated
  - In-links from important documents should receive more weight

PageRank

- Recursive definition (models intuition of propagation of importance score)

rank

$$r_i = \sum_{j:(j,i)\in E} \frac{r_j}{d_j}$$

out-degree

- Matrix notation $\pi = (r_1, \dots, r_n)'$

$$\mathbf{P} \in \mathbb{R}^{n\times n},\ p_{ij} = \begin{cases} 1/d_j & \text{if } (j,i) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = \mathbf{P}\pi$$

eigenvector with eigenvalue of 1

dominant eigenvector of a stochastic or Markov matrix

Random Surfer

- The **P**-matrix is the transition matrix of a **Markov chain**
- This models a memoryless stochastic **surfer**:
  - At every page, **randomly** chose one of the links and follow the link to the next page
  - Repeat ad infinitium
  - PageRanks should correspond to the probabilities of a **stationary distribution**
- In order to ensure **irreducibility** of chain (avoid getting trapped in subgraphs): **teleportation probability**

$$\bar{\mathbf{P}} = \alpha\mathbf{P} + (1-\alpha)\frac{1}{n}\mathbf{1}\mathbf{1}'$$

teleportation probability to random page/node

PageRank Computation

- Use **power method** to compute principal eigenvector of the irreducible stochastic matrix $\bar{\mathbf{P}}$
- Multiplicity of dominant eigenvalue is 1, all other eigenvalues have modulus strictly less than 1
- Convergence speeds depends on **separation** between dominant and sub-dominant eigenvalues (can be controlled by α)