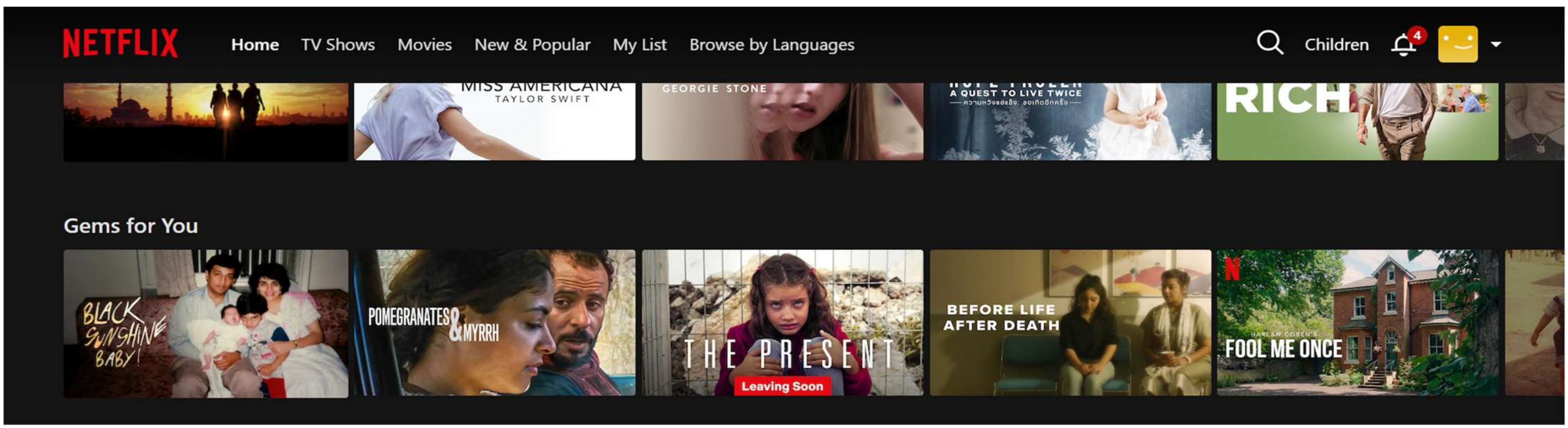
Recommender System

Matrix Factorization

Latent Semantic Indexing

Recommender Systems

- Recommender systems are information filtering systems that aim to predict users' preferences and recommend items (such as products, movies, music, etc.) that they might like.
- Most e-commerce sites have such systems
- Mainly serve two important functions
  - Help users deal with the information overload by giving them recommendations of products, etc.
  - Help business make more profits, i.e., selling more products



Everyday Applications

- **E-commerce**: Recommending products to users based on their purchase history or browsing behavior.
- **Streaming services**: Recommending movies, TV shows, or music based on users' viewing or listening history.
- **Social media**: Recommending friends, groups, or posts based on users' interests and social connections.
- **News websites**: Personalizing article recommendations based on users' reading habits and preferences.

Recommendation System – Basic Approaches

- **Content-based filtering**: Recommends items similar to those the user liked in the past, based on the attributes of the items.
- **Collaborative filtering**: Recommends items based on the preferences of other users with similar tastes.
- **Hybrid approaches**: Combines content-based and collaborative filtering techniques to provide more accurate recommendations.

Content Based Recommendation

- Content-based recommendation systems suggest items to users based on the features or characteristics of the items themselves, rather than relying on user-item interactions or similarities between users.

➢ In a content-based movie recommendation system, each movie is described by its attributes such as genre, director, actors, plot keywords, and ratings.

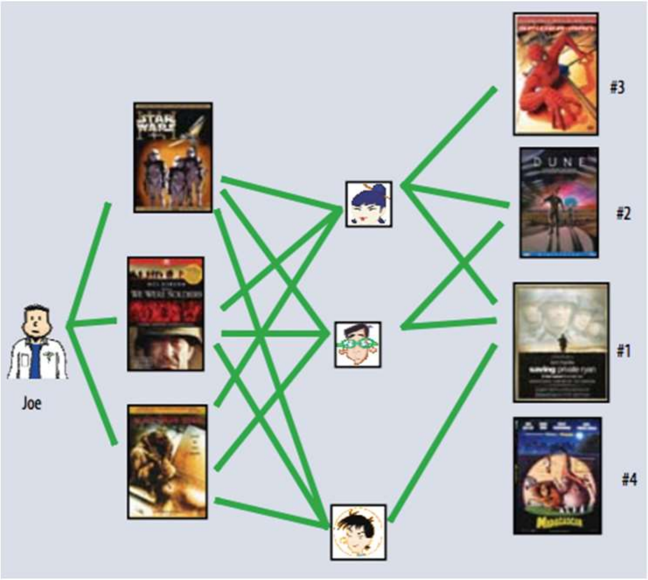
➢ If a user has previously liked action movies starring a particular actor, the system can recommend other action movies with the same actor.

- Recommendations are made by computing the similarity of the user profile with the candidate items expressed in the same set of features and selecting the top matched ones.

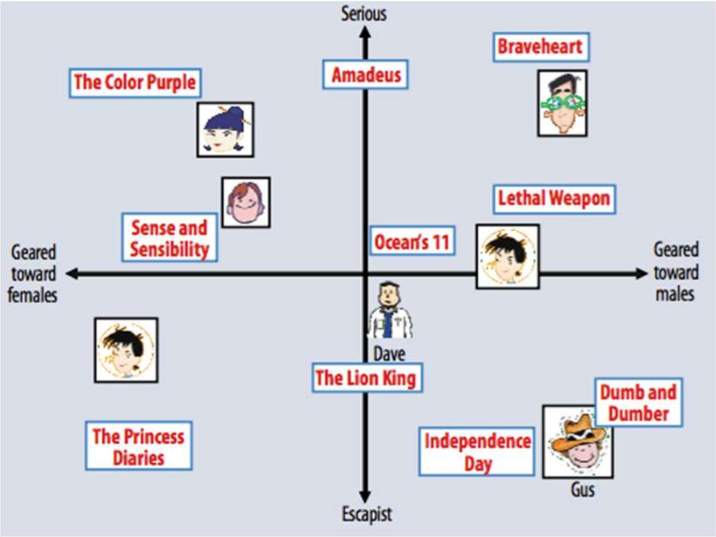
Collaborative Filtering

- Collaborative Filtering is the most-widely used recommendation approach in practice, where it predicts the utility of items for a user based on the items previously rated by other like minded users.

1. Neighbourhood Methods



2. Latent Factor Methods



Figures from Koren et al. (2009)

What is a latent variable?

- It is a variable that is not directly observed or measured but is instead inferred from other observed variables.
- Latent variables are a transformation of the data points into a continuous lower-dimensional space

Person	Age	Liked	Movie	Genre
X	16	Y	Spiderman	Action
Y	9	N	Hangover	Comedy
Y	9	Y	Clueless	Comedy
X	16	Y	Black Panther	Action
Y	9	Y	Terminal	Comedy
Z	27	Y	Annabelle	Horror
X	16	Y	Star wars	Action
Z	27	N	The Nun	Horror
Z	27	Y	Conjuring	Horror
Y	9	Y	Ted	Comedy

Latent Factor Methods

- Assumes that both the users and movies live in some low-dimensional space describing their properties.

- Recommends a movie based on its proximity to the user in the latent space.

- Latent Features Evaluated with:
  - Matrix Factorization
    - Singular Value Decomposition [SVD]
      - Low Rank Matrix Approximation
    - LU Decomposition
      - Solving system of equations

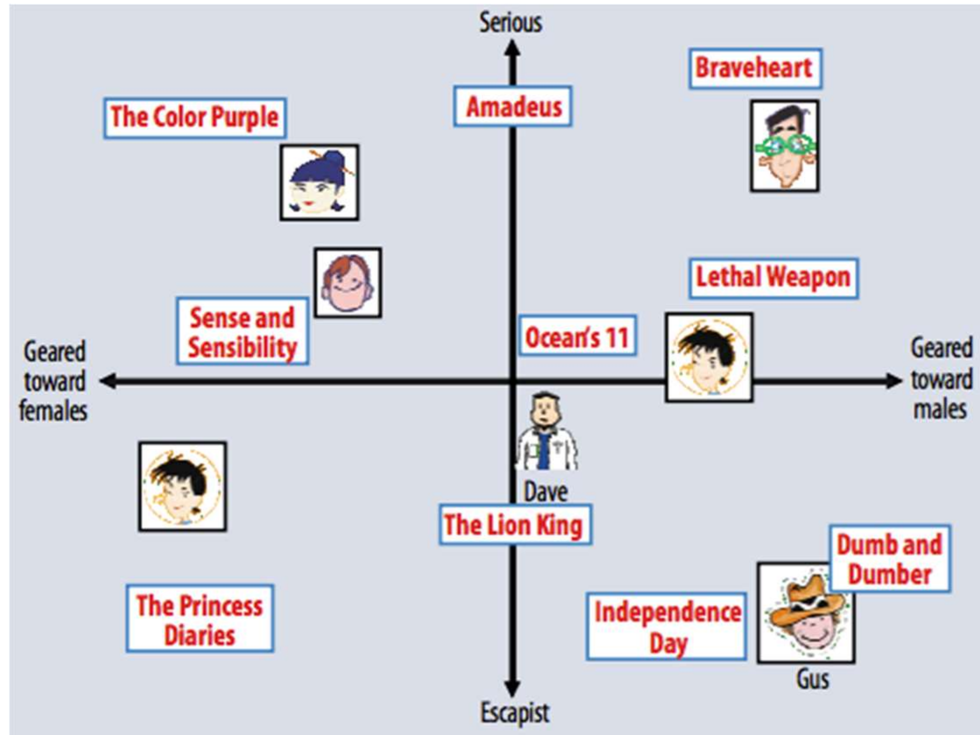


Figure from Koren et al. (2009)

Matrix Factorization

- Matrix Factorization is the process of decomposing a matrix M into a product of several factor matrices, i.e., $\boldsymbol{M} = \boldsymbol{F_1} \times \boldsymbol{F_2} \times \cdots \times \boldsymbol{F_N}$, where N can be any number.

$$\begin{pmatrix} 12 & 17 & 7 & 5 & 10 \\ 11 & 16 & 6 & 5 & 10 \\ 11 & 15 & 7 & 4 & 8 \\ 8 & 11 & 5 & 3 & 6 \end{pmatrix}_{4 \times 5 \text{ matrix}} = \begin{pmatrix} 3 & 2 \\ 4 & 1 \\ 1 & 3 \\ 1 & 2 \end{pmatrix}_{4 \times 2} \times \begin{pmatrix} 2 & 3 & 1 & 1 & 2 \\ 3 & 4 & 2 & 1 & 2 \end{pmatrix}_{2 \times 5}$$

- Similarly a matrix $\boldsymbol{M}_{1000\times 50} = \boldsymbol{A}_{1000\times 5} \times \boldsymbol{B}_{5\times 50}$.

- Matrix Factorization is a way to generate latent features.

Movie Recommendation System

- Consider a $n \times m$ rating matrix R with some entries unknown:
  - n rows represent n users
  - m columns represent m movies
  - Entry R_{ij} represents the i^{th} user's ratings on the j^{th} movie

- We are interested in predicting user's ratings which are the possible values of the unknown entries

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User 1	3	5	4	4	2
User 2	5	3		3	1
User 3		2	5	3	1
User 4	5	3	2	3	1

Matrix Factorization

If we can learn U and V from existing ratings, then we can compute unknown entries by multiplying these two matrices

- We can model the problem as $R_{n\times m} = |U_{n\times k}| \times |V_{m\times k}|^T$ with $k \ll m, n$

- The matrix $U_{n\times k}$ is the latent feature matrix for users, with each row of U representing the strength of association between the user and the features.

    - How much the user likes comedy movies?
    - How much the user like thriller movies?

- The matrix $V_{m\times k}$ is the latent feature matrix for movies, with each row of V representing the strength of association between the movies and the features.

    - To what extent is the movie a comedy movie?
    - To what extent is the movie a thriller movie?

Mathematics of Matrix Factorization

- Rating Matrix can be decomposed as $\boldsymbol{R_{n\times m} = |U_{n\times k}| \times |V_{m\times k}|^T = R^{pred}_{n\times m}}$, $\boldsymbol{k \ll m, n}$

$$\boldsymbol{r^{pred}_{ij} = \sum_{l=1}^{k} u_{il} v_{lj}}$$

- The matrices U, V can be obtained using the **Gradient Descent Method**, considering squared error as the loss function

$$\boldsymbol{J = e^2_{ij} = \left(r_{ij} - r^{pred}_{ij}\right)^2 = \left(r_{ij} - \sum_{l=1}^{k} u_{il} v_{lj}\right)^2}$$

Gradient Descent

1. Initialize the matrices U, V with random values
2. Compute $R_{n\times m}^{pred}$ and calculate the squared error e_{ij}^2
3. Compute the gradient of the error function J at u_{il}, v_{lj},
   $\nabla J = {\partial J}/{\partial u_{il}}, {\partial J}/{\partial v_{lj}}$
4. $u_{il}^{t+1} = u_{il}^{t} + \alpha \frac{\partial J}{\partial u_{il}}, \quad v_{lj}^{t+1} = v_{lj}^{t} + \alpha \frac{\partial J}{\partial v_{lj}}$
5. Repeat steps 2,3 and 4 until convergence

$$\frac{\partial J}{\partial u_{il}} = -2\left(r_{ij} - r_{ij}^{pred}\right) v_{lj} = -2 e_{ij} v_{lj}$$

$$\frac{\partial J}{\partial v_{lj}} = -2\left(r_{ij} - r_{ij}^{pred}\right) u_{il} = -2 e_{ij} u_{il}$$

Convergence: When $\alpha \frac{\partial J}{\partial u_{il}}, \alpha \frac{\partial J}{\partial v_{lj}}$ becomes small.

- Regularization can also be introduced to avoid overfitting:

$$J = e_{ij}^2 = \left(r_{ij} - \sum_{l=1}^{k} u_{il} p_{lj}\right)^2 + \frac{\beta}{2} \sum_{l=1}^{k} \left(\|U\|^2 + \|V\|^2\right)$$

Non-Negative Matrix Factorization (NMF)

- NMF approximates a non-negative matrix R with a low-rank matrix approximation such that $\boldsymbol{R_{m\times n}} \approx \boldsymbol{W_{m\times k}H_{k\times n}}$, where $\boldsymbol{W, H}$ are non-negative matrices.
  - Each column of W is a basis element, i.e. captures the underlying features or patterns present in the data.
  - Each column of H gives the 'coordinates of a data point' in the basis of W, i.e. represents how the features contribute to each sample in the dataset.
  - Often results in sparse factor matrices – many entries in W, H are close to zero. Sparsity helps in identifying the most relevant features and reducing noise in the data.

- NMF ensures that both the basis and coefficient matrices contain non-negative elements, which makes the resulting factors more interpretable, especially in applications such as topic modeling or image processing.

NMF – Image Processing

$$\underbrace{X(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\substack{\text{importance of features}\\ \text{in } j\text{th image}}} = \underbrace{WH(:,j)}_{\substack{\text{approximation}\\ \text{of } j\text{th image}}} .$$

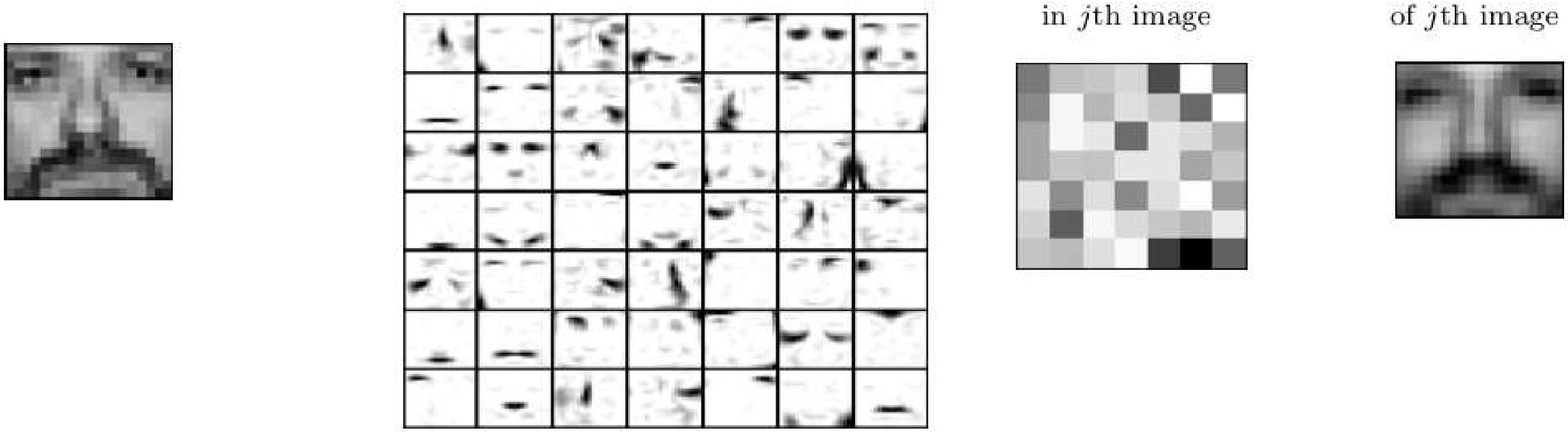


Figure 1: Decomposition of the CBCL face database, MIT Center For Biological and Computation Learning (2429 gray-level 19-by-19 pixels images) using $r = 49$ as in [79].

https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/

Singular Value Decomposition (SVD)

- SVD gives the decomposition for any arbitrary matrix, $\boldsymbol{R} = \boldsymbol{U}\,\boldsymbol{\Lambda}\,\boldsymbol{V}^{\boldsymbol{T}}$

$$\boldsymbol{R}_{m\times n} = \boldsymbol{U}_{m\times k}\,\boldsymbol{\Lambda}_{k\times k}\,\boldsymbol{V}^{\boldsymbol{T}}_{k\times n}$$

✓ R is a $m \times n$ ranking matrix

✓ U is a $m \times k$ orthogonal left singular matrix ($\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U} = \mathbf{1}$), representing the relationship between users and the latent factors

- Consists of orthonormal eigenvectors of RR^T

✓ Λ is the $k \times k$ diagonal matrix describing the strength of each latent factor

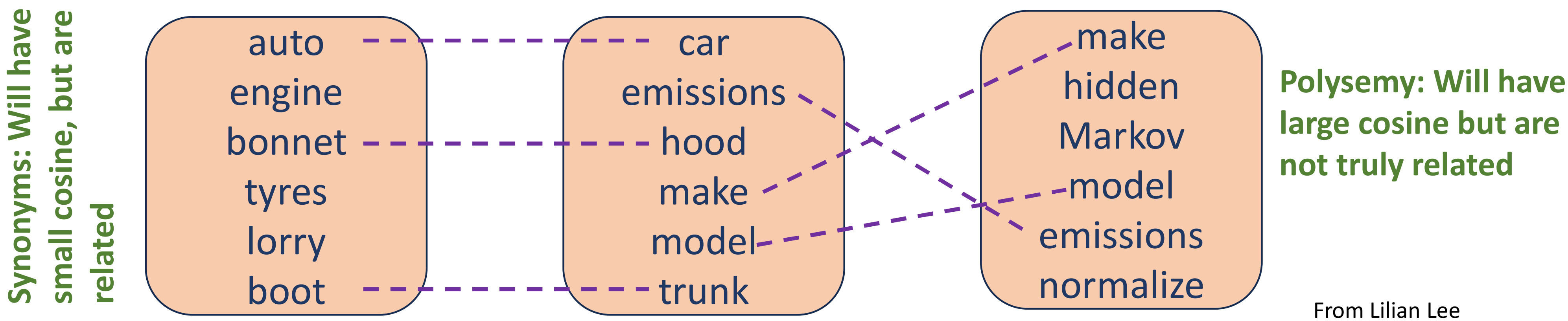
- Equal to the root of the positive eigenvalues of RR^T or R^TR

✓ V is the $k \times n$ orthogonal right singular matrix, indicating the similarity between movies and the latent factors

- Consists of orthonormal eigenvectors of R^TR

Vector Space Method

- Given a collection of documents, retrieve documents that are relevant to a given query
  - Match terms in documents to terms in query
- The vector space method was used, with the terms(rows) by document (columns) matrix, based on occurrence.
  - One vector is assigned for each document
  - Cosine Similarity to measure the distance between vectors
- Cons:
  - Synonyms: Same object referred by many words, e.g., car, automobile [Poor Recall]
  - Polysemy: Most words have more than one distinct meaning, e.g., python, chips [Poor Precision]



From Lilian Lee

Latent Semantic Indexing

- It is a technique used in NLP and information retrieval to analyze relationships between a set of documents and the terms they contain.

- It aims to capture the underlying semantic structure of the text corpus by identifying patterns of word co-occurrence and reducing the dimensionality of the document-term matrix through SVD

Problem Statement: Use Latent Semantic Indexing to rank these documents for the query "**gold silver truck**".

D1: *Shipment of gold damaged in a fire*.
D2: *Delivery of silver arrived in a silver truck*.
D3: *Shipment of gold arrived in a truck*.

Note: LSI vs. LSA

- ✓ LSI refers to indexing or information retrieval
- ✓ LSA refers to everything else

(taken from Grossman and Frieder's Information Retrieval, Algorithms and Heuristics)

Latent Semantic Indexing

1) Set term weights and construct the term-document matrix A and query matrix q

Terms

a
arrived
damaged
delivery
fire
gold
in
of
shipment
silver
truck

$$A = \begin{array}{c} \begin{array}{ccc} \mathbf{D1} & \mathbf{D2} & \mathbf{D3} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

$$q = \begin{array}{c} \mathbf{q} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \end{array}$$

Latent Semantic Indexing

2. The matrix A is decomposed using SVD, and the matrices U, Λ, V are calculated

$$U = \begin{bmatrix} -0.42 & 0.07 & -0.05 \\ -0.29 & -0.20 & 0.41 \\ -0.12 & 0.27 & -0.45 \\ -0.15 & -0.30 & -0.20 \\ -0.12 & 0.27 & -0.45 \\ -0.26 & 0.38 & 0.15 \\ -0.42 & 0.07 & -0.05 \\ -0.42 & 0.07 & -0.05 \\ -0.26 & 0.38 & 0.15 \\ -0.31 & -0.60 & -0.40 \\ -0.29 & -0.20 & 0.41 \end{bmatrix}, \Lambda = \begin{bmatrix} 4.09 & 0.00 & 0.00 \\ 0.00 & 2.36 & 0.00 \\ 0.00 & 0.00 & 1.27 \end{bmatrix}, V = \begin{bmatrix} -0.49 & 0.65 & -0.58 \\ -0.65 & -0.72 & -0.26 \\ -0.58 & 0.25 & 0.78 \end{bmatrix}$$

Latent Semantic Indexing

3. Implement a Rank 2 Approximation by keeping the first two columns of U, V and the first two rows and columns of Λ

$$U = \begin{bmatrix} -0.42 & 0.07 \\ -0.29 & -0.20 \\ -0.12 & 0.27 \\ -0.15 & -0.30 \\ -0.12 & 0.27 \\ -0.26 & 0.38 \\ -0.42 & 0.07 \\ -0.42 & 0.07 \\ -0.26 & 0.38 \\ -0.31 & -0.60 \\ -0.29 & -0.20 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 4.09 & 0.00 \\ 0.00 & 2.36 \end{bmatrix} \quad V = \begin{bmatrix} -0.49 & 0.65 \\ -0.65 & -0.72 \\ -0.58 & 0.25 \end{bmatrix}$$

4. Find the new document vector coordinates in this reduced 2-dimensional space.

$$d1 = (-0.49.\ 0.65)$$
$$d2 = (-0.65, -0.72)$$
$$d3 = (-0.58, 0.25)$$

Latent Semantic Indexing

5. Find the new query vector coordinates in the reduced 2-D space, $q = q^T U_k \Lambda_k^{-1}$

$$q = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1] \begin{bmatrix} -0.42 & 0.07 \\ -0.29 & -0.20 \\ -0.12 & 0.27 \\ -0.15 & -0.30 \\ -0.12 & 0.27 \\ -0.26 & 0.38 \\ -0.42 & 0.07 \\ -0.42 & 0.07 \\ -0.26 & 0.38 \\ -0.31 & -0.60 \\ -0.29 & -0.20 \end{bmatrix} \begin{bmatrix} \frac{1}{4.09} & 0.00 \\ 0.00 & \frac{1}{2.36} \end{bmatrix} = [-0.21 \quad -0.18]$$

Latent Semantic Indexing

6. Rank documents in decreasing order of query-related cosine similarities, $\mathbf{CS}(\boldsymbol{q}, \boldsymbol{d}) = \frac{\boldsymbol{q}.\boldsymbol{d}}{|\boldsymbol{q}||\boldsymbol{d}|}$

$$\mathrm{CS}(q, d_1) = \frac{q.d_1}{|q||d_1|} = -0.05$$

$$\mathrm{CS}(q, d_2) = \frac{q.d_2}{|q||d_2|} = 0.99$$

$$\mathrm{CS}(q, d_3) = \frac{q.d_3}{|q||d_3|} = 0.45$$

- Document d2 scores higher than d3 and d1. Its vector is closer to the query vector than the other vectors.

Latent Semantic Indexing

- LSI captures the underlying semantic structure of the text corpus by identifying latent topics.
- LSI reduces the dimensionality of the document-term matrix, making it more manageable and computationally efficient.
- LSI can improve information retrieval tasks by considering semantic similarity rather than just keyword matching.
- Finding optimal dimension for semantic space sometimes is an issue
    - precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
    - run SVD once with big dimension, say k = 1000, then can test dimensions <= k
    - in many tasks 150-350 works well, still room for research