Latent Semantic Analysis

Dr. Maunendra Sankar Desarkar
IIT Hyderabad

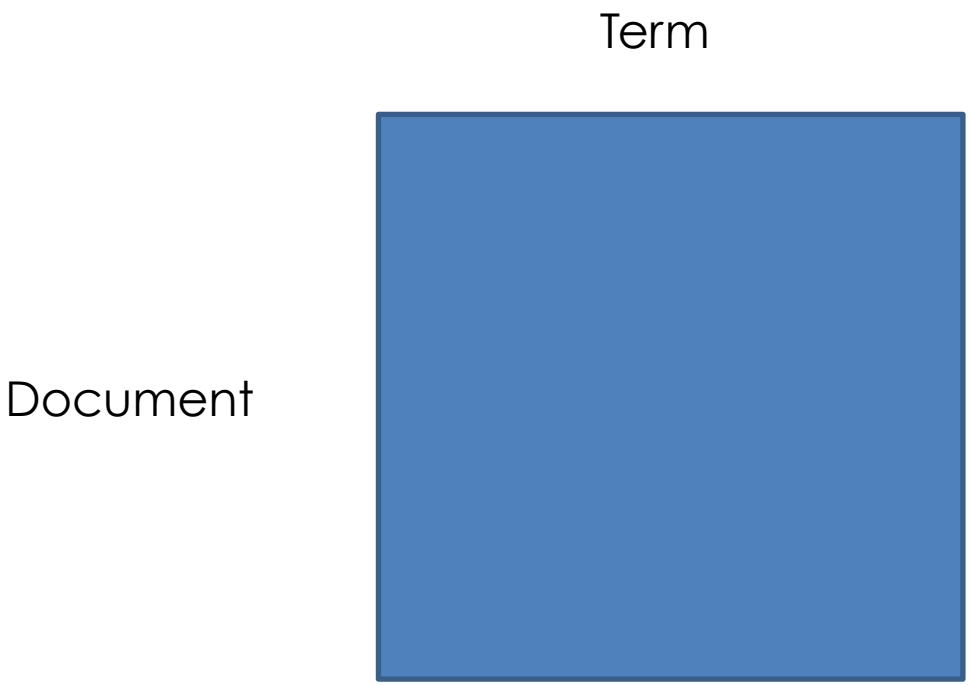
- **Social and Information Networks Analysis: Problems, Models and Machine Learning Methods**
- Functional Foods and Nutraceuticals: Fundamental and Mechanistic Approaches
- High Voltage Engineering Applications
- Fuzzy Techniques for Intelligent Decision Making
- Cultural Diversity in American Theatre
- Big Data Analytics & Business Intelligence
- New approaches in tuberculosis research and drug development
- Advances in Web and Data Analytics
- Online Learning and Principal Component Analysis (PCA) - Theory, Algorithms and Applications
- ALTERNATE ENERGY SOURCES FOR DISTRIBUTED GENERATION

- Text -> Vector
- **Example text:** Social and Information Networks Analysis
- **Corresponding vector (in sparse notation):**
  - **“social”:1, “and”: 1, “information”: 1, “networks”: 1, “analysis”: 1**
  - **“social and”: 1, “and information”: 1, “information networks”: 1, “network analysis”: 1**

	Social	and	Information	network	analysis	social and	and information	information network	network analysis	…
Social and Information Networks Analysis	1	1	1	1	1	1	1	1	1	0

	Social	and	Information	network	analysis	social and	and information	information network	network analysis	…
Social and Information Networks Analysis	1	1	1	1	1	1	1	1	1	0
Social Networks Analysis	1	0	0	1	1	0	0	0	1	0

- C: Document-Term Incident Matrix



- $C : m \times n$ Document-Term Incidence Matrix
  - $C = U \Sigma V^T$
  - U : Corresponds to Document Factors. Rows: Documents, Column: new dimensions
  - V : Corresponds to Term Factors. Rows: Terms, Column: new dimensions
  - Σ : Singular values, arranged in decreasing order, indicating importance of the new dimensions

- Reducing number of dimensions
- $C \approx C_k = U_k \Sigma_k V_k^T$

What is Latent Semantic Analysis?

```
lsimodel = gensim.models.lsimodel.LsiModel (corpus=None, num_topics=200)
```

$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ Matrix

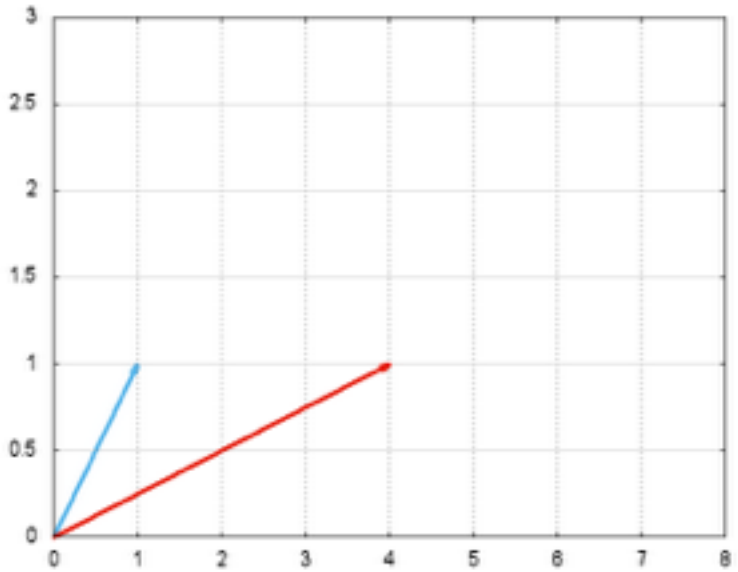
$[1, 5]$ Vector Matrix

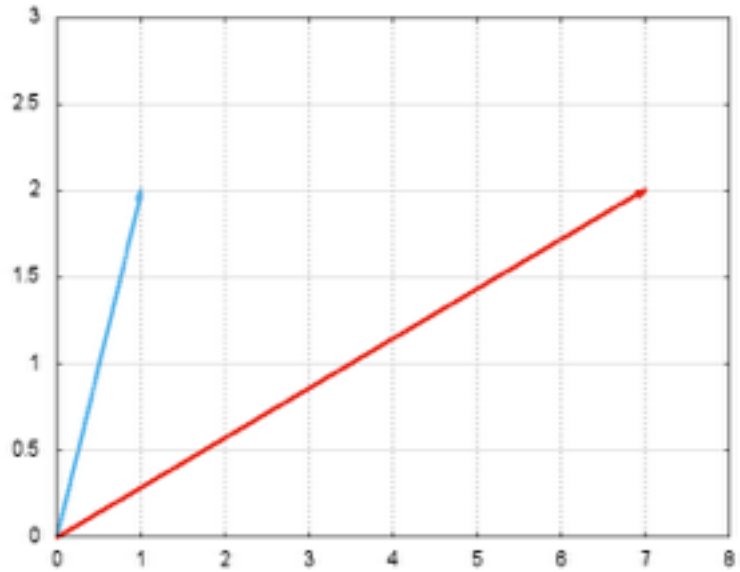
3 Number Vector Matrix

- 3 * 2 = 6

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

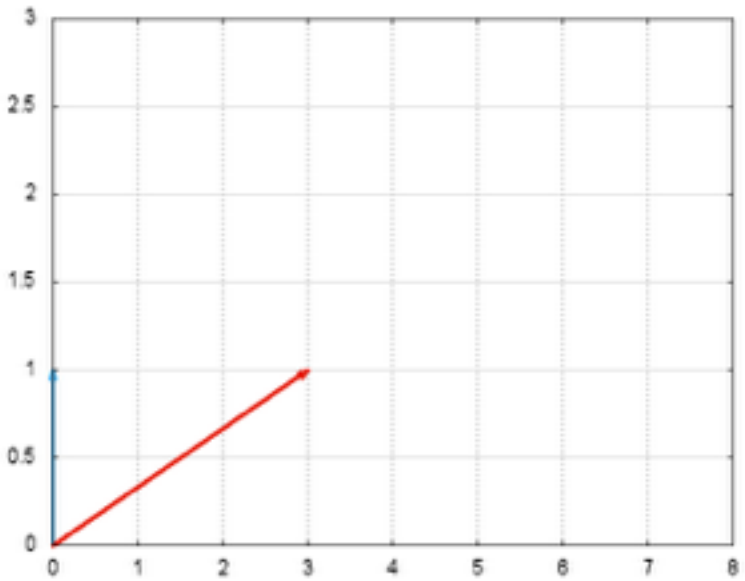
$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$$



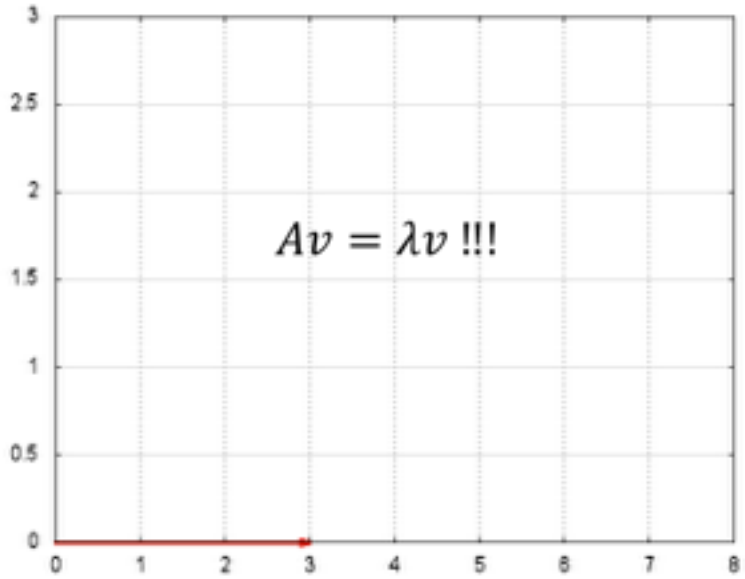


Transformation: Rotation + Scaling

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

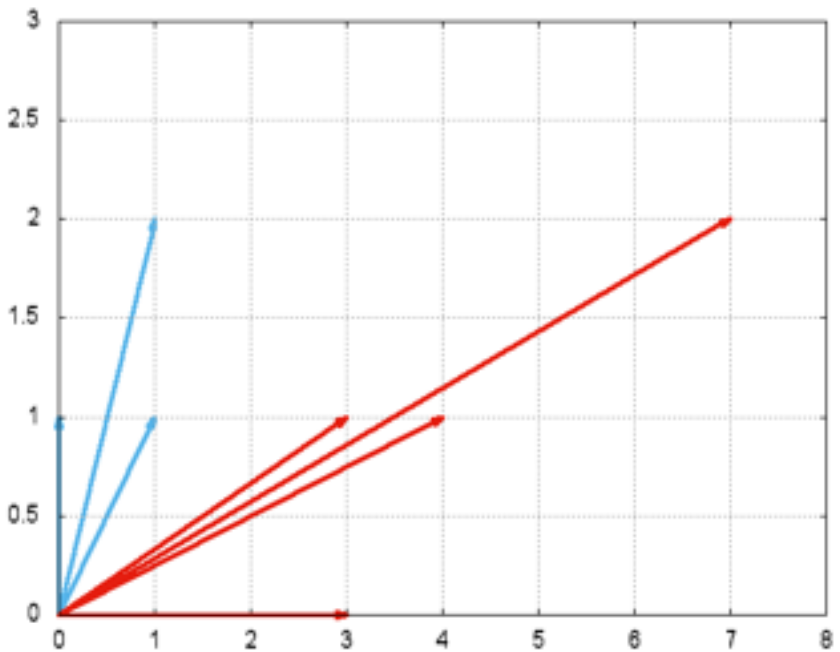


$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$



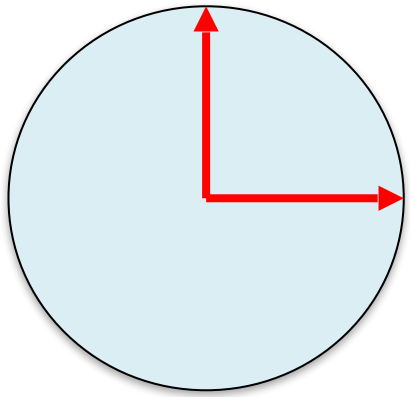
Transformation: Rotation + Scaling

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 3 \\ 1 & 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 7 & 3 & 3 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$



- $A\,B = C$
  - A : Transformation Matrix
  - B : Collection of points
  - C : Transformed representations of the points in B

- Let us consider B to contain the basis vectors in a space
- Then A can also be simply viewed as axis transformation rules for the basis vectors



$$Av_1 = \sigma_1 u_1$$
$$Av_2 = \sigma_2 u_2$$
$$Av_3 = \sigma_3 u_3$$
$$A[v_1\ v_2\ v_3\ \ldots] = [u_1\ u_2\ u_3\ \ \ldots]diag[\sigma_1\ \sigma_2\ \sigma_3\ \ \ldots]$$
$$\ldots$$
$$AV\ = U\ \Sigma$$

Transformation: Rotation + Scaling

- $AV = U\,\Sigma$
  - $A = U\,\Sigma\,V^{-1}$
  - $A = U\,\Sigma\,V^{T}$ (If V is orthonormal)

  - $AA^{T} = U\Sigma^{2}U^{T}$
  - $(AA^{T})U = U\Sigma^{2}$
  - Columns of U are eigen vectors of AA^{T}

  - Columns of V are eigen vectors of $A^{T}A$

SVD example

Let $A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

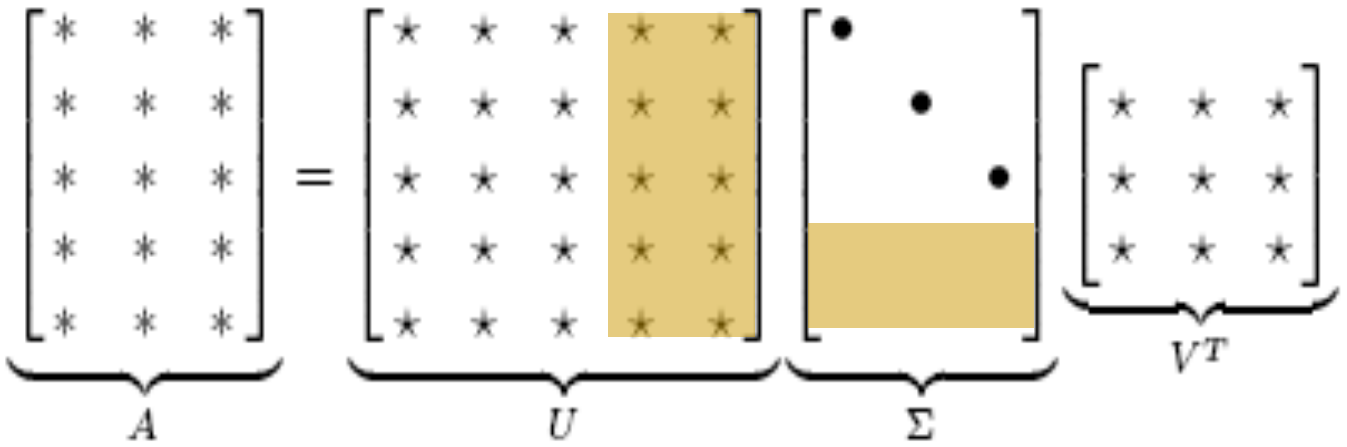
Thus M=3, N=2. Its SVD is

$$\begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Typically, the singular values arranged in decreasing order.

Singular Value Decomposition

- Illustration of SVD dimensions and sparseness

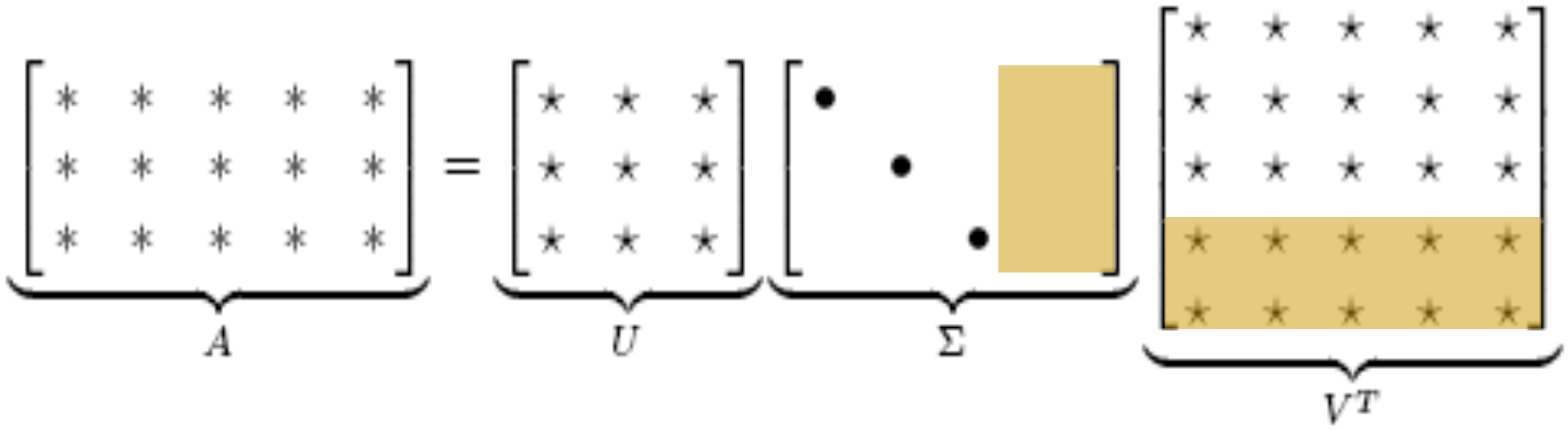


$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{18}} & -\frac{1}{\sqrt{18}} & \frac{4}{\sqrt{18}} \\ \frac{2}{3} & -\frac{2}{3} & -\frac{1}{3} \end{pmatrix}$$

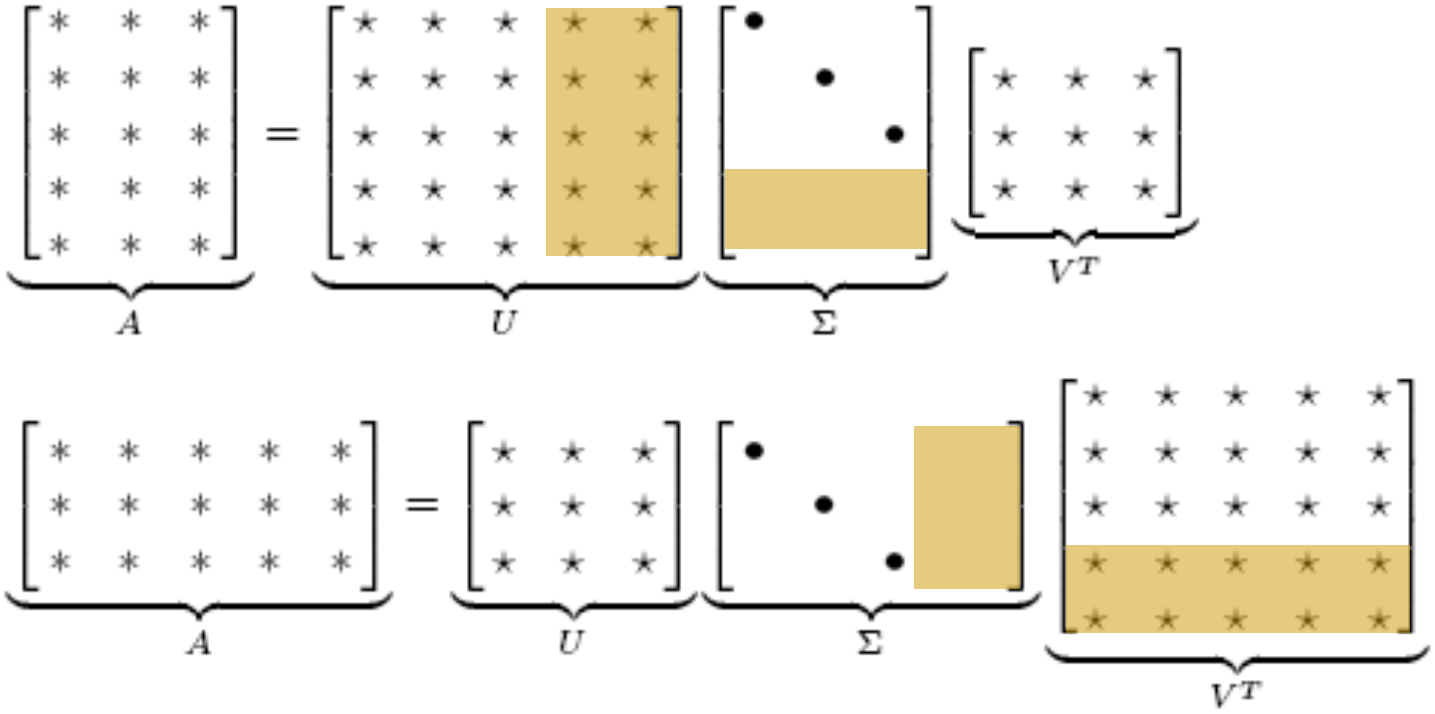
Singular Value Decomposition

- Illustration of SVD dimensions and sparseness



Singular Value Decomposition

- Illustration of SVD dimensions and sparseness

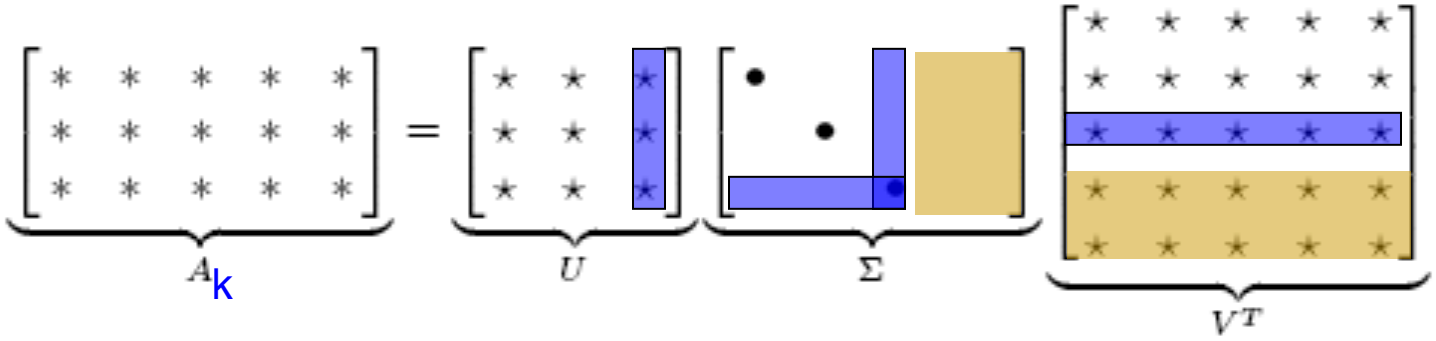


Low-rank Approximation

- SVD can be used to compute optimal **low-rank approximations**.
  - *Approximation problem*: Find $\boldsymbol{A_k}$ of rank $\boldsymbol{k}$ such that
    - $A_k = \text{argmin}_{A_k:\, Rank(A_k)=k} \|A - A_k\|$

Low-rank Approximation

- Solution via SVD

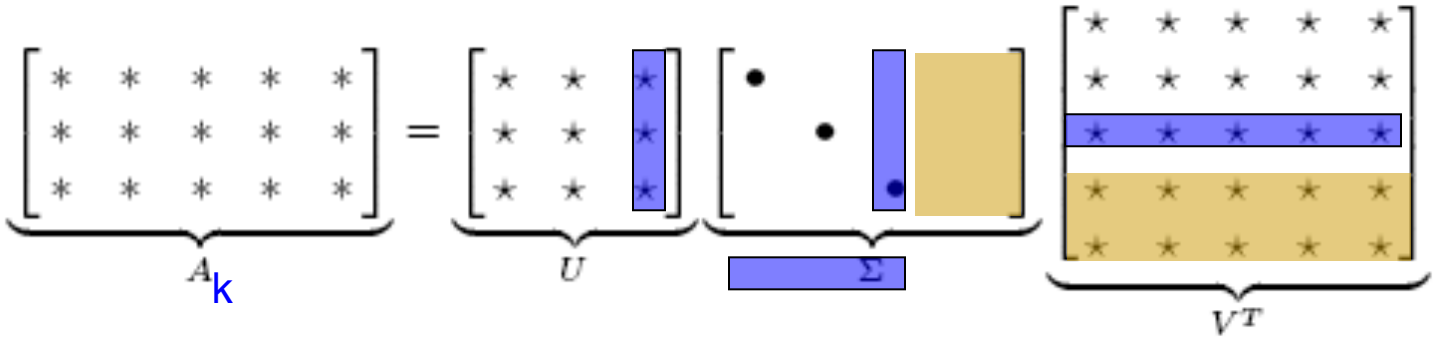


$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

column notation: sum of rank 1 matrices

Reduced SVD

- If we retain only *k* singular values, and set the rest to 0, then we don't need the matrix parts in blue
- Then Σ is *k*×*k*, *U* is *M*×*k*, V^T is *k*×*N*, and A_k is *M*×*N*
- This is referred to as the *reduced SVD*
  - It is the convenient (space-saving) and usual form for computational applications



SVD Low-rank approximation

- Whereas the term-doc matrix A may have M=50000, N=10 million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
  - Of all rank 100 matrices, it would have the lowest Frobenius error.

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

- $C : m \times n$ Document-Term Incidence Matrix
- $C = U \Sigma V^T$
- $C \approx C_k = U_k \Sigma_k V_k^T$

- Closeness among documents can be computed using the entries in U_k
  - We do not need the complete n-dimensional representation of the documents
  - We can work with $k \ll n$ dimensions.
- Closeness among terms can be computed using the entries in V_k
  - We do not need the complete m-dimensional representation of the terms
  - We can work with $k \ll m$ dimensions.

SVD Low-rank approximation

- Whereas the term-doc matrix *A* may have *M*=50000, *N*=10 million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
  - Of all rank 100 matrices, it would have the lowest Frobenius error.

- Great … but why would we??
- Answer: *Latent Semantic Indexing*

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

What it is

- From term-doc matrix A, we compute the approximation $A_{k.}$
- There is a row for each term and a column for each doc in A_k
- Thus docs live in a space of $k<<r$ dimensions
  - These dimensions are not the original axes
- But why?

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 severe in very heterogeneous collections)*.
  - The vector space model is unable to discriminate between different meanings of the same word.

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

Problems with Lexical Semantics

- **Synonymy**: Different terms may have **identical or similar meanings** (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 con

Latent Semantic Indexing (LSI)

- 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 this **latent semantic space**

Goals of LSI

- Similar terms map to similar location in low dimensional space
- Noise reduction by dimension reduction

Technical Memo Example

Titles:

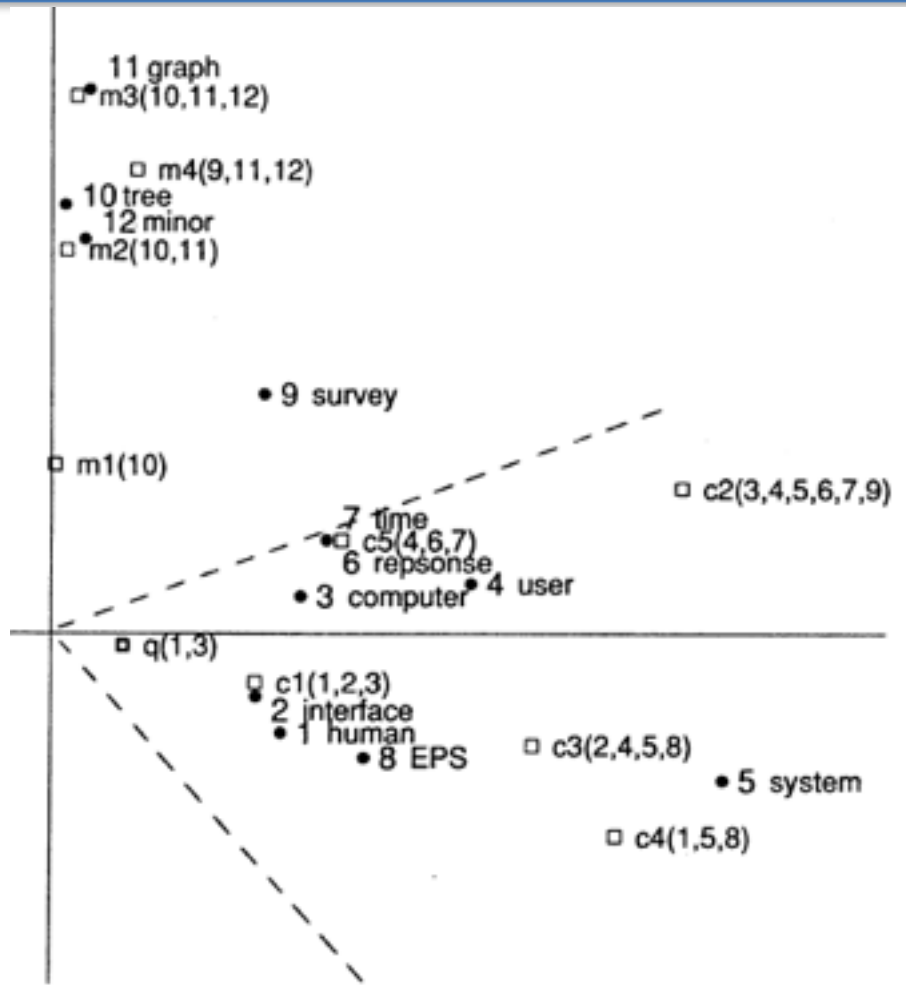
c1: *Human* machine *interface* for Lab ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Ref: Deerwester, Scott, et al. "Indexing by latent semantic analysis." *Journal of the American society for information science* 41.6 (1990)

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

Ref: Deerwester, Scott, et al. "Indexing by latent semantic analysis." *Journal of the American society for information science* 41.6 (1990)



Ref: Deerwester, Scott, et al. "Indexing by latent semantic analysis." *Journal of the American society for information science* 41.6 (1990)

Does it work?

- $C : m \times n$ Document-Term Incidence Matrix
- $C = U \Sigma V^T$
- $C \approx C_k = U_k \Sigma_k V_k^T$

- Closeness among documents can be computed using the entries in U_k
  - We do not need the complete n-dimensional representation of the documents
  - We can work with $k \ll n$ dimensions.
- Closeness among terms can be computed using the entries in V_k
  - We do not need the complete m-dimensional representation of the terms
  - We can work with $k \ll m$ dimensions.

LSA on Earthquake Tweet (SMERP + FIRE) dataset

Top terms per cluster:

- **Cluster 0:** death, toll, italy, rises, quake, earthquake, climbs, injured, central, least, italian, dead, hits, italyearthquake,
- **Cluster 1:** nepal, food, earthquake, water, victims, need, tents, send, help, kathmandu, people, packets, please, shelter, nepalearthquake, destroyed, damaged,
- **Cluster 2:** relief, medical, nepal, team, earthquake, rescue, teams, doctors, victims, aid, sending, hospital, materials, material, nepalearthquake, india, fund, disaster,
- **Cluster 3:** dead, earthquake, quake, central, injured, damage, people, least, killed, casualties, magnitude, amatrice, buildings, damaged, many, news, reports,
- **Cluster 4:** donate, red, cross, italian, earthquake, help, victims, please, wifi, passwords, italy, blood, money, disable, italians, fi, wi, people,

LSA on BBC Dataset

Top terms per cluster:

- **Cluster 0:** mobile, phone, broadband, digital, people, technology, phones, tv, bt, said, music, service, video, mobiles, services,
- **Cluster 1:** show, tv, said, series, star, musical, bbc, us, film, comedy, book, channel, best, new, television,
- **Cluster 2:** economy, growth, economic, dollar, said, rate, rates, us, year, bank, figures, prices, 2004, exports, spending,
- **Cluster 3:** lord, lords, said, blunkett, blair, home, government, secretary, law, house, rights, terror, would, clarke, human,

LSA on BBC Dataset (Contd.)

- **Cluster 4:** games, game, software, said, microsoft, users, people, computer, search, virus, online, security, net, mail, information,
- **Cluster 5:** music, band, album, rock, song, best, chart, number, singer, said, one, top, us, awards, tour,
- **Cluster 6:** said, government, would, eu, people, uk, party, minister, public, police, could, also, plans, new, local,
- **Cluster 7:** film, best, films, oscar, festival, awards, actor, award, director, actress, aviator, box, movie, star, year,
- **Cluster 8:** said, company, shares, firm, us, oil, market, sales, profits, bank, year, yukos, deal, stock, euros,
- **Cluster 9:** labour, election, blair, brown, party, howard, tax, chancellor, said, tory, prime, would, minister, tories, campaign,

LSA on NIPS Papers

Top terms per cluster:

- **Cluster 0:** model, data, posterior, distribution, variational, models, latent, gaussian, bayesian, inference, log, likelihood, prior, sampling,
- **Cluster 1:** data, model, learning, training, set, using, algorithm, one, two, classification, models, features, time, figure,
- **Cluster 2:** neurons, neuron, spike, synaptic, cells, stimulus, model, firing, cell, network, activity, input, time, circuit, neural,
- **Cluster 3:** network, networks, units, training, neural, learning, input, layer, hidden, output, weights, error, function, time, unit,

LSA on NIPS Papers (Contd.)

- **Cluster 4:** policy, state, reward, action, learning, agent, reinforcement, actions, mdp, policies, function, value, states, optimal, algorithm,
- **Cluster 5:** image, images, object, model, features, visual, objects, training, recognition, learning, network, feature, layer, segmentation, using,
- **Cluster 6:** graph, tree, nodes, algorithm, node, clustering, graphs, submodular, set, data, model, xi, edge, cluster, edges,
- **Cluster 7:** kernel, kernels, svm, data, xi, learning, training, function, set, matrix, classification, space, feature, problem, yi,
- **Cluster 8:** matrix, rank, sparse, algorithm, norm, lasso, convex, problem, data, log, theorem, matrices, recovery, tensor,
- **Cluster 9:** regret, algorithm, loss, learning, bound, convex, theorem, xt, log, bounds, function, online, risk, let, algorithms,

LSA on BBC Tech news

Top terms per cluster:

- **Cluster 0:** patent, patents, eu, directive, software, european, inventions, patenting, bill, source, ministers, draft, open, legal, council, critics, computer,
- **Cluster 1:** people, computer, technology, world, us, used, blog, says, radio, could, blogs, make, information, users, domain, music, would, use,
- **Cluster 2:** search, google, microsoft, yahoo, desktop, web, jeeves, firefox, engine, browser, results, information, users, msn, blogs,
- **Cluster 3:** sony, digital, music, dvd, show, definition, gadget, gadgets, portable, high, hd, devices, ipod, tv, content, technologies, consumer, players, video,

LSA on BBC Tech news

- **Cluster 4:** games, game, nintendo, gaming, sony, gamers, console, titles, xbox, playstation, play, ea, halo, handheld, cell, chip, psp,
- **Cluster 5:** mobile, phone, phones, mobiles, 3g, people, services, camera, handsets, multimedia, music, operators, tv, vodafone, technology, mda, cameras,
- **Cluster 6:** broadband, bt, tv, uk, net, people, service, million, internet, digital, services, online, fast, speeds, connections, cable, customers, access,
- **Cluster 7:** virus, security, spam, spyware, mail, attacks, microsoft, anti, software, windows, viruses, programs, users, mails, malicious, program, net, infected,
- **Cluster 8:** peer, apple, file, bittorrent, legal, music, files, sharing, piracy, p2p, court, system, mpaa, industry, action, networks, software, movie, filed,
- **Cluster 9:** china, chinese, net, cafes, government, news, people, makover, google, country, local, foreign, programmes, pc, online, use, media,

Applicability

- Text Mining
  - Information Retrieval
  - User modeling
  - …

- Starting point:
  - Create the initial matrix C
  - C cascades the representations of all documents together

Exercise: Vector to text re-construction

Election campaign

Election	campaign	Election campaign
1	1	1

Thank You