Information retrieval

Flexible querying systems and ranking systems

Clovis Galiez

Laboratoire Jean Kuntzmann, Statistiques pour les sciences du Vivant et de l'Homme

December 8, 2020

Objectives of the course

- Acquire a culture in information retrieval
- Master the basics concepts allowing to understand:
  - what is at stake in novel IR methods
  - what are the technical limits

This will allow you to have the basics tools to analyze current limitations or lacks, and imagine novel solutions.

Today's outline

- Short summary of last lecture
- tf-idf
- Querying in the vector-space model
- Latent semantics
- Ranking

What to remember from last time?

Remember...

What are the main points you remember from last lecture?

What to remember from last time?

Remember...

What are the main points you remember from last lecture?

- Web IR is split in distinct steps:
  - Gathering and indexing data from the web (**crawling**)
  - Retrieving documents relevant to a query
  - Ranking the valid answers according to relevance
- The involved data is **big**

  Need efficient representation and algorithms

Drawbacks of the boolean querying systems

Can you list some drawbacks?

Drawbacks of the boolean querying systems

Can you list some drawbacks?

The boolean queries are not flexible

Query: `result elections United States`
Doc title: "White House election: live results!"

Drawbacks of the boolean querying systems

Can you list some drawbacks?

The boolean queries are not flexible

Query: `result elections United States`
Doc title: "White House election: live results!"

With a good stemming and tokenization, we will match `result` and `election`... we miss the match between `United States` and `White House` :-/

Drawbacks of the boolean querying systems

Can you list some drawbacks?

The boolean queries are not flexible

Query: `result elections United States`
Doc title: "White House election: live results!"

With a good stemming and tokenization, we will match `result` and `election`... we miss the match between `United States` and `White House` :-/

The boolean querying does not rank

When querying using a boolean querying system, the output is binary.
$\rightarrow$Unable to distinguish the relevant matches from non-relevant ones.

The vector space model and the latent semantics

Representing documents as vectors in $\mathbb{R}^T$

From binary presence/absence...

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...

Representing documents as vectors in $\mathbb{R}^T$

...to real vector space.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	0.01	0.02	0	0	0.006	...
doc 2	0	0.013	0.001	0	0.001	...
doc 3	0.0031	0.008	0.0043	0	0.0021	...
...	...	...	...	...	...	...

What numbers can be useful here ?

Not every term is informative

How do you quantify information according to Shannon theory?

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(e) = f(P(e))$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(e) = f(P(e))$
- it should be contravariant with the probability:

$$P(e_1) < P(e_2) \Rightarrow I(e_1) > I(e_2)$$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(e) = f(P(e))$
- it should be contravariant with the probability:

$$P(e_1) < P(e_2) \Rightarrow I(e_1) > I(e_2)$$

- when e_1 and e_2 are independent, we would like that:

$$I(e_1 \& e_2) = I(e_1) + I(e_2)$$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(e) = f(P(e))$
- it should be contravariant with the probability:

$$P(e_1) < P(e_2) \Rightarrow I(e_1) > I(e_2)$$

- when e_1 and e_2 are independent, we would like that:

$$I(e_1 \& e_2) = I(e_1) + I(e_2)$$

If we moreover ask for f to be continuous and non-zero, there is only one possible class of functions: $-log_b$

Information

The information of an event e is defined as $I(e) = -log(P(e))$

Definition

We can now compute the information of a token as:

$$I(t) = -\log(\frac{\#\text{doc including token } t}{\#\text{docs}})$$

Information

The information of an event e is defined as $I(e) = -log(P(e))$

Definition

We can now compute the information of a token as:

$$I(t) = -\log(\frac{\text{\#doc including token } t}{\text{\#docs}})$$

Exercise

I throw a die. What is the more informative:

- the outcome is even
- the outcome is ≥ 5

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

$||\vec{D}||_1$ carries the total information carried by a document:

- low if
- average if
- high if

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

$||\vec{D}||_1$ carries the total information carried by a document:

- low if the document contains only common tokens
- average if the document contains few exceptional tokens
- high if the document contains only exceptional items

The tf-idf matrix

Definition

The matrix M which rows – corresponding to each document – are:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

is called the **tf-idf** (term frequency-inverse document frequency) representation.

The tf-idf matrix

Definition

The matrix M which rows – corresponding to each document – are:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

is called the **tf-idf** (term frequency-inverse document frequency) representation.

Question

What is the unit of elements of the tf-idf matrix?

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query?

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query? Naïve approach: dot product.

Indeed, it makes sense: For each document, compute:

$$\vec{D} \cdot \vec{Q} = \sum_t D_t . Q_t$$

The higher the dot product, the more informative tokens $\vec{Q}$ and $\vec{D}$ share... and the more relevant should be the D with respect to the query Q.

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query? Naïve approach: dot product.

Indeed, it makes sense: For each document, compute:

$$\vec{D} \cdot \vec{Q} = \sum_t D_t.Q_t$$

The higher the dot product, the more informative tokens $\vec{Q}$ and $\vec{D}$ share... and the more relevant should be the D with respect to the query Q.

Exercise

Code this scalar product in an efficient way!

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\#\text{ t in Q}}{\#\text{ tokens in Q}} \times I(t)$$

How to retrieve documents related to the query? Naïve approach: dot product.

Indeed, it makes sense: For each document, compute:

$$\vec{D} \cdot \vec{Q} = \sum_t D_t . Q_t$$

The higher the dot product, the more informative tokens $\vec{Q}$ and $\vec{D}$ share... and the more relevant should be the D with respect to the query Q.

Exercise

Code this scalar product in an efficient way!

For querying purposes, one can select documents such that $\vec{D} \cdot \vec{Q} > \tau$, but it can directly be used for ranking documents.

Correcting for cheaters

Problem

Imagine a way of cheating with this approach.

Correcting for cheaters

Problem

Imagine a way of cheating with this approach.

Content farms

$$\begin{aligned}\vec{D}\cdot\vec{Q} &= \sum_t D_t.Q_t \\ &= \sum_t \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t).\frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t) \\ &\propto \frac{1}{\text{\# tokens in D}} \sum_t \text{\#t in D} \times \text{\#t in Q} \times I(t)^2\end{aligned}$$

Correcting for cheaters

Problem

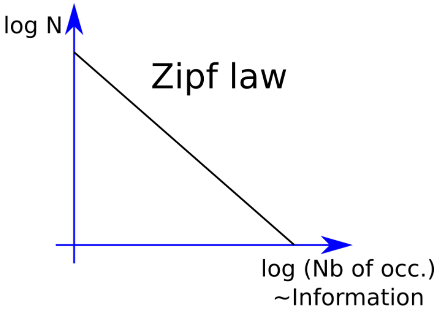
Imagine a way of cheating with this approach.

Content farms

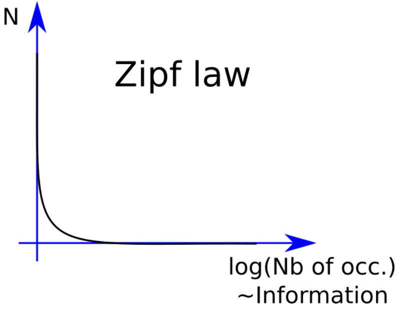
$$\begin{array}{rcl}\vec{D}\cdot\vec{Q} & = & \sum_t D_t.Q_t \\ & = & \sum_t \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t).\frac{\#\text{ t in Q}}{\#\text{ tokens in Q}} \times I(t) \\ & \propto & \frac{1}{\#\text{ tokens in D}} \sum_t \#\text{t in D} \times \#\text{t in Q} \times I(t)^2\end{array}$$

Documents containing many informative words will be selected and ranked first.

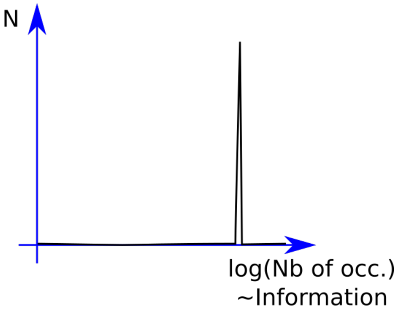
Content farms: pull informative words together



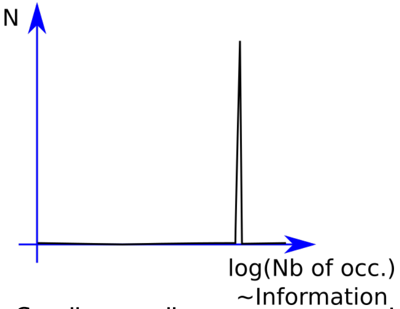
Content farms: pull informative words together



Content farms: pull informative words together



Content farms: pull informative words together



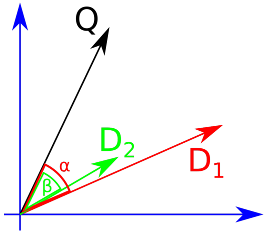
...Can "answer" many more queries

The cosine similarity

How could you correct for content farms cheats?

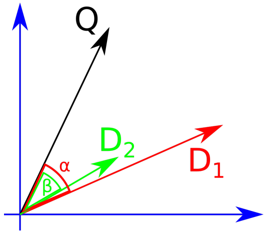
The cosine similarity

How could you correct for content farms cheats?



The cosine similarity

How could you correct for content farms cheats?



Correct by normalizing the similarity:

Consine similarity

$$\text{cosim}(\vec{D}, \vec{Q}) = \frac{\vec{D} \cdot \vec{Q}}{||\vec{D}||_2 . ||\vec{Q}||_2}$$

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

As already pointed out, we could use a semantic approach (ontologies), but need a fixed and manually curated work.

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

As already pointed out, we could use a semantic approach (ontologies), but need a fixed and manually curated work.
Can we work directly from the data?

Latent semantics

Special structure of the data: correlations

In practice a tf matrix looks like:

Interlude

Video

Special structure of the data: correlations

In practice a tf matrix looks like:

Interlude

Video

We observe...

A block structure.

Special structure of the data: correlations

In practice a tf matrix looks like:

Interlude

Video

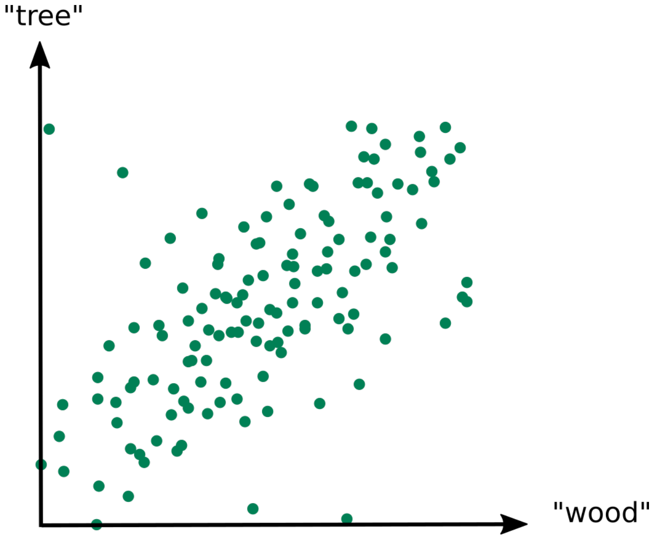
We observe...

A block structure.

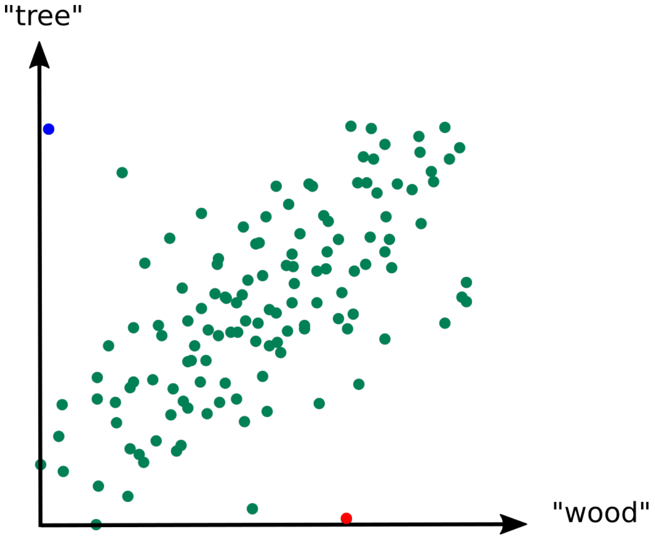
We observe...

How to recover automatically those blocks ?

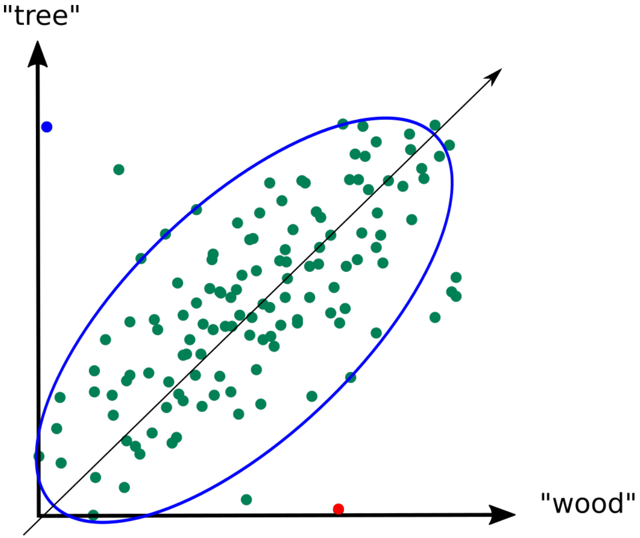
Reminders from linear algebra



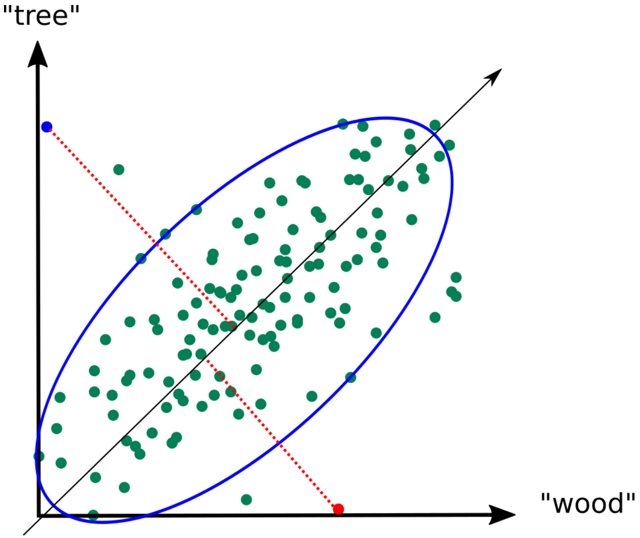
Reminders from linear algebra



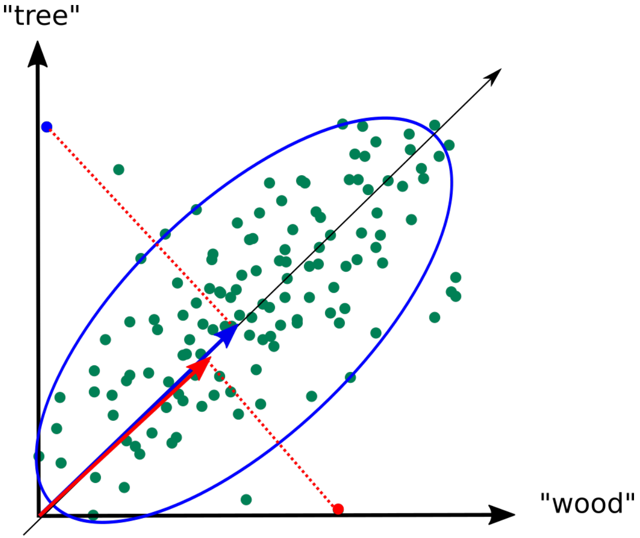
Reminders from linear algebra



Reminders from linear algebra



Reminders from linear algebra



Low rank approximation

Theorem (Eckart–Young–Mirsky)

The best[a] r-rank approximation $\hat{M}$ of M is given by the projection on the subspace formed by the eigenvectors of $M^\top M$ corresponding to the r biggest eigen values.

[a] In the sense minimizing $||M - \hat{M}||_F = \sum_{i,j}(m_{i,j} - \hat{m}_{i,j})^2$

The projection to the low rank space (columns of $V^\top$ in SVD decomposition $M = U\Sigma V^\top$) collapse similar (i.e. *correlated*) tokens to the same component. This space is called the **Latent semantic space**.

Algebra theorem

Eigenvectors of $M^\top M$, $\vec{C}_i$ are orthogonal and form a basis of the token space.
We can define a new scalar product:

$$\vec{D'} = \sum \alpha_i \vec{C}_i$$
$$\vec{Q'} = \sum \beta_i \vec{C}_i$$

We can compare search documents matching query Q using $\vec{D'}.\vec{Q'} = \sum \alpha_i.\beta_i$ or cosim$(\vec{D'}, \vec{Q'})$:)

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

If D is a binary vector over documents, what does $M^\top D$ represent?

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

If D is a binary vector over documents, what does $M^\top D$ represent?

The cumulated frequencies of each token in the corpus D.

If Q a binary vector over tokens, what does $M^\top MQ$ represent?

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

If D is a binary vector over documents, what does $M^\top D$ represent?

The cumulated frequencies of each token in the corpus D.

If Q a binary vector over tokens, what does $M^\top MQ$ represent?

The cumulated frequencies of tokens in the (virtual) corpus matching Q.

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.
If Q a binary vector over tokens, what does $M^\top MQ$ represent?
The cumulated frequencies of tokens in the (virtual) corpus matching Q.

What does it mean that $M^\top MQ = \lambda.Q$?

Interpretation of the correlation matrix

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.
If Q a binary vector over tokens, what does $M^\top MQ$ represent?
The cumulated frequencies of tokens in the (virtual) corpus matching Q.

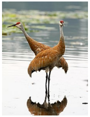
What does it mean that $M^\top MQ = \lambda.Q$? What if λ is small? big?

Vector model: bright and dark side

The tf-idf vector model is good...

- Similarity based on information carried by tokens
- Flexible querying (latent semantics)
- Naturally rank documents
- Works well in practice

...but still not perfect:

- ignore polysemy  vs. 
- ignore the *truth* of the information 

Information function is unique up to a $\times$ constant

Let $a \in \mathbb{R}_+$ and $p \in \mathbb{N}$.
$f(a) = f(a^{\frac{q}{q}}) = f((a^{\frac{1}{q}})^q) = q.f(a^{\frac{1}{q}})$.
So for any $p, q \in \mathbb{N}$,

$$f(a^{\frac{p}{q}}) = \frac{p}{q} f(a)$$

By density of $\mathbb{Q}$ in $\mathbb{R}$ and continuity of f, $f(a^x) = x.f(a)$.
If $f \neq 0$, there is a b such that $f(b) = 1$, so that $\forall x \in \mathbb{R}_+, f(b^x) = x$ so that $f = \log_b$

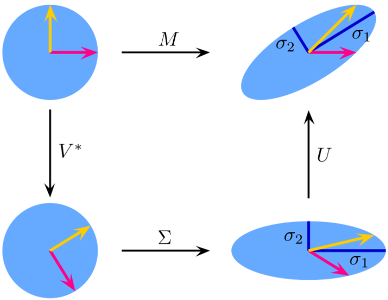
$||R_i||_1 < ||R_{i+1}||_1$ comes from sinks

If $\forall j$ there exists at least a page i and a link $j \rightarrow i$, then:

$$\begin{aligned} ||R_{i+1}||_1 &= ||A.R_i||_1 \\ &= \sum_i \sum_{j \rightarrow i} \frac{R_j}{N_j} \\ &= \ldots \\ &= 1 \end{aligned}$$

Reminders from linear algebra

We can decompose a matrix as a composition of orthogonal operation, scaling and again orthogonal operation.



$$M = U \cdot \Sigma \cdot V^*$$

This decomposition is coined the Singular Value Decomposition (SVD).

Low rank approximation of the tf-idf matrix

Eckart-Young-Mirsky Theorem

Let $M \in \mathbb{R}^{d \times t}, t < d$. If $M = U \Sigma V^\top$ is the SVD decomposition of M with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_t$, then the best[a] r-rank approximation of M is $(r < t)$:

$$\hat{M} := U_r \Sigma_{r,r} V_r^\top$$

where X_r is the restriction of X to the first r columns, and $\Sigma_{r,r}$ to the first r lines and columns.

[a] In the sense minimizing $||M - \hat{M}||_F = \sum_{i,j} (m_{i,j} - \hat{m}_{i,j})^2$