Information retrieval

Flexible querying systems and ranking systems

Clovis Galiez

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

December 9, 2019

Today's outline

- Short summary of last lecture
- 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 ?

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.

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

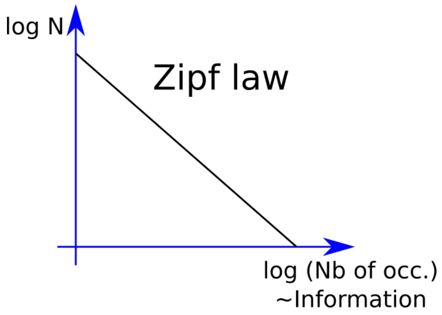
Imagine a way of cheating with this approach.

Content farms

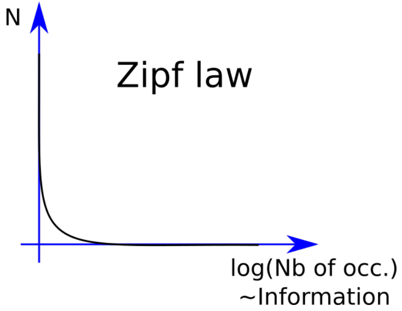
$$\begin{aligned}\vec{D}\cdot\vec{Q} &= \sum_t D_t.Q_t \\ &= \sum_t \frac{\#\text{ t in D}}{\#\text{ tokens in D}}\times I(t).\frac{\#\text{ t in Q}}{\#\text{ tokens in Q}}\times I(t) \\ &\propto \frac{1}{\#\text{ tokens in D}}\sum_t \#\text{t in D}\times\#\text{t in Q}\times I(t)^2\end{aligned}$$

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

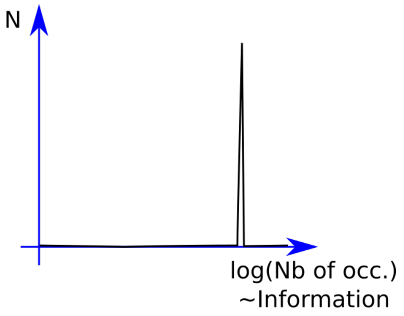
Content farms: pull informative words together



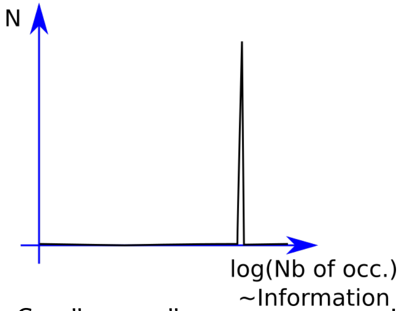
Content farms: pull informative words together



Content farms: pull informative words together



Content farms: pull informative words together



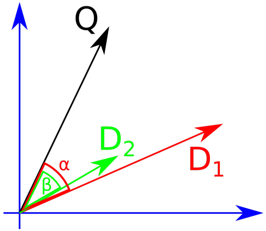
...Can "answer" many more queries

The cosine similarity

How could you correct for content farms cheats?

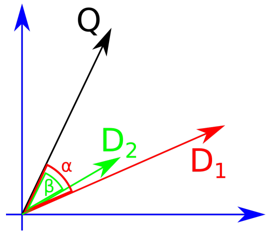
The cosine similarity

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 look like:

Interlude

Video

Special structure of the data: correlations

In practice a tf matrix look like:

Interlude

Video

We observe...

A block structure.

Reminders from linear algebra

Exercise

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

Reminders from linear algebra

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.

Reminders from linear algebra

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?

Reminders from linear algebra

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.

Reminders from linear algebra

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?

Reminders from linear algebra

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.

Reminders from linear algebra

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 $M^\top MQ = \lambda.Q$?

Reminders from linear algebra

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?

Algebra theorem

Theorem

$M^\top M$ is symmetric and its eigenvectors $\vec{C}_i$ are orthogonal and form a basis of the token space.

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 $||M - \hat{M}||_F = \sum_{i,j}(m_{i,j} - \hat{m}_{i,j})^2$

Low rank approximation

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**.

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

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

Machine learning in IR

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent **concepts**.

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent **concepts**.
We represent documents by projecting their

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent **concepts**.
We represent documents by projecting their frequency vector in the low dimensional space formed by the important

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent **concepts**.
We represent documents by projecting their frequency vector in the low dimensional space formed by the important **eigen vectors**.

From (linear) latent semantics to embeddings

In latent semantics, we define base vectors of the vector space of token frequencies that represent **concepts**.
We represent documents by projecting their frequency vector in the low dimensional space formed by the important **eigen vectors**.

Recent techniques (well, mostly since 2013)

Machine learning techniques can be used to **learn better vector representation**[a] **of tokens**, and more generally of any data (document, sentence, word, image, etc.).

[a] aka embeddings

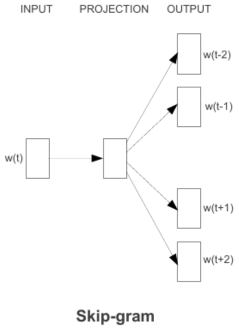
Embeddings: a general technique with many derivatives

Many models have been developed for representing various type of data. Here is a small list of freely available models:

Model	Data represented
`word2vec`	Tokens
`GloVe`	Tokens
`fastText`	Tokens
`doc2vec`	Documents
`dna2vec`	Genomic sequences

Word2vec: predict the context of a token

The core idea of word2vec is to learn a vector representation allows to predict the context of the token. Thereby, tokens appearing in similar context will be encoded closely in the vector space.

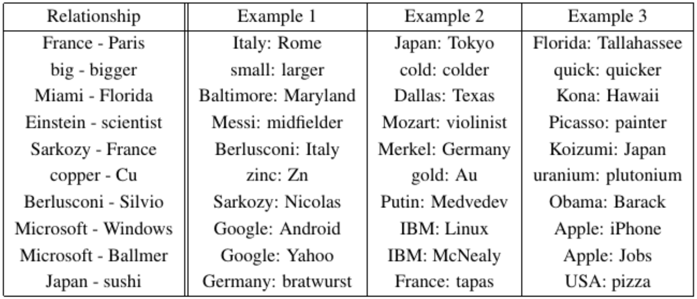


[Mikolov, Tomas; et al. (2013)]

word2vec's latent semantics

The word2vec embeddings have interesting semantic features[1].

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*



Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

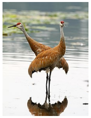
[1]Note that `GloVe` is better at this

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

Dealing with the *truth*

How to deal with the truth?

It is almost impossible to deal with truth judgment only from the document data.

However, we can assume that we trust information coming from *authorities* (well-known newspaper, official website, etc.).

How to deal with the truth?

It is almost impossible to deal with truth judgment only from the document data.

However, we can assume that we trust information coming from *authorities* (well-known newspaper, official website, etc.).

Idea

Rank the results of the querying system according to their authority.

How do we know who is the authority ?

How to deal with the truth?

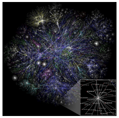
It is almost impossible to deal with truth judgment only from the document data.

However, we can assume that we trust information coming from *authorities* (well-known newspaper, official website, etc.).

Idea

Rank the results of the querying system according to their authority.

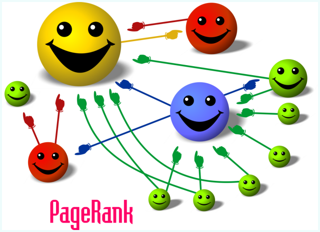
How do we know who is the authority ?

$\rightarrow$ We extract it from the web structure  !

Authority and web structure

Who is the authority?

If you only represent the web by a graph where each node is a web page and each directed edge is an HTML link.

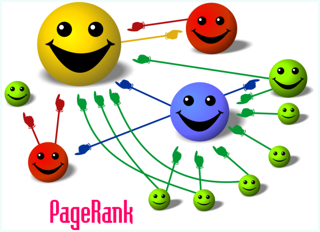


How would you recognize an authority?

Authority and web structure

Who is the authority?

If you only represent the web by a graph where each node is a web page and each directed edge is an HTML link.

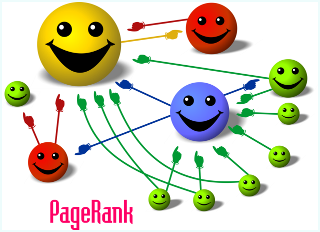


How would you recognize an authority?
The authority is higher when a node is pointed at (by other authorities).

Authority and web structure

Who is the authority?

If you only represent the web by a graph where each node is a web page and each directed edge is an HTML link.



How would you recognize an authority?
The authority is higher when a node is pointed at (by other authorities).

Imagine an algorithm able to detect/rank authorities.

PageRank formalization (simple version)

Random surfer model

Imagine a user having the following behavior clicking on random links on the Internet.

The more links leading to a page, the more chance (and the more times) the user visits the page.

PageRank formalization (simple version)

Random surfer model

Imagine a user having the following behavior clicking on random links on the Internet.

The more links leading to a page, the more chance (and the more times) the user visits the page.
After a loooong time, we measure the average number of times the user visited a given page P, we denote R_P.

Definition of the rank according to PageRank

We define the authority/ranking of a page by the R_P value.

PageRank algorithm (simple version)

```
**Data:** $A :=$ graph of the WWW   $A_{ij} = \begin{cases} \frac{1}{N_j} & \text{if link from } j \text{ to } i \\ 0 & \text{else} \end{cases}$
**Result:** Ranking of web pages
$R_0 := S$ ;
**repeat**
    $R^{(i+1)} \leftarrow AR^{(i)}$
    $\delta \leftarrow ||R^{(i)} - R^{(i+1)}||_1$
**until** $\delta \leq \epsilon$;
```

Algorithm 1: simplified PageRank

Milestone of Google (algo designed by L. Page, Google co-founder), and drove the initial success of Google.

PageRank without sink effect

Sink effect

What if a page does not have any outgoing connection?

It will "trap" the user and have an artificially high rank.

PageRank without sink effect

Sink effect

What if a page does not have any outgoing connection?

It will "trap" the user and have an artificially high rank.

The random eager surfer

Imagine the user having now the following behavior[a]

- click on a random link on the current web page with probability $p(t)$
- or jump to a random web page on the Internet with probability $1 - p(t)$

[a] In the original paper by Page, the balance between the two events is given by its trap feeling: the more trapped it gets, the more likely the user will jump somewhere else.

Full PageRank

To avoid a *sink* effect, we introduce random jumps to a set of pages encoded in E.

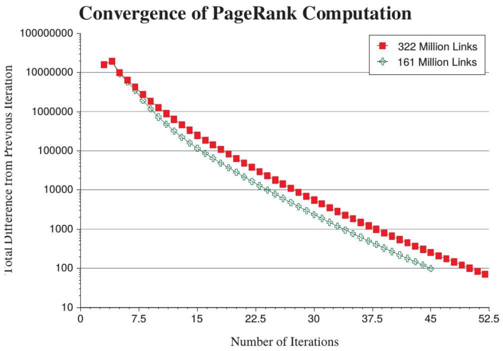
Data: Graph of the WWW

Result: Ranking of web pages

```
$R_0 := S$ ;
repeat
    $R_{i+1} \leftarrow AR_i$
    $d \leftarrow ||R_i||_1 - ||R_{i+1}||_1$
    $R_{i+1} \leftarrow R_{i+1} + d.E$
    $\delta \leftarrow ||R_i - R_{i+1}||_1$
until $\delta \leq \epsilon$;
```

Algorithm 2: PageRank

PageRank convergence



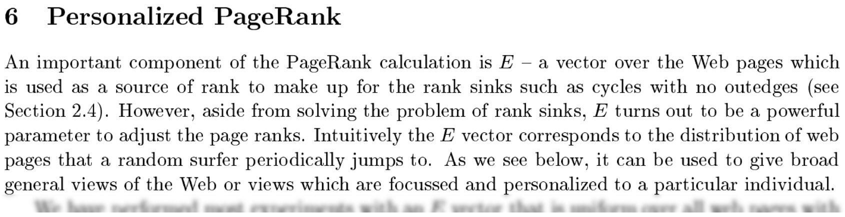
[L. Page, 98]

Full PageRank

Note that the vector E encodes the distribution of pages where the user is willing to jump to.

Full PageRank

Note that the vector E encodes the distribution of pages where the user is willing to jump to.

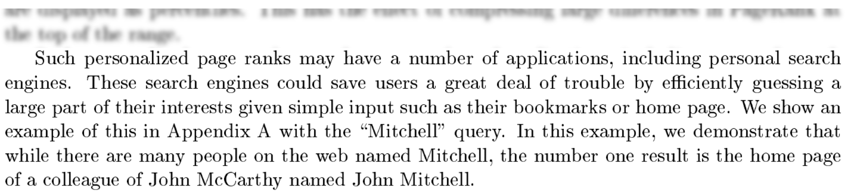


6 Personalized PageRank

An important component of the PageRank calculation is E – a vector over the Web pages which is used as a source of rank to make up for the rank sinks such as cycles with no outedges (see Section 2.4). However, aside from solving the problem of rank sinks, E turns out to be a powerful parameter to adjust the page ranks. Intuitively the E vector corresponds to the distribution of web pages that a random surfer periodically jumps to. As we see below, it can be used to give broad general views of the Web or views which are focussed and personalized to a particular individual.

We have performed most experiments with an E vector that is uniform over all web pages with

...



are displayed as percentiles. This has the effect of compressing large differences in PageRank at the top of the range.

Such personalized page ranks may have a number of applications, including personal search engines. These search engines could save users a great deal of trouble by efficiently guessing a large part of their interests given simple input such as their bookmarks or home page. We show an example of this in Appendix A with the "Mitchell" query. In this example, we demonstrate that while there are many people on the web named Mitchell, the number one result is the home page of a colleague of John McCarthy named John Mitchell.

Summary

- Tf-Idf vector representation of a document
- Flexible vector queries (cosine similarity)
- Latent semantics (lower rank projection of the tf matrix)
- PageRank

Next lectures: can we make it?

- Machine learning in IR
- TP (Implemenation and experiments around IR systems)
  - Tokenizer
  - Tf-Idf matrix construction
  - Page Rank implementation
  - Mini-search engine

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 $f = \log_b$

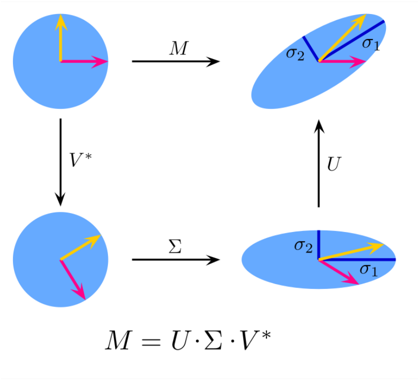
$||R_i||_1 < ||R_{i+1}||_1$ comes from sinks

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.



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 $||M - \hat{M}||_F = \sum_{i,j}(m_{i,j} - \hat{m}_{i,j})^2$