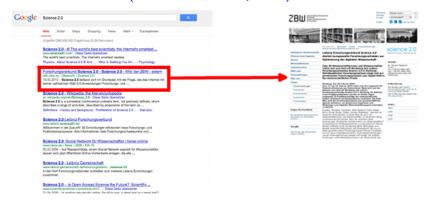# A Contribution to Rating and Recommendation Systems: Concepts, Development and Evaluation

Oliver Diestel
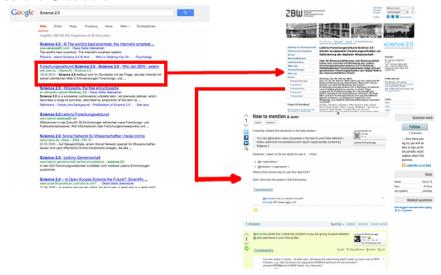
February 27, 2013

Current Situation (Just an assumption)

User visits the zbw.eu website searchs for the information that he would like to have. Leaves the website.

Possible Situation

User visits the website, sees an interesting question from another user, reads the question and the answer, asks hopefully more questions. Goal increase the user interaction on a website.

q/A System

General Idea: Use a Question/Answer System(Picture) to make the information retrieval process public. Use this public questions and answers to increase the user interaction on the website.

Beginning

What do we have at the beginning? Website, User and Information(Question)
After every new concept answer the question what do we have now

Rating

- Click
- active time on page

If a user stays shorter or longer on a page than the average user - it is a high or low rating.
We rate QA pages and webpages, however we only recommend qa pages.
Now we have Qustion Answer System, User and the information which user likes which question as well as our own website.

Tagging

- Standard-Thesaurus Wirtschaft, tripple store
- Stemming: Compute the root of a word.
- Computing Levenshtein distance: Calculate the distance between two words.

Principle

Every question will be tagged with a prefered label from the ZBW Thesaurus. For every word in the question find the root of the word and match it against the thesaurus. If we find a matching prefered label, Done. If we find a matching synonym take the preferd label as tag, Done.

What do we have now? Information about which user likes which question and one or more descriptor that describes the subject of the question

Input/Output

Input: ⟨user_id, item_id, rating⟩, [List of tags], number of recommendations
Output: Sorted list of Items with avg. rating that correspond with the pref. input label

- ▶ Item: id, name, tag
- ▶ User: id, name

Question: How does this black box work? Three part recommendation. Pipeline architecture

- ▶ Calculate similarity between items
- ▶ Add similar ratings to a user/rating vector.
- ▶ Calculate singular value decomposition
- ▶ Find similar users

Item-Based Algorithm

Display a matrix with five users and five objects not every user has a rating for every object To find similar users we need as much user/item ratings as possible so we would like to guess as much ratings as possible.

Find items that have similar ratings. Calculate the similarity values for every item.

Cosinus Similarity

$$sim(\overrightarrow{a}, \overrightarrow{b}) = \frac{\overrightarrow{a} * \overrightarrow{b}}{|\overrightarrow{a}| * |\overrightarrow{b}|}$$

Take the differences of the average rating behaviour of the user into account.

Advanced Cosinus Similarity

$$sim(a, b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Calculate predictions for similar items.

User u, Product p $pred(u, p) = \frac{\sum_{i \in ratedItems(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItems(u)} sim(i,p)}$

Fill recommendation vector of each user with similar item ratings
What do we have now? Matrix with less empty fields than before

Singular Value Decomposition

Now we would like to find similar users with our matrix
Create a SVD with the matrix $M = U * \Sigma * V^t$

- ▶ U dimension m x m, orthogonal matrix, spans the column space of matrix m
- ▶ Σ dimension m x n, diagonal matrix having only r nonzero entries
- ▶ V dimension n x n, orthogonal matrix, spans the row space of matrix m

Σ Properties

Diagonal entries of Σ have the property, that $\sigma > 0$ and $\sigma_i \geq \sigma_{i+1}$

Calculate cosinus similarity between users
Now we have a two dimensional matrix, so we can calculate the cosinus similarity between our users. Display a graph with the users, show which users are similar display the original matrix next to the graph to validate the result.

Is this a good approach?

We could have calculated the SVD directly out of the orriginal matrix. Would we get a similar result? Is it better if we use the v matrix from the SVD to calculate the item similarity and update the SVD afterwards? Is it better if we just recommend similar items from the items the user already likes?

Evaluation

The test data: I use real data from the zbw econdesk. However we do not have real user ratings for this data. Every data gets a random quality value. 1-5 if a user would rather rate it positive or negative. Furthermore I generate 1000 test users these users will have a rating preferation, so a user might be a person that rates an item more positive or more negative. I will try to evaluate the algorithms with this test data. I might use a movie db as well for this.

Find Questions

Find questions that are hightly rated by similar users that match
the current topic

Display Questions

Add a tag and a div-placeholder on every webpage. Display top questions for the current user on the webpage.

Software Architecture

Scala: Finagle twitter framework

Every part of the software is a service.

The tagger, the rating algorithm and the recommendation algorithm can be used as an individual software.

Pros:

Cons:

Timetable

Start: 18.12.2012
End: 18.06.2012
1st month(18.01): Theory
2nd month(18.02): Theory + Technology
3rd month(18.03: Implementation
4th month(18.04: Implementation + First writings
5th month(18.05): Final thesis
Personal Goal
Technology implemented: 31.03.2012
Diplom Thesis ready: 01.05.2012