Time-Based Recommendations for Lecture Materials

Christoph Hermann
Institute for Computer Science
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 51
79110 Freiburg
Germany
hermann@informatik.uni-freiburg.de

Abstract: In this paper we describe the implementation of a system for improving the utilization of lecture materials through recommendations. We present results of an evaluation of the usage of lecture materials as well as an evaluation of download-statistics. These findings led us to develop a novel approach for a time based recommender system for lecture materials. Our approach to making recommendations is particularly useful for Learning Management Systems or Lecture Archives. It differs significantly from previous work due to the fact, that we are employing the timestamps from usage data to create a similarity measure for items. The algorithm of this recommender is presented. We evaluate our algorithm in comparison with other recommendation algorithms like a Loglikelihood recommender for boolean preferences. We show that our algorithm has a better performance then a Loglikelihood recommender.

Introduction

Since more than 15 years several educational institutions like ours started to explore and utilize lecture recordings to record and distribute their "ex-cathedra" courses. Lecture recordings (Electures) represent the contents of a lecture as a multimedia document, which contains the voice of the presenter, the recorded slides, as well as annotations such as handwritten notes. In 2003 we started the development of a web based lecture archive (the Electures-Portal) for the archival and distribution of our Electures. Using our lecture archive we provide students access to the lecture materials at any time and anywhere and in different file formats. The materials are organised hierarchically following the course structure and the academic calendar. An overview of best-practice issues related to the organisation and distribution of lecture recordings can be found in (Lauer, 2004).

Additionally to perform full-text search in text as well as the audio transcription (Welte, 2005, Hürst, 2006) we further want to re-use the materials in our archive. Since we discovered similar usage patterns in our students' behaviour, we want to recommend materials that similar students preferred to them. This is usually done by employing a recommender system. The summarized results of our evaluation in section three show that students are using the different file formats for different purposes. Therefore we are recommending entire lectures from a course instead of single files. We discovered that the behaviour of our students follows periodical time-based patterns, and this led us to the implementation of a new recommendation algorithm based on time data. This new approach for recommending lecture materials based on the timestamps of previously downloaded materials achieves better performance on our data than other recommender systems based on boolean preferences that does not include the date of the preference. Our approach differs significantly from previous work due to the fact, that we are employing timestamps from usage data to create a similarity measure for items.

The outline of the paper is as follows. In the next section we describe the area of research our work belongs to and reference related works in the fields of recommender systems and web usage analysis. Many researchers have been trying to analyze web usage data to detect similarities in users' behaviour. We summarize their results in exploiting the temporal characteristics of log data and the discovery of periodicities and bursts. Since our work focuses on utilising timestamps in a recommender system we are interested in other recommender systems pursuing similar goals. Therefore we give an overview of the works on multidimensional and recommender systems that exploit the temporal feature of the recommended items. In the following section we then describe a study we conducted with our students that led us to develop a new approach for a time based recommender system that recommends lectures to students based on a time-based similarity measure. We present a new way of calculating the similarity between two items based purely on time data. We describe our novel recommendation algorithm that utilises this measure for making recommendations based on web usage data. Our experiments show, that in the context of a Lecture Archive or a Learning Management System this approach outperforms other recommendation algorithms using boolean preferences. We conclude with an outlook on further work and experiments in this area of research.

Related Work

In our work we want to explore possibilities for making recommendations for lecture materials to students while they are browsing our lecture archive. Those recommendations can be created based on historical usage data.
Most of the research on lecture recordings not directly related to the technical aspects or the didactic settings can be classified in the field of data mining, and more specifically content mining. Data mining is the process of extracting patterns from data. If the data is provided by means of the internet, one also speaks of Web (data) mining. According to Kosala and Blockeel (2000) who gave an excellent overview of the area of Web mining research, Web mining can be divided into three different areas of interest, based on which part of the Web to mine: Web usage mining, Web structure mining and Web content mining. Regarding lecture recordings there has been a lot of research in the field of (Web) content mining: Indexing of object-based data from slides (analysis of word-occurrences, term frequencies and inverse document frequencies), the analysis of n-Grams, the role of colours and fonts as well as positioning information (Hürst, 2005). There is also research on video-based contents (using optical character recognition, slide transitions, display-times, etc.). The indexing of audio-contents (with speech recognition, term spotting, pause detection, etc.) as well as textual meta-data (author information, title, duration, etc.) is also part of the research.
The work of Kosala and Blockeel (2000) gives a good overview about current research in the area of web mining.
According to Srivastava *et al.* (2000), Web usage mining is "the application of data mining techniques to discover usage patterns from Web data, in order to better understand and serve the needs of Web-based applications". They describe several methods of collecting data from different sources which represent the navigation patterns of users, as well as the phases of *pre-processing*, *pattern discovery* and *pattern analysis*. They also summarize several aspects like making dynamic recommendations which is a sub-aspect of *personalisation*. In the following we will give an overview about the research in the areas of Web usage mining and recommendations.

Web usage mining

The following works on the analysis of web search engine query logs describe a similar idea to ours: identifying similar behaviour of users by analysing temporal aspects of usage data.
Beitzel *et al.* (2007) described the temporal analysis of the log file of a search engine to identify topics and categories in users search queries that trend differently than the main query log stream over varying periods. They developed a method called *temporal analysis* to analyse query logs to be able to capture topical trend information over time. Their work emphasised the changing query stream characteristics over the temporal aspect of query logs. During their analysis, they found trends that were stable over time, despite the usual fluctuation in the query volume. They concluded that the results for the trend information can be used to assist in query disambiguation and query routing, which allows the search system to redirect queries to a specialised backend database.
Zhao *et al.* (2006) presented a framework exploiting the temporal characteristics of historical click-through data. They stated that intuitively more accurate similarity values between queries can be obtained by taking into account the timestamps of the data, because the query term similarity is dependant from factors such as seasons, holidays, special events, etc. They propose a novel time-dependent query term semantic similarity model which can exploit the click-through data more effectively than traditional approaches. They concluded, that for most query pairs similarities are time-dependent and their model can calculate more accurate similarity values than other approaches.
Vlachos *et al.* (2004) described several methods to identify similarities, periodicities and bursts in search query logs. They discovered these similarities by comparing the demand patterns of query terms. They described experiments conducted with a tool especially developed for the analysis of log data and described with several examples the applicability of their contributions. Automatically identifying bursts in usage data allows detecting and reacting adequately to special events occurring over time. In the context of a learning system we could use their approach to identify changes in demand and prominently position "featured" learning materials.
A similar approach is presented by Chien and Immorlica (2005) for finding the highest temporally correlated queries for an input query. They developed a method using far less space and time than the naïve approach, and described how this made real-time implementation possible. They demonstrated that they are able to efficiently compute their measure given the scale of a search engine query stream such as the MSN search engine.

Recommendations

Recommender systems (also known as recommendation engines) work from a set of data (usually a set of user-item relationships, sometimes enriched with additional data) and attempt to recommend items (films, music, web pages, etc.) that are likely to be of interest to the current user. Since the first appearance of the subject of recommender systems there has been much work done in this area. Recommender systems are usually based on collaborative

filtering (Pierrakos, 2003, Su, 2009), content-based filtering (Kosala, 2000), or a combination of those methods (Balabanović, 1997, Adomavivius, 2005, Burke, 2002). For this work we are especially interested in multidimensional recommender systems and systems employing additional data, especially time data.
Adomavicius *et al.* (2005) described a multidimensional approach to recommender systems that provided recommendations based on additional contextual information. They described several use-cases where employing additional data is useful for improving the recommendations, for example the personalisation of a web site, where the user wants to read the news in the *morning* and the stock market report in the *evening*. On the *weekend* this user might be more interested in films and shopping. It seems that it is important to incorporate additional data like time and other contextual information in the recommendation process. They demonstrated that *context matters* and their multidimensional approach can produce better rating estimations than other approaches in some situations. They also did not forget to mention that there are cases where context does not matter at all. Therefore they implemented a combined approach and demonstrated its strength with an empirical evaluation of their implementation.
Tang, Winoto and Chang (2003b) argued that collaborative filtering and content-based methods are the two most-used approaches in recommender systems. They pointed out that most hybrid approaches focus on the contents of the items, and not on the temporal feature of them. In their study they examined recommendations for movies and found out that the movies' production year can significantly affect users' preferences. They called this the *temporal effect* of the items on the performance of recommender systems. Their experiments show that the performance of the recommender can be improved by taking into account the temporal effect, if the data were accurately chosen. They concluded that the usefulness of the temporal data might be due to the genre of a movie (there might be genres of movies where the production date does not matter). This approach is similar to ours. But instead of not considering older data, the time-distance to the current date is a crucial element of our similarity measure.
To the best of our knowledge there has not been much research in the area of usage mining of lecture recordings. Some groups, such as the research group from the University of Osnabrück developing *virtPresenter,* are analysing the importance of parts of lecture recordings. They are analysing the view count and view duration of snippets of lecture recordings and integrating the results of this analysis in social networks like Facebook (Fox, 2009). But so far, there has not been a lot of research on recommender systems which analyse downloads / views of documents (especially lecture materials) based on the time when the documents were accessed. In the next section we describe our findings in this area, show why time is an important datum in usage logs and how we can exploit this to create a recommender system for lecture materials. The emphasis on using time data from usage logs as a similarity measure and not only as metadata or classification metric, clearly distinguishes our work from prior research in this area.

Analysis and Evaluation of Students' Behaviours

To find out what lecture materials and file formats are most used and if there is a correlation between the demand for a certain document and events such as exams or deadlines for tutorials we analysed web usage data from our Electures-Portal.The Electures-Portal stores meta data about each course that allowed us to assign every log file entry to each corresponding course (their title, semester and lecturers). We analysed this data over a period of one year, which contained around 2.6 million entries. Before processing we cleaned our log from entries that are likely to be requests from robots. Then we assumed that most of the log file entries are requests made by our students although they also might contain requests from foreigners since some contents are freely available. We analysed two of the courses which contained the most requested files (one basic course "Informatik II", a course on Computer Science and one more advanced course "Algorithmentheorie", an advanced course on algorithm theory).
We analysed which file formats were most often downloaded by students and if materials from older versions of the course were still downloaded in current courses. We also investigated if there are specific times where the download rate is especially high and several other questions. For answering the question about the file formats we categorized the files in the three categories *video, flash* and *slides*. We found out that students preferred the video files followed by slides, and flash was the least used category of files. We also found out that materials from courses in previous semesters were still downloaded during the semester of current courses and that an archive was needed since we detected an over-average demand for older materials at the beginning of each semester.
Especially interesting for this work was to find out if there is a correlation between the students downloads and dates like tutorial deadlines or exams. The students downloads seem to follow a similar demand pattern which is periodical and correlated to the deadlines of tutorials and exam dates. This alone is not especially surprising, but similar to materials of the current course materials from similar courses from previous semesters are also requested. More detailed results of these studies can be found in the publication of Hermann *et al.* (2007).

Implementing and Evaluating our Recommender System

Based on these results we wanted to improve the direct access of our lecture materials in the Electures-Portal and give students item-based recommendations on other lecture materials while they are downloading a specific file. Since we could not make sure which file format really best matched the users preferences we decided to group the files and recommend specific lectures (with all their materials) instead of recommending single files. Our log data consists of entries of downloads in the period from 22. November 2005 to 20. January 2010. In a first step we pre-processed the data to assign every download in our log files the corresponding `modulId` (an ID identifying the course), the `electureId` (the ID identifying the corresponding lecture) and cleaned our logs from unwanted entries. We made a detailed analysis on our data to identify bots and marked these entries in our data set. Every entry in our log file represents a preference with a `userId`, an `electureId` and a timestamp. After cleaning we loaded the 4,791,278 entries in an efficient memory-based representation of our recommender data model. To minimize memory consumption we are representing our timestamp data as a Unix-timestamp (seconds since 1.1.1970). Another advantage of this representation is that it eases date computations, for example the distance between two dates can be obtained by simply subtracting two timestamps.

Our new Approach for a Time-Based Recommender

Other recommender systems usually use either plain boolean preferences (user-item tuples) or an additional rating (user-item-rating triples). The systems then computes for every two items for a certain user, the similarity of these items with the given ratings or boolean preferences. We also processed our data set (using boolean preferences) with a default implementation of a recommender system for benchmarking purposes. Our implementation is based on the open source implementation of the recommender system *Taste* which is part of the *Apache Mahout* project. For evaluation purposes we applied their Loglikelihood recommender for boolean preferences on our data. We chose the Loglikelihood recommender since this gave the best results (based on precision and recall) on our data set. According to Owen and Anil the metric of the Loglikelihood recommender is based on the number of items that two users have in common, but "its value is more an expression of how unlikely it is for two users to have so much overlap, given the total number of items and the number of items each user has preferences for". A detailed explanation of the Loglikelihood metric can be found in the original publication from Dunning (1993).
To recommend items based on users' preferences for other items we need to convert the logs into a list of `TimePreferences(long userId, long itemId, Date timestamp)`. Since we need to be able to make recommendations in real time we seek a representation that reduces the memory usage of our application. Following the example of the recommender implementation from the Mahout project, our `TimeDataModel`, therefore, consists of a special representation of this list of `TimePreferences` called `TimePreferenceArray`. The main difference to a naïve implementation is that our representation stores the timestamps as an array of `long` values instead of `Date` objects, and that we avoid the duplication of the `userId` for every `TimePreference`. This representation allows us to ask our model for all `TimePreferences` that exist for a specific `itemId` very efficiently.

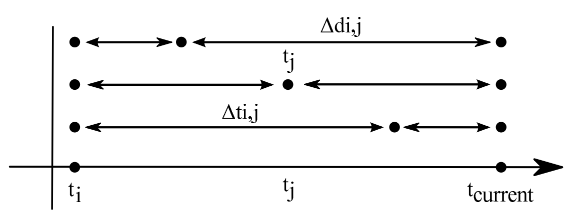

Image 1: Different possibilities of the similarity between two items i *and* j related to the current date $t_{current}$.

Since we want to exploit the time value of our log entries and also take into account the current time (the time when the recommendations are requested) we define a similarity measure. Let $S_{i,j}$ be the similarity measure for two items i and j, and let $\Delta t_{i,j}$ and $\Delta d_{i,j}$ be the date difference and minimum distance of the items i and j with their corresponding download dates t_i and t_j to the current date $t_{current}$, respectively.

Then calculate $S_{i,j} = \frac{1}{(\Delta t_{i,j} + \Delta d_{i,j})}$ where $\Delta t_{i,j} = \left| (t_i - t_j) \right|$ and $\Delta d_{i,j} = \left| \min(t_i, t_j) - t_{current} \right|$.

The intuitive approach behind this similarity measure is that two items are more similar to each other when their download dates t_i and t_j were close (for the same user). Additionally we consider the distance between downloads and the current date $t_{current}$, because we deem more recent downloads to be more relevant. Image 1 depicts the intuition behind this approach. There are three possible cases for the relation of two downloads and the current date (if one considers the two downloads as interchangeable).

We penalize the date difference $\Delta t_{i,j}$ between two items additionally and therefore choose the minimum download date value to calculate the distance between the two downloads and the current date (one could also take the maximum and double the value of $\Delta d_{i,j}$). Let $p(u,i)$ be a boolean function that returns true whenever user u has a preference for item i. Our time-based recommender then works as described in Algorithm 1. A comparison of the results of our time-based recommender system with the Loglikelihood recommender is shown in the next section.

Input: A `TimeDataModel` (a list of `TimePreferences` for every user)
Ouput: A ranked list of recommendations for an item
Calculate the list of users UwP having a preference for item i and item j as follows:

$$UwP_{i,j} = \{u \mid p(u,i) \wedge p(u,j)\}$$

$\forall u \in UwP_{i,j}$ calculate $S_{i,j}$ and

return a list of the top n items ranked by their averaged similarities $avg(S_{i,j})$.

Algorithm 1: Pseudocode of our time based recommendation algorithm.

Evaluating our Recommender: Measuring Performances

Measuring the performance of a recommender system is quite a difficult task, since recommenders try to predict which items users will most likely prefer in the future. To achieve a reliable comparison between our algorithm and the Loglikelihood recommender we created a set of over 2000 test cases. These test cases consist of an item to make recommendations for, as well as two data sets. One of the data sets consisting of all the data to be used for making recommendations (with download dates before the current item was downloaded by a specific user), and another data set with items that the same user downloaded after downloading the item we are currently creating recommendations for. Let x be the item and t_x the time to make recommendations for. Then $\sigma(u)$ is the set of relevant items that the user u has downloaded after t_x and $\phi(x)$ the set of recommended items that were recommended for the item x (retrieved items for item x). We then evaluated the performance of the recommender by calculating the precision P and recall R for the top 10 recommendations.

$$P = \frac{|\sigma(u) \cap \phi(x)|}{|\phi(x)|} \text{ and } R = \frac{|\sigma(u) \cap \phi(x)|}{|\sigma(u)|}$$

The average precision and recall values for our recommender are slightly better (P=0.87, R=0.17) then those for the Loglikelihood recommender (P=0.84, R=0.16) on our data set. Our algorithm performs best, when there is a lot of data available for the specific users. In practice this means that students should login before downloading lecture materials to get better recommendations. The improved precision and the incorporation of the timestamps of our algorithm is useful for Learning Management Systems and Lecture Archives when ratings for the items are not available. It is especially useful, when the demand of groups of users is similar over time. Using our complete data set we are able to make recommendations in 5ms on average and a maximum of 18ms.

Conclusion and Outlook

We have shown that when making recommendations for lecture materials, analysing the time of a preference is important, since students who are attending the same courses usually are interested in the same lecture materials at the same time. Our studies have shown that because of the preferences for different file formats it is better to recommend whole lectures instead of files. Based on these findings a new approach of a time-based recommendation algorithm has been presented and our experiments show that this algorithm outperforms existing recommendation algorithms for boolean preferences. In the future we will track downloads done made after receiving a recommendation and further compare the performance of the different algorithms. We also want to implement a content-based recommender that will allow the students to discover learning materials that are related to the topic of the materials they are currently downloading. We hope that this will help the students to locate important lecture materials and therefore shorten the time finding the relevant contents in our archive.

References

Adomavicius, G. Sankaranarayanan, R., Sen, S. and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. In *ACM Transactions on Information Systems* Volume 23, Issue 1, pages 103-145.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. In *Communications of the ACM*. Volume 40, Issue 3, pages 66-72.

Beitzel, S., Jensen, E., Chowdhury, A., Frieder, O. and Grossman, D. (2007). Temporal analysis of a very large topically categorized Web query log. *In Journal of the American Society for Information Science and Technology.* Volume 58, Issue 2, pages 166-178.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. In *Proceedings of User Modeling and User-Adapted Interaction*. Volume 12, Issue 4, pages 331-370.

Chien, S. and Immorlica, N. (2005). Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, New York, pages 2-11.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. In *Computational Linguistics* Volume 19, Issue 1, pages 61-74.

Fox, P. Emden, J., Neubauer, N. and Vornberger, O. (2009). Vorlesungsaufzeichnungen im Kontext sozialer Netzwerke am Beispiel von Facebook. In *Proceedings of the Pre-Conference Workshop of the DeLFI2009: eLectures 2009 – Anwendungen, Erfahrungen und Forschungsperspektiven*.

Hermann, C., Welte, M., Latocha, J., Wolk, C. and Hürst, W. (2007). Eine logfilebasierte Evaluation des Einsatzes von Vorlesungsaufzeichnungen In *Tagungsband der 5. e-Learning Fachtagung Informatik (DeLFI 2007)* Siegen, Sep. 2007.

Hürst, W. (2005). Multimedia Information Search in Presentation and Lecture Recordings, Ph.D. thesis, University of Freiburg.

Hürst, W. and Deutschmann, N. (2006). Searching in recorded lectures. In *Proceedings of the World Conference on E-Learning in Corporate Government, Healthcare & Higher Education (E-Learn 2006)*, AACE, Honolulu, HI, USA.

Kosala, R. and Blockeel, H. (2000). Web mining research: a survey. In *SIGKDD Explorations*, Volume 2 Number 1, pages 1–15.

Lauer, T., Müller, R. and Trahasch, S. (2004). Learning with Lecture Recordings: Key Issues for End-Users, ICALT, In *Proceedings of the Fourth IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, Joensuu, Finland, pages 741-743.

Owen, S. and Anil, R. (to appear). *Mahout in Action*. Manning Publications Co.

Pierrakos, D., Paliouras, G., Papatheodorou, C. and Spyropoulos, C. (2003). Web Usage Mining as a Tool for Personalization: A Survey. In *Proceedings of User Modeling and User-Adapted Interaction.* Volume 13, Number 4, Springer Netherlands, pages 311-372.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. (2000). Web usage mining: discovery and applications of usage patterns from Web data. In *SIGKDD Explorations* Volume 1, Issue 2, pages 12-23.

Su, X. and Khoshgoftaar, T. (2009). A survey of collaborative filtering techniques. In *Advances in Artificial Intelligence* Volume 2009, Article Number 4.

Tang, T., Winoto, P. and Chan, K. (2003b). On the Temporal Analysis for Improved Hybrid Recommendations. In *Proceedings of the 2003 IEEE/WIC international Conference on Web Intelligence*. IEEE Computer Society, Washington, DC, 214.

Vlachos, M., Meek, C., Vagena, Z. and Gunopulos, D. (2004). Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. SIGMOD '04. ACM, New York, pages 131-142.

Welte, M. Eschbach, T. and Becker, B. (2005). Automated Text Extraction And Indexing Of Video Presentation Recordings For Keyword Search Via A Web Interface. In *Proceedings of E-Learn 2005,* AACE Press.

Zhao, Q., Hoi, S., Liu, T., Bhowmick, S., Lyu, M. and Ma, W. (2006). Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th International Conference on World Wide Web*.ACM Press, New York, pages 543-552.