A Survey on Using Baum-Welch Algorithm For Sharing Fine Grained Knowledge in Mutual Environments

A.M.Pushpalatha and A.Nirmal Kumar

Received: 23 September 2015; Accepted: 25 October 2015

ABSTRACT

Knowledge Sharing is an activity through which knowledge is exchanged among people, friends, families, communities or organizations. Mutual Environments which enable company-wide global teams to identify the source of the antidote to a lack of preparedness. This paper investigates Fine grained knowledge sharing in collaborative environments. Two step framework is used. 1) Web surfing data are clustered into tasks by LEGDP (Laplacian Eigenmap Gaussian Drichlet Process) Model. 2) From Each Task Micro Aspects are extracted by d-iHMM (Discriminative-infinite Hidden Markov Model) model. And to find proper members for knowledge sharing, the classic expert search method is applied to the mined results. Existing Hidden Markov Models takes larger memory and execution time. To overcome this we propose Baum-Welch algorithm. This provides more accuracy than HMM and also takes less execution time to find the best advisor for our related query.

Key words - User Request, Traditional Expert Search, Preprocessing, LEGDP, d-iHMM, Advisor search

INTRODUCTION

Data Mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data Mining is also called as data or knowledge discovery. In this project dataset is created based on the user request. This clusters all web surfing data into tasks.

Web Mining is the application of data mining techniques to discover patterns from the World Wide Web (WWW). Web Mining has three types. Web Usage mining is used to discover interesting usage patterns from web data. Usage data captures the identity or origin of web users along with their browsing behavior [1] at the web site. Web structure Mining is the process of using graph theory to analyze the node and connection structure of web site. Web Content Mining is the mining, extraction and integration of useful data, information and knowledge from Web page content. Data Mining applications range from commercial to social domains. Present-Day data mining is a progressive multidisciplinary endeavor. This inter and multidisciplinary approach is well reflected within the field of information systems.

Data Mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data Mining is also called as data or knowledge discovery. In this project dataset is created based on the user request. This clusters all web surfing data into tasks.

Web Mining is the application of data mining techniques to discover patterns from the World Wide Web (WWW). Web Mining has three types. Web Usage mining is used to discover interesting usage patterns from web data. Usage data captures the identity or origin of web users along with their browsing behavior [1] at the web site. Web structure Mining is the process of using graph theory to analyze the node and connection structure of web site. Web Content Mining is the mining, extraction and integration of useful data, information and knowledge from Web page content. Data Mining applications range from commercial to social domains. Present-Day data mining is a progressive multidisciplinary endeavor. This inter and multidisciplinary approach is well reflected within the field of information systems.

Most people in collaborative environments would be happy to share experiences with and give suggestions to others on specific problems. However, finding a right person is challenging due to the variety of information needs. This paper investigates how to enable such knowledge sharing mechanism by analyzing user data. Our goal is to find proper “advisors” who are most likely possessing the desired piece of fine-grained knowledge based on their web surfing activities.

In order to analyze the knowledge acquired by web users, we propose to log and analyze users’ web surfing data (not only search, but also browsing...
activities, which reveal a user’s knowledge gaining process. Users’ interactions with the web can be segmented into different “tasks”. Textual contents of a task are usually cohesive. We define a session as an aggregation of consecutively browsed web contents of a user that belong to the same task. Sessions are atomic units in our analysis. The content of sessions in a task could evolve gradually: people usually learn basic concepts first and then move towards advanced topics. A task can be further decomposed into fine-grained aspects (called micro-aspects). A micro-aspect could be roughly defined as a significantly more cohesive subset of sessions in a task. When pursuing a task, a user could spend many sessions on a micro-aspect. Mining these micro-aspects (micro-knowledge) is critical: it can provide a detailed description of the knowledge gained by a person, which is the basis for advisor search.

Related Works:

My work is closely related to several groups of research works

A. Traditional Expert Search:

Expert search aims at retrieving people who have expertise on the given query topic. Early approaches involve building a knowledge base which contains the descriptions of people’s skills within an organization. Expert search became a hot research area since the start of the TREC enterprise track in 2005. Balog et al. proposed a language model framework [1] for expert search. Their Model 2 is a document-centric approach which first computes the relevance of documents to a query and then accumulates for each candidate the relevance scores of the documents that are associated with the candidate. This process was formulated in a generative probabilistic model. Model 2 performed better [1] and it became one of the most prominent methods for expert search. Other methods have been proposed for enterprise expert search but the nature of these methods is still accumulating relevance scores of associated documents to candidates. Expert retrieval in other scenarios has also been studied, e.g. online question answering communities, academic society [6].

The proposed advisor search problem is different from traditional expert search. (1) Advisor search is dedicated to retrieving people who are most likely possessing the desired piece of fine-grained knowledge, while traditional expert search does not explicitly take this goal. (2) The critical difference lies in the data, i.e. sessions are significantly different from documents in enterprise repositories. A person typically generates multiple sessions for a micro aspect of a task, e.g. a person could spend many sessions learning about Java multithreading skills. In other words, the uniqueness of sessions is that they contain semantic structures which reflect people’s knowledge acquisition process. If we treat sessions as documents in an enterprise repository and apply the traditional expert search methods (e.g. [1]), we could get incorrect ranking: due to the accumulation nature of traditional methods, a candidate who generated a lot of marginally relevant sessions (same task but other micro aspects) will be ranked higher than the one who generated less but highly relevant sessions. Therefore, it is important to recognize the semantic structures and summarize the session data into micro-aspects so that we can find the desired advisor accurately. In this paper we develop nonparametric generative models to mine micro aspects and show the superiority of our search scheme over the simple idea of applying traditional expert search methods on session data directly.

B. Analysis of Search tasks:

Search tasks are interleaved and used classifiers to segment the sequence of user queries into tasks [1] and combined task stage and task type with dwell time to predict the usefulness of a result document, using a three-stage and two-type controlled experiment used graph regularization to identify search tasks in query logs and designed classifiers to identify same-task queries for a given query and to predict whether a user will resume a task formulated the cross-session search task mining problem as a semi-supervised clustering problem where the dependency structure among queries in a search task was explicitly modeled and a set of automatic annotation rules were proposed as weak supervision.

This line of research tries to recover tasks from people’s search behaviors and bears some similarity to our work. Nevertheless, our work differs from theirs from the following aspects. First, we consider general web surfing contents (including search), rather than search engine query logs. Query logs do not record the subsequent surfing activity after the user clicked a relevant search result. Moreover, it is found that 50 percent of a user’s online pageviews are content browsing [2] Web surfing data provides more comprehensive information about the knowledge gaining activities of users. Although various methods were proposed for extracting search tasks in query logs, these methods cannot be applied in our setting since they exploit query log specific properties. Second, none of the above works tried to mine fine-grained aspects for each task. This work sets the stage for evaluating search engines, not on a per-query basis, but on the basis of user tasks.

B. Topic Modelling:

Topic modeling is a popular tool for analyzing topics in a document collection. The most prevalent topic modeling method is Latent Dirichlet Allocation (LDA) [3]. Based on LDA, various topic modeling methods have been proposed, e.g. the dynamic topic model for sequential data and the hierarchical topic model for building topic hierarchies. The Hierarchical DP (HDP) model can also be
instantiated as a nonparametric version of LDA. However, our problem is not a topic modeling problem. Our goal is to recover the semantic structures of people’s online learning activities from their web surfing data, i.e. identifying groups of sessions representing tasks and micro-aspects. While topic modeling decomposes a document into topics. After applying topic modeling methods on session data, it is still difficult to find the right advisor by using the mined topics. This is because a person with many sessions containing partially relevant topics would still be ranked unexpectedly high, due to the accumulation of relevance among sessions. Grouping sessions into micro-aspects is important for advisor search.

D. Session Clustering:

Laplacian Eigenmap Gaussian Drichlet Process(LEGDP) is used for Session Clustering. When using probabilistic models for clustering, the Gaussian mixture model is a common choice and can be viewed as a probabilistic version of k-means [4]. The input of this step is W, where each \(w_i\) is a \(D_0\ times 1\) word frequency vector with \(D_0\) as the vocabulary size. The intuition is that contents generated for the same task are textually similar while those for different tasks are dissimilar. Hence, clustering is a natural choice for recovering tasks from sessions. In our case, it is difficult to preset the number of tasks given a collection of sessions. Therefore, we need to automatically determine the number of clusters \(k\), which is also one of the most difficult problems in clustering research. Most methods for automatically determining \(k\) run the clustering algorithm with different values of \(k\) and choose the best one according to a predefined criterion which could be costly.

DPs provide nonparametric priors for \(k\) and the most likely \(k\) is learned automatically. A DP, written as \(G \rightarrow \text{DP}(\alpha, G_0)\) can be interpreted as drawing components (clusters here) from an infinite component pool, with a called the scaling parameter and \(G_0\) being the prior for a random component. An intuitive interpretation of DP is the stick-breaking construction: \(\pi(v) = \prod_{k=1}^\infty (1-\nu_k) G_0 \sum_{\nu_k} \pi \delta_{\nu_k}\) Where \(v = \{v_1, v_2, \ldots\}\) with each \(v_i\) drawn from the Beta distribution.Beta(1, \(\alpha\), \(\nu_i\)) is a component drawn from \(G_0\) and \(\delta_{\nu_i}\) is an atom at \(\nu_i\). \(\pi\) is the mixture weight of \(\nu_i\) given by breaking the current length of the “stick”

However, the data dimensionality \(D_0\) is too high to apply Gaussian distributions in our case (often above 10K). Therefore, we first apply the well-known Laplacian Eigenmap (LE) technique [5] to reduce the dimensionality from \(D_0\) to \(D\) where \(D_0 >> D\). We choose LE since it could also capture the nonlinear manifold structure of a task, e.g. the topics of a task could evolve and drift which could be described by the “half-moon” structure.

E. Mining Fine Grained Knowledge:

The discriminative-infinite Hidden Markov Model(d-iHMM) [6] is used for extracting micro-aspects in each task. The major challenge of mining micro-aspects is that the micro-aspects in a task are already similar with one another. If we model each component (i.e. micro-aspect) independently (as most traditional models do), it is likely that we mess up sessions from different micro-aspects, i.e. leading to bad discrimination. Therefore, we should model different micro-aspects in a task jointly, separating the common content characteristics of the task from the distinctive characteristics of each micro-aspect. To this end, we extend the infinite Hidden Markov Model (iHMM) and propose a novel discriminative infinite Hidden Markov Model to mine micro-aspects and possible evolution patterns in a task.

An HMM defines a probability distribution over sequences of observations (symbols) \(Y = \{y_1, y_2, \ldots, y_T\}\) by invoking another sequence of unobserved, or hidden, discrete state variables \(s = \{s_1, s_2, \ldots, s_T\}\). The basic idea in an HMM is that the sequence of hidden states has Markov dynamics i.e. given \(s_t, s_{t-1}\) is independent of all other variables given \(s_t\). The model is defined in terms of two sets of parameters, the transition matrix whose \(i^{th}\) element is \(p(s_{t+1} = i | s_t = j)\) and the emission matrix whose \(i^{th}\) element is \(p(y_t = q | s_t = i)\). The emission process \(s_t \rightarrow y_t\) is identical to the transition process \(s_t \rightarrow s_{t+1}\) in every respect except that there is no concept analogous to a self-transition. The usual procedure for estimating the parameters of an HMM is the Baum-Welch algorithm, a special case of EM, which estimates expected values of two matrices \(n\) and \(m\) corresponding to counts of transitions and emissions respectively, where the expectation is taken over the posterior probability of hidden state sequences. iHMM tries to model the background content in each state independently, which leads to low discriminative power. On the contrary, d-iHMM has higher discriminative power by modeling background words in each state by a common background unigram model. d-iHMM is more costly. Fortunately, the computation for different tasks can be parallelized. The advisor search phase only requires a few milliseconds since its main cost is one matrix vector multiplication.

F. Advisor Search:

After we obtain the mined micro-aspects of each task, advisor search can then be implemented on the collection of learned micro-aspects. Advisor search is dedicated to retrieving people who are most likely possessing the desired piece of fine-grained knowledge. We employ the traditional language model based expert search method [1] which is used as a retrieval method. Let \(d\) be a document (i.e. micro-aspect). Given a query \(q\), the method uses \(p(e|q)\) to rank advisor candidates. By assuming
uniform prior distributions \( p(e) \) and \( p(d) \) and applying Bayes’ rule, it is equivalent to rank candidates by \( p(q|e) = \sum p(q|d) p(d|e) \) [1]. \( p(q/d) \) is the probability of generating \( q \) by \( d’s \) unigram model, with proper smoothing [1]. Intrinsically, the method can be viewed as a weighted accumulation of \( p(q/d)’s \) from the associated documents of \( e \). Recall that the weight between \( e \) and \( d \) is the number of sessions of \( e \) which fall in \( d \). \( p(d/e) \) and \( p(e/d) \) encode the normalized association weights between candidates and documents from a candidate’s perspective (candidate scheme) and a document’s perspective (document scheme), respectively. The candidate scheme is not intuitive in our context. Consider two candidates \( e_1 \) and \( e_2 \). \( e_1 \) viewed totally 100 sessions in which 10 sessions fall in \( d \), while for \( e_2 \) the two numbers are 10 and 2. Hence, \( p(d/e_1) = 0.1 < p(d/e_2) = 0.2 \). However, \( e_1 \) viewed more sessions in \( d \) than \( e_2 \) and should have a stronger association. Therefore, the document scheme is used for ranking. Compared to applying traditional expert search methods directly on session data, searching over micro-aspects has the advantage that the associations between candidates and “documents” are correctly normalized.

**Conclusion:**

Fine-grained knowledge reflected by people’s interactions with the outside world is the key to solving this problem. We proposed a two-step framework to mine fine-grained knowledge and integrated it with the classic expert search method for finding right advisors. For mining micro aspects we used discriminative-infinite Hidden Markov Model which is more costly and also it provides less accuracy. It consumes large memory and longer execution time. We proposed Baum-Welch algorithm, which is the extension of HMM, for mining micro-aspects. This algorithm gives more accuracy than HMM. This algorithm uses two matrices for finding probability which implies the best Advisor related to our query. Emission is at every hidden and observed State. Transition is between two Hidden states. In this work, we demonstrate the feasibility of mining task micro-aspects for solving this knowledge sharing problem.

**References**