Modeling Semantics and Structure of Discussion Threads[*]

Chen Lin[†], Jiang-Ming Yang[‡], Rui Cai[‡], Xin-Jing Wang[‡], Wei Wang[†], Lei Zhang[‡]

[†]School of Computer Science, Fudan University. {chen_lin, weiwang1}@fudan.edu.cn

[‡]Microsoft Research, Asia. {jmyang, ruicai, xjwang, leizhang}@microsoft.com

ABSTRACT

The abundant knowledge in web communities has motivated the research interests in discussion threads. The dynamic nature of discussion threads poses interesting and challenging problems for computer scientists. Although techniques such as semantic models or structural models have been shown to be useful in a number of areas, they are inefficient in understanding discussion threads due to the temporal dependence among posts in a discussion thread. Such dependence causes that semantics and structure coupled with each other in discussion threads. In this paper, we propose a sparse coding-based model named SMSS to **S**imultaneously **M**odel **S**emantic and **S**tructure of discussion threads.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

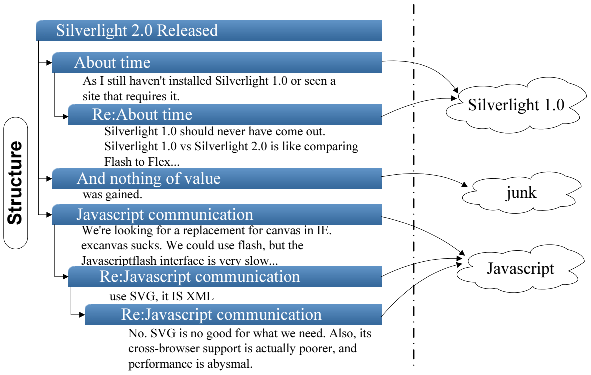
Algorithms, Experimentation

Keywords

Threaded discussion, sparse coding, reply reconstruction

1. INTRODUCTION

Discussion threads have long been a popular option for web users to exchange opinions and share knowledge, e.g. thousands of web forum sites, mailing lists, chat rooms, and so on. A discussion thread usually originated from a root post by the thread starter. Fig. 1 gives an intuitive description of a thread[1]. It contains 7 posts. The first post is a piece of news about the release of "SilverLight 2.0". Some users comment on this post, *i.e.*, the 2^{nd} and 3^{rd} posts are about the "update time"; some users have further questions and initiate sub-discussions, *i.e.*, the 5^{th}, 6^{th}, and 7^{th} posts are about "Javascript communication"; others troll or complain, *i.e.*, the 4^{th} post. As more users joining in and making comments, the thread grows, forming a nested dialogue ***structure*** as shown in the left part of Fig. 1. Furthermore, discussion threads show rich complexity in the ***semantics***. Since users always response to others, previous posts affect later posts and cause the topic to drift in a thread. This is shown in the right part of Fig. 1. The goal of this paper is to model both the ***structure*** and ***semantics*** of a discussion thread in a simultaneous way.



Silverlight 2.0 Released

About time

As I still haven't installed Silverlight 1.0 or seen a site that requires it.

Re:About time

Silverlight 1.0 should never have come out. Silverlight 1.0 vs Silverlight 2.0 is like comparing Flash to Flex...

And nothing of value

was gained.

Javascript communication

We're looking for a replacement for canvas in IE. excanvas sucks. We could use flash, but the Javascriptflash interface is very slow...

Re:Javascript communication

use SVG, it IS XML

Re:Javascript communication

No. SVG is no good for what we need. Also, its cross-browser support is actually poorer, and performance is abysmal.

Figure 1: An example of the structure and semantics of a discussion thread from Slashdot.

2. THE SMSS MODEL

A discussion thread has the following four characteristics.

A discussion thread has several topics. Suppose there are T topics and V words, the j^{th} topic is described as a distribution over the word space $\mathbb{R}^V$, as $\mathbb{R}$ is real numbers and $\vec{x}^{(j)} \in \mathbb{R}^V, 1 \leq j \leq T$. Then, each post $\vec{d}^{(i)}$ is expressed as a mixture of topics, as $\vec{d}^{(i)} \simeq \sum_{j=1}^{T} \theta_j^{(i)} \cdot \vec{x}^{(j)}$, where $\theta_j^{(i)}$ is the coefficient of $d^{(i)}$ on topic $\vec{x}^{(j)}$. To estimate the topic space $X = \{\vec{x}^{(1)}, \ldots, \vec{x}^{(T)}\}$, in SMSS we minimize the loss function $\|D - X\Theta\|_F^2$. Here the thread contains L posts as $D = \{\vec{d}^{(1)}, \ldots, \vec{d}^{(L)}\}$; and the coefficient matrix $\Theta = \{\vec{\theta}^{(1)} \ldots \vec{\theta}^{(L)}\}$.

An individual post is related to a few topics. Although one thread may contain several semantic topics, each individual post usually concentrates on a limited number of topics. Therefore, we assume $\vec{\theta}^{(i)}$ of each post is sparse and introduce a regularizer $\|\vec{\theta}^{(i)}\|_1$ in SMSS.

A post is related to its previous posts. Users usually read current posts in a thread before they reply. Thus the semantics of a reply post is related to its previous posts. In SMSS we formally describe such reply structure as a regularizer $\|\vec{\theta}^{(i)} - \sum_{k=1}^{i-1} b_k^{(i)} \cdot \vec{\theta}^{(k)}\|_F^2$, where $b_k^{(i)}$ is the structural coefficient between the i^{th} and k^{th} posts. In other words, $\vec{\theta}^{(i)}$ can be expressed as a linear combination of $\vec{\theta}^{(k)}$.

The reply relations are sparse. In most situations, users only intend to comment on one or two previous posts. Again, in SMSS we introduce a regularizer to favor such sparse structural coefficients $\|\vec{b}^{(i)}\|_1$.

Based on the above observations, the SMSS model is to estimate the value of topic matrix X, the coefficient matrix Θ, and the structural coefficients b for each post, by mini-

[*]This work was done when the first author visiting Microsoft Research, Asia.

[1]http://developers.slashdot.org/comments.pl?sid=1000769

Copyright is held by the author/owner(s).

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

mizing the following loss function f:

$$f = \|D - X\Theta\|_F^2 + \lambda_1 \sum_{i=1}^{L} \|\vec{\theta}^{(i)}\|_1$$

$$+\lambda_2 \sum_{i=1}^{L} \|\vec{\theta}^{(i)} - \sum_{k=1}^{i-1} b_k^{(i)} \cdot \vec{\theta}^{(k)}\|_F^2 + \lambda_3 \sum_{i=1}^{L} \|\vec{b}^{(i)}\|_1 \qquad (1)$$

Here, the optimization objective balances the four terms by parameter λ_1, λ_2, and λ_3. In this way, both the semantics and the structure information are estimated simultaneously. Furthermore, for a collection of M threads which shared the same topics matrix X, we can optimize them together by minimizing:

$$\text{minimize}_{X, \{\Theta^{(t)}\}, \{\vec{b}^i\}} \sum_{n=1}^{M} f^{(n)}(\cdot) \qquad (2)$$

The optimization problem is not jointly convex but can be solved by iteratively minimizing the convex sub-problems.

3. APPLICATIONS

To demonstrate the efficiency of the proposed SMSS model, we reconstruct the reply relationships among posts using the semantics and structures estimated by SMSS.

3.1 Reply Reconstruction

Intuitively, posts with reply relations should have similar terms. However, the term similarity is unreliable as posts in discussion threads are usually very short. Our idea is to integrate the revealed semantic topics and structure as additional information in the similarity measure. Formally the similarity of a given post j and a previous post i is the combination of all the features, as:

$$sim(i,j) = sim(\vec{d}^{(i)}, \vec{d}^{(j)}) + w_1 \cdot sim(\vec{b}^{(i)}, \vec{b}^{(j)})$$

$$+w_2 \cdot sim(\vec{\theta}^{(i)}, \vec{\theta}^{(j)}) \qquad (3)$$

Based on the similarity, we propose an approach to analyze a thread with L posts. That is, for a new post we compute the similarity between itself and all previous posts, rank the similarity, choose the post with highest score as a candidate parent. In case that the candidate parent is not similar enough to the new post, we assume this post initializes a new discussion branch of the thread.

3.2 Experiment

In experiment, we adopt two forums, Apple Discussion[2] and Slashdot[3], as our data sources. These two forums are carefully selected because they have provided clear reply relations for evaluation. Since threads in Apple Discussion are much shorter, we sample 2000 threads including 20000 posts. For Slashdot, we sample 100 threads which also contains about 20000 posts. We manually write a wrapper to parse these pages and extract the exact reply relations as the ground truth. The evaluation metric is precision.

For comparison, we also adopt some naive methods such as Nearest-Previous (NP), Reply-Root (RR), and Only Document Similarity (DS). NP assigns each post to the nearest previous post as the reply target; RR assigns each post to the root post as the reply target; and DS assigns each post to the post has most similar terms.

Moreover, we also compared our SMSS model with some state-of-the-art models which can provide semantic topic analysis, such as latent dirichlet allocation (LDA) [1] and the special words with background model (SWB) [2]. Similar to Eq.3, we compute the post similarity by $sim(i,j) = sim(\vec{d}^{(i)}, \vec{d}^{(j)}) + w_1 \cdot sim(\vec{\theta}^{(i)}_{LDA}, \vec{\theta}^{(j)}_{LDA})$ from LDA. While SWB is an extension of LDA and allows words in documents to be modeled as either originating from general topics, or from post-specific "special" word distributions, or from a thread-wide background distribution. We leverage both its topic distribution $\vec{\theta}^{(i)}_{SWB}$ and special-words distribution $\vec{\psi}^{(i)}_{SWB}$ for similarity computing, as $sim(i,j) = sim(\vec{d}^{(i)}, \vec{d}^{(j)}) + w_1 \cdot sim(\vec{\theta}^{(i)}_{SWB}, \vec{\theta}^{(j)}_{SWB}) + w_2 \cdot sim(\vec{\psi}^{(i)}_{SWB}, \vec{\psi}^{(j)}_{SWB})$.

Table 1: Performance of reply reconstruction in all posts v.s. high-quality posts

Method	Slashdot		Apple	
	All Posts	Good Posts	All posts	Good Posts
NP	0.021	0.012	0.289	0.239
RR	0.183	0.319	0.269	0.474
DS	0.463	0.643	0.409	0.628
LDA	0.465	0.644	0.410	0.648
SWB	0.463	0.644	0.410	0.641
SMSS	**0.524**	**0.737**	**0.517**	**0.772**

All the three methods (LDA, SWB, and SMSS) achieve best average performance at $w_1 = 0.9$ while the w_2 is tuned for SWB and SMSS respectively. The experiment results are shown in Table 1.

From Table 1, we have four observations: (I) in Slashdot, a certain number of posts reply to the thread root, few to the nearest previous post; while in Apple Discussion, there are almost equal number of posts replying to the nearest previous post and the root. This is because: discussion threads in Apple Discussion follow a Question-Answering style. New solutions and fresh questions in replies invoke a serial of discussions. However, threads in Slashdot are usually initialized by a piece of news. Interesting aspects of the news and brilliant replies arise branches of discussions. (II) SWB and LDA show slight improvements to the baseline DS. This has verified our assumption that topics are robust in modeling the semantics; but topics are not capable enough to extract reply relations. (III) In our experiment SWB achieves best performance when w_2 is very small. This is because posts in threaded discussions are usually short. It is very difficult to estimate a sound coefficient for document specific word distribution. (IV) SMSS demonstrates significant improvement. The major difference between SMSS and former approaches is that SMSS resolves the structure representation $b^{(i)}$ for post p_i in each discussion thread. The best parameter of structural similarity is $w_2 = 0.9$. This indicates that, besides of semantic similarities, structure similarities are more distinguishing in identifying reply relations. Furthermore, we also analyze the performance for the posts of different quality. We define posts whose score is larger than 3 in Slashdot and the posts marked as "Helpful" or "Solved" in Apple Discussion as good posts. The similarity based methods have better performance for these posts with high quality. It makes sense since the posts with high quality may cause more significative replies.

4. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(6):993–1022, 2003.

[2] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. *Advances in newral information processing systems*, 41(6):391–407, 1990.

[2] http://discussions.apple.com/

[3] http://www.slashdot.org/