Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Memory-restricted latent semantic analysis to accumulate term-document co-occurrence events

Seung-Hoon Na [a,*], Jong-Hyeok Lee [b]

[a] *Dept. of Computer Science, National University of Singapore, Singapore*
[b] *Dept. of Creative IT Excellence Engineering & Future IT Innovation Laboratory, POSTECH, South Korea*

ARTICLE INFO

Article history:
Received 22 November 2011
Available online 11 May 2012

Communicated by Jie Zou

Keywords:
Co-occurrence
Dimensionality reduction
Partial-update algorithm
Latent semantic analysis

ABSTRACT

This paper addresses a novel adaptive problem of obtaining a new type of term-document weight. In our problem, an input is given by a long sequence of co-occurrence events between terms and documents, namely, a *stream of term-document co-occurrence events*. Given a stream of term-document co-occurrences, we learn unknown latent vectors of terms and documents such that their inner product adaptively approximates the target *query-based term-document weights* resulting from accumulating co-occurrence events. To this end, we propose a new incremental dimensionality reduction algorithm for adaptively learning a latent semantic index of terms and documents over a collection. The core of our algorithm is its *partial updating style*, where only a small number of latent vectors are modified for each term-document co-occurrence, while most other latent vectors remain unchanged. Experimental results on small and large standard test collections demonstrate that the proposed algorithm can stably learn the latent semantic index of terms and documents, showing an improvement in the retrieval performance over the baseline method.

1. Introduction

In this paper, we address the novel task of learning *query-based term-document weights*, often referred to as *query-based weights*. In our problem, the term-document weights of a term and a document are not provided explicitly; they are obtained indirectly from *term-query* and *document-query* weights. Instead of explicitly stating the target query-based term-document weights, we have a long sequence of term-document co-occurrence events, referred to as a *stream of term-document co-occurrence events*. Each co-occurrence event is described by two sets of terms and documents as evidence of a tighter relationship between them. Given a stream of term-document co-occurrence events, the objective of the problem is to gradually accumulate the co-occurrence events and learn a query-based term weighting metric for documents such that the weight of a term in a document is likely to be proportional to their co-occurrence rate.

A typical scenario for accumulating term-document co-occurrence events is presented in Algorithm 1, where a search engine continuously processes user queries online. In this scenario, each term-document co-occurrence event is defined for single retrieval. A term and a document are considered to have *co-occurred* if *the document* is *retrieved* in the top-ranked results by a query that includes *the term*.

Algorithm 1: Brief description of accumulating term-document co-occurrence

```
input: $m$ terms and $n$ documents in collection $\mathcal{C}$
    1. Initialization: $W_{ij} = 0$ for $1 \leqslant i \leqslant m$ and $1 \leqslant i \leqslant n$;
    2. Querying: Query $Q$ is provided by a user;
    3. Retrieval: Obtain top retrieved documents $\mathcal{F}$ for query $Q$;
    4. Update term weight for given $\mathcal{F}$ and $Q$;

for $t_i \in Q$ do
    for $d_j \in \mathcal{F}$ do
        $W_{ij} \leftarrow W_{ij} + \Delta$
    end
end
Iterate Step 2–4 until learning is stopped.
```

An obvious way to accumulate term-document co-occurrence events is simply to store all term weight values directly in an $m \times n$ term-document matrix, where each entry is assigned by a

* Corresponding author.
E-mail addresses: nash@comp.nus.edu.sg (S.-H. Na), jhlee@postech.ac.kr (J.-H. Lee).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.05.002

weight value, W_{ij}, between two objects. However, when the number of terms and documents is very large, the term-document matrix is high dimensional, which requires a less tractable manipulation that is not easily applicable.

To achieve better retrieval efficiency, we propose a novel algorithm called *memory-restricted latent semantic analysis* that effectively approximates target query-based term-document weights. Without maintaining a large-scale term-document matrix, our algorithm manages only low-dimensional *latent* vectors of terms and documents, and indirectly stores the target weight of a term in a document into the inner product between their latent vectors. In the proposed method, we first define the target query-based weights of terms in documents that are obtained by accumulating co-occurrence events. To restrict the memory capacity further, we then propose the use of a *partial-update criterion* that needs to be minimized, thereby modifying only a small number of latent vectors, called *focused latent vectors*, which are relevant to a given specific term-document co-occurrence event. Finally, we obtain the fixed-point iteration, which incrementally updates a set of focused latent vectors for each co-occurrence event.

Experimental results on small and large information retrieval (IR) test collections show that the proposed algorithm gradually and incrementally learns the target query-based term weighting metric from co-occurrence events, thereby improving the retrieval performance.

2. Related work

Dimensionality reduction has found extensive application in diverse areas, such as information retrieval (Dumais et al., 1988; Deerwester et al., 1990; Dumais, 1992; Bartell et al., 1992, 1995; Berry et al., 1995 Hofmann, 1999; Xu et al., 2003; Wei and Croft, 2006; Wang et al., 2011), computer vision (Levy and Lindenbaum, 2000; Brand, 2002), collaborative filtering (Hofmann, 1999, 2003; Koren et al., 2009), and data mining. Existing works have investigated singular value decomposition (SVD) (Dumais et al., 1988; Deerwester et al., 1990; Dumais, 1992; Berry, 1992; Berry et al., 1995; Berry et al., 1999; Levy and Lindenbaum, 2000; Brand, 2002; Wang et al., 2011), probabilistic latent semantic analysis (Hofmann, 1999, 2003), latent Dirichlet allocation (Blei et al., 2003), probabilistic principal component analysis (Tipping and Bishop, 1999; Lawrence, 2005), kernel principal component analysis (Schölkopf et al., 1997), non-negative matrix factorization (Lee and Seung, 1999, 2000; Berry et al., 2006), nonlinear dimensionality reduction (Roweis and Saul, 2000; Lawrence, 2005), and so on. Recent works have addressed the scalability issue so as to scale up the applicability of dimensionality reduction to large-scale (Yu et al., 2009; Wang et al., 2009) and, more recently, Web-scale data sets (Liu et al., 2010) over the distributed MapReduce framework (Dean and Ghemawat, 2008).

Although dimensionality reduction has mostly been studied in a batch mode, some existing works have developed incremental dimensionality reduction such that the algorithm can be scaled up to a larger size of data set and can also provide adaptability to environmental changes of co-occurrence events (Berry et al., 1995; Bartell et al., 1995; Levy and Lindenbaum, 2000; Brand, 2002). For example, Bartell et al. (1995) explicitly used co-relevance events among documents to define a novel static target inter-document similarity, and they applied multidimensional scaling (MDS) to approximate target similarities with latent semantic vectors. Brand (2002) also suggested an incremental method for updating SVD as a robust extension of Levy and Lindenbaum (2000)'s sequential updating algorithm of SVD. However, unlike our method, all previous incremental methods are an *full-update* solution; that is, for each co-occurrence event, all latent vectors for terms or documents should be updated. This contrasts with our approach, which suggests a *partial-update* algorithm in a memory-restricted manner.

To the best of our knowledge, no existing methods use a partial update algorithm, except for the work of Yu et al. (1985). Yu et al. (1985) proposed an approximation method for adaptively learning similarities among objects. For each object, Yu et al. (1985) introduced a one-dimensional latent vector, called the latent position, which is randomly initialized before learning the similarities. In (Yu et al., 1985)'s method for each co-occurrence event, the moving procedure on latent positions is conducted as follows: Given a set of objects that co-occur in a co-occurrence event, the latent positions of the objects are moved slightly toward their central positions such that the objects move slightly closer to each other after each latent position is moved. Thus, the moving procedure is continuously applied to all other co-occurrence events until the latent positions of the objects are finally converged. Although Yu et al. (1985)'s method was originally based on a one-dimensional latent space, it is quite simple to extend it to a multidimensional latent space.

Yu et al. (1985)'s method can be used to seemingly accumulate co-occurrence events, however, it suffers from the critical limitation; Yu et al. (1985)'s method likely falls into the collapsing problem. In other words, with an increase in the number of co-occurrence events that are processed, all the latent positions eventually tend to converge toward the same position such that the similarities (or distances) learned among objects are not distinguishable, causing all the objects to move into a single cluster.

3. Memory-restricted latent semantic analysis to accumulate term-document co-occurrence events

3.1. Target query-based term-document matrix

To describe our algorithm, we first need to define the target query-based term-document matrix. Suppose that a_{ij}^N is the target query-based weight of the ith term in the jth document obtained after processing the total number N of term-document co-occurrence events. Let q_N be the Nth co-occurrence event and $\mathcal{T}(q_N)$ and $\mathcal{F}(q_N)$ the sets of terms and documents that appear in an co-occurrence event q_N, respectively. For convenience, we also refer to $\mathcal{T}(q_N)$ and $\mathcal{F}(q_N)$ as $\mathcal{T}_N$ and $\mathcal{F}_N$, respectively.

Now, suppose that $f(i,j,q_N)$ is the increment of the query-based weight of the ith term in the jth document for the co-occurrence event q_N. In order to accumulate the co-occurrence events, the target term weight a_{ij}^N is updated according to the formula

$$a_{ij}^N = a_{ij}^{N-1} + f(t_i, d_j, q_N) \tag{1}$$

where a_{ij}^0 is set to zero for all (i,j) entries.

Suppose that we further decompose $f(i,j,q_N)$ into two weight parts, term-query weight function $g(t_i,q_k)$ and document-query weight function $h(d_j,q_k)$. Eq. 1 is rewritten as

$$a_{ij}^N = a_{ij}^{N-1} + g(t_i, q_N)h(d_j, q_N) \tag{2}$$

Here, the target weight differs depending on the definition of $g(t_i,q_k)$ and $h(d_j,q_k)$. This paper considers the following setting for $g(t_i,q_k)$ and $h(d_j,qk)$ given as

$$\begin{aligned} g(t_i, q_k) &= P(t_i|\theta_{q_k}) \\ h(d_j, q_k) &= \mathcal{I}(d_j \in \mathcal{F}(q_k)) \end{aligned} \tag{3}$$

where $\mathcal{I}(e)$ is the indicator function that returns 1 when expression e is true, or otherwise 0, θ_{q_k} is the *query language model* for q_k, and $P(t_i \mid \theta_{q_k})$ is the generative probability of term t_i from the query model θ_{q_k}. To specify the query model θ_{q_k}, we might use either maximum likelihood estimation (MLE) or a relevance model (RM)