Future Generation Computer Systems 86 (2018) 1503–1512

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Incremental term representation learning for social network analysis

Hao Peng [a,1], Mengjiao Bao [a,1], Jianxin Li [a,*], Md Zakirul Alam Bhuiyan [b], Yaopeng Liu [a], Yu He [a], Erica Yang [c]

[a] School of Computer Science and Engineering, Beihang University, Beijing, China
[b] Department of Computer and Information Sciences, Fordham University, NY, USA
[c] Scientific Computing Department, STFC Rutherford Appleton Laboratory, Oxfordshire, UK

HIGHLIGHTS

- An incremental matrix factorization model designed for term representation.
- An incremental term representation learning method for social network analysis.
- The model convergence is proved based on stochastic gradient method.
- Experiments conducted on word similarity, label classification and user clustering.

ARTICLE INFO

Article history:
Received 15 December 2016
Received in revised form 20 March 2017
Accepted 14 May 2017
Available online 23 May 2017

Keywords:
Term representation
Incremental learning
GloVe model
Social network analysis

ABSTRACT

Term representation methods as computable and semantic tools have been widely applied in social network analysis. This paper provides a new perspective that can incrementally factorize co-occurrence matrix to query latest semantic vectors. We divide the streaming social network data into old and updated training tasks respectively, and factorize the training objective function based on stochastic gradient methods to update vectors. We prove that the incremental objective function is convergent. Experimental results demonstrate that our incremental factorizing can save a substantial amount of time by speeding up training convergence. The smaller the updated data is, the faster the update factorizing process can be, even 30 times faster than existing methods in certain cases. To evaluate the correctness of incremental representation, social text similarity/relatedness, linguistic tasks, network event detection, social user multi-label classification and user clustering for social network analysis are employed as benchmarks in this paper.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Social network analysis is the mining, measuring, and representation of relationships and flows between people, groups, organizations, computers, URLs, and other connected information/knowledge entities [1,2]. In essence, social influence propagation, user behavior analysis and modeling community influence, can be attributed to social network text and structure. Therefore, text and structure representation learning methods [3–8], capable to capture a variety of latent semantic features from social network analysis scenarios s, including social text similarity/relatedness and word analogy [7,9,10], social event detection [11–13], information retrieval, document classification [14–17], social network structure analysis, user multi-label classification and user clustering tasks [5,6], have become significant tools for social network analysis.

For most social network analysis models, an outstanding issue remains. Despite their focus on static social co-occurrence relationships, they lack the textual information related to users. There are numerous social network applications such as social influence and user analysis, which require incremental update of the text and structure representation so as to keep pace with the fast evolving in working domains. For example, the semantic distance between '*Leonardo DiCaprio*' and '*Kate Winslet*'[2] increases after they each produce many movies with other actors/actresses. There are two typical representation learning methods, namely Neural-Network

* Corresponding author.
E-mail addresses: penghao@act.buaa.edu.cn (H. Peng), baomj@act.buaa.edu.cn (M. Bao), lijx@act.buaa.edu.cn (J. Li), mbhuiyan3@fordham.edu (M.Z.A. Bhuiyan), liuyp@act.buaa.edu.cn (Y. Liu), heyu@act.buaa.edu.cn (Y. He), erica.yang@stfc.ac.uk (E. Yang).

[1] Note: Hao Peng and Mengjiao Bao contributed equally to this work.

[2] Leonardo DiCaprio and Kate Winslet were the heroes of the famous film Titanic.

http://dx.doi.org/10.1016/j.future.2017.05.020
0167-739X/© 2017 Elsevier B.V. All rights reserved.

Models [6,7,9], such as Skip-gram (SG) and CBOW, and Non-negative constraints Matrix Factorization Models (NMFMs) [3,4], like GloVe model. But neither is capable for dynamic social network analysis. This is because the latent dimension for which social representation has is typically too big to cope with for dynamic evolution. Furthermore, when the changes of semantic and social network structure are relatively small compared with the existing one, these methods are inefficient to retrain with the updated social network data.

This paper jointly learns hybrid representation form text and social network and incrementally learns the representation for dynamic social network data. In particular, the popular GloVe [3,4,18] tool is adopted, due to its relatively high interpretability and time efficiency than other matrix factorization models [18–20] and Neural-Network Models. SG and CBOW models depend on complicated neural network parameters, which are adverse to decompose the objective function incrementally [8]. In our method, social network semantic and structure representation learning share the unique object function of GloVe. When using GloVe to generate term vectors, there is a preprocessing step of building a global statistical term context co-occurrence matrix [3,4,18,19,21,22]. More precisely, it uses co-occurrence counts to construct matrix over the whole social network data, and conducts weighted least squares model [3,18,21] to train on the matrix. When the frequency of terms is changed, term co-occurrence matrix will be updated accordingly. To tackle this problem, we retain the dimension of the matrix and perform incremental iterations on the added values between old and new matrices. When updating vectors for the raised weighting function, we follow the original matrix factorization algorithms in GloVe based on stochastic gradient descent method. When updating the vectors for the reduced weighting function, stochastic gradient ascent is used to modify changes. In this way, only the second incremental iteration is performed while all other vectors remain identical to those trained based on the old data. Since the updating process is independent of all bias parameters and semantic term vectors, the parallel incremental term representation method based on GloVe has been successfully developed. Our system is publicly available at https://github.com/RingBDStack/Incremental-GloVe.

Contributions. To solve this complex problem, we design an incremental matrix factorization model for term co-occurrence matrix to provide evolving representation for social network analysis. Specifically, we make the following contributions in this paper:

(1) We propose a novel incremental model, which is based on matrix factorization method, to satisfy the analysis requirements for dynamic social network. The model achieves precisely decomposing renewal objective function for feature embeddings.

(2) We also put forward an incremental algorithm to realize the term representation, and prove the convergent of the algorithm based on stochastic gradient method. Not only the weight function enlarges with the updated social network streaming, but all parameters can be inherited.

(3) We conduct extensive experiments on social network data representing that incremental term representation learning model, for updating features, obtains great acceleration. The smaller the updated social network data is, the faster the update training process can be, even 30 times faster than before in certain cases.

Experimental results show that we have achieved almost the same term embeddings as fully re-trained GloVe in different benchmark tasks. In order to demonstrate the correctness of this model, we examined both individual vectors' cosine similarity and downstream society network applications, such as word similarity/relatedness, linguistic tasks and social event detection for word representations, and user multi-label classification and complementary visualized social clustering tasks for user representations. In addition, we proved that the incremental objective function has the characteristic of convex optimization in convergence analysis.

Our paper contains four further sections. Section 2 surveys related work on term representation learning and incremental learning used in social network analysis. Section 3 investigates the background of matrix factorization based GloVe model. Section 4 explains the principle of incremental GloVe model and convergence analysis. Section 5 presents the experimental results, analysis, and discussion with existing global and incremental models.

2. Related work

In this paper, we consider the problem of training social network term representation based on new data incrementally. In particular, the popular unsupervised matrix factorization GloVe model is adopted due to its interpretable and time efficiency, and quality performance to other representation learning models [4–7,9,23]. To handle term evolution, we explain the principle of incremental learning problem [8,24–28].

Representation learning. In general, representation learning models adopt two typical unsupervised learning techniques, namely Neural Network Language Models [6,7,9] and Non-negative Constraints Matrix Factorization Models [3,4,18–20], so as to speed up the process of feature learning and querying. Hierarchical softmax was first proposed by Mnih and Hinton [10] where a hierarchical tree is constructed to index all the words in a corpus as leaves, while negative sampling is developed based on noise contrastive estimation [21,29], and randomly samples the words not in the context to distinguish the observed data from the artificially generated random noise. It is empirically shown that hierarchical softmax performs better in the case of infrequent words while negative sampling performs better in case of frequent words [7,9]. Negative sampling uses random sampling to sample the non-context words, which is more likely to emphasize the frequent words [7,9]. The Non-negative Constraints Matrix Factorization Models for representing learning like GloVe, via stochastic gradient methods, factorizes the original matrix for learning effective representation that outperforms other models for text and network structure.

Incremental learning. Incremental learning is a machine learning paradigm where the learning process takes place whenever new example(s) emerge and adjusts what has been learned according to the example(s) [8,24,25,30]. The most prominent difference between incremental learning and traditional machine learning is that the former does not assume the availability of a sufficient training set before the learning process, but the training examples appear over time [24–26]. In fact, the desirable way to cope with such situations is to enable the object function to learn incrementally through updating the current model in accordance with the newly arriving data. Incremental Matrix Factorization models, such as incremental Singular Value Decomposition (ISVD) [28], incremental Regularized Matrix Factorization (IRMF) [27] have significant improved accuracy and scalability in online recommendation system. Compared with existing incremental matrix factorization methods [28], we also take into consideration the error boundedness analysis and the convergence of incremental objection function in the weighted least squares regression based matrix factorization model, GloVe, for scalability and robustness.

3. Background

This section introduces the background of the GloVe model based on factorizing the term co-occurrence matrix. It is supposed that the term co-occurrence matrix has been constructed from a given social network task data set $\mathcal{W}$, where all unique terms in $\mathcal{W}$ have unigram and uniform distributions.

3.1. The GloVe model

In the GloVe model, given the term co-occurrence matrix X constructed by task data $\mathcal{W}$, the training objective is to minimize the average log-likelihood function.

$$\mathcal{J} = \sum_{i,j=1}^{V} f_0(X_{ij})(W_i^T \tilde{W}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2. \tag{1}$$

Where X_{ij} is the number of co-occurrences between term w_i and w_j, V as the size of training matrix built by $\mathcal{W}$. W_i^T and $\tilde{W}_j$ are semantic vectors, b_i and $\tilde{b}_j$ are bias vectors related to term and context respectively. And f_0 is a down-weighting rare co-occurrence function [3] whose formalization is

$$f_0(x) = \begin{cases} (x/x_{max})^{\alpha}, & x < x_{max} \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

The performance of the model hardly depends on the cutoff, which is fixed according to artificial experience of x_{max} for all experiments. Moreover, when $\alpha = 3/4$ shows modest improvement, it is the same coefficient with NNLMs [4–6,9,23]. The weighting function obey the following properties:

- $f_0(0) = 0$. Weighting function is a continuous function, so it vanishes as $x \to 0$ fast enough that the $\lim x_{x\to 0} f_0(0) log^2 x$ is finite.
- Weighting function will non-decreasing so that rare co-occurrences will not be over-weighted.
- Weighting function will be relatively small for large values of x, so that frequent co-occurrences are not over-weighted. So when x exceed the threshold x_{max}, weighting function $f_0(x)$ will be the upper bound 1.

Using stochastic gradient descent, the bias vectors b_i and $\tilde{b}_j$, as well as term vectors W_i and $\tilde{W}_j$ in the context can be updated as follows:

$$\begin{aligned} W_i^T &:= W_i^T - \eta f(X_{ij})(W_i^T \cdot \tilde{W}_j + b_i + \tilde{b}_j - \log X_{ij})\tilde{W}_j \\ \tilde{W}_j &:= \tilde{W}_j - \eta f(X_{ij})(W_i^T \cdot \tilde{W}_j + b_i + \tilde{b}_j - \log X_{ij})W_i^T \\ b_i &:= b_i - \eta f(X_{ij})(W_i^T \cdot \tilde{W}_j + b_i + \tilde{b}_j - \log X_{ij}) \\ \tilde{b}_j &:= \tilde{b}_j - \eta f(X_{ij})(W_i^T \cdot \tilde{W}_j + b_i + \tilde{b}_j - \log X_{ij}). \end{aligned} \tag{3}$$

Where η is a degenerative learning rate.

3.2. Learning rate

The learning rate η is an important parameter of stochastic gradient iteration [31]. In the GloVe model, an adaptive rate using AdaGrad [32,33] is set to be:

$$\Delta\eta_\tau = -\frac{\eta}{\sqrt{\sum_{\tau=1}^{t} g_\tau^2 + \epsilon}}. \tag{4}$$

Where η is the initial learning rate, and g_τ is the gradient in τ round of iteration. In order to avoid the circumstance where the denominator value is zero, ϵ is initialized to be a very small positive number. The learning rate is governed by the accumulation of gradients $\sum_{\tau=1}^{t} g_\tau$, which controls the rate to decrease η_τ after a certain number of iterations, e.g., updating η_τ after every iteration. In GloVe, a minimum value $\eta_{\min}$ is also set to enforce the update vectors based on the gradients.

4. Incremental learning model

It can be learned from the above sections that learning term vectors involve not only the term vectors themselves, but also the bias vectors, which rely on the decomposition of the term co-occurrence matrix. When new data is added to the old one, it is necessary the re-build the decomposition matrix based on the combined data, as shown in Fig. 1. Term co-occurrence matrix is sensitive to the term frequency's distribution, and the increased data will affect the determination of x_{max}. Therefore, the change in matrix may influence the vector representations of both term and bias. We decompose the objective function into an old matrix and an incremental matrix, then distinguish the update between parts from old and new social network data.

4.1. Notations and definitions

In this section, we illustrate the detailed notations and definitions of basic concepts for incremental decomposition of term co-occurrence matrix for vector representations.

Basic Concepts. Given an original term co-occurrence matrix $\mathcal{X}$ and an objective function $\mathcal{J}$, where $f_0(x)$ is the weighting function and x_{max} represents maximum threshold in $\mathcal{W}$. More importantly, all term representation vectors and bias vectors are W_i^T, $\tilde{W}_j$, b_i, $\tilde{b}_j$. The incremental matrix factorization problem is to re-build renewed term co-occurrence matrix, and retain all machine learning parameters for iterative training. To be more precise, all symbols (functions) and explanations are summarized in Table 1.

4.2. Matrix initialization and inheritance

Suppose we have old data $\mathcal{W}$ and new data $\mathcal{W}' = \mathcal{W} \cup \Delta\mathcal{W}$. We can then build the term co-occurrence matrices $\mathcal{M}$ and $\mathcal{M}'$ respectively. We fix the size of term co-occurrence matrix by V, and initialize its value to zero. Concerning the matrix's value, if the word has not been observed in corpus, it is simply initialized to zero. When increasing data, new matrix's value with $X_{ij'} = X_{ij} + \Delta X_{ij}, \quad i, j = 1 \cdots V$ will be formalized. If the term is new, then it will be randomly initialized as a random vector:

$$W_i' = \begin{cases} W_i, & w_i \in \mathcal{W} \\ \text{random}, & w_i \notin \mathcal{W}, \end{cases} \tag{5}$$

where W_i is the vector of term w_i for old and new matrices respectively. Similarly, new $\tilde{W}_j$, b_i, $\tilde{b}_j$ vectors and bias parameters are also initialized as random values.

4.3. Model updates

Given the term vectors and bias parameters by comparing old and new data, we also decompose the log-likelihood functions for GloVe models. For GloVe model, we consider to factorize log-likelihood function by aggregating the cost term $f_0(X_{ij})(W_i^T \tilde{W}_j^+ b_i + \tilde{b}_j - \log(X_{ij}))^2$ in Eq. (1).

$$J' = \sum_{i,j=1}^{V} f_1(X_{ij} + \Delta X_{ij})(W_i'^T \tilde{W}_j' + b_i + \tilde{b}_j - \log(X_{ij} + \Delta X_{ij}))^2. \tag{6}$$

Here we first split the weighting function by $f_1(X_{ij} + \Delta X_{ij}) = f_0(X_{ij}) + \hbar$, we assume that the term distribution will not mutate, so $\hbar = (x'/x_{max})^{3/4}$ is a small variation. As to the words in $\mathcal{W}$, it will be factorized based on the common iterative function:

$$J' = \mathcal{J} + \sum_{i,j=1}^{V} f_0(X_{ij})(L^2(X_{ij}) - 2\dot{M}L(X_{ij})) + \sum_{i,j=1}^{V} \hbar(M - L(X_{ij}))^2 \tag{7}$$

where $M = W_i'^T \tilde{W}_j' + b_i + \tilde{b}_j - \log(X_{ij})$ and $L(X_{ij}) = \log(1 + \Delta X_{ij}/X_{ij})$.

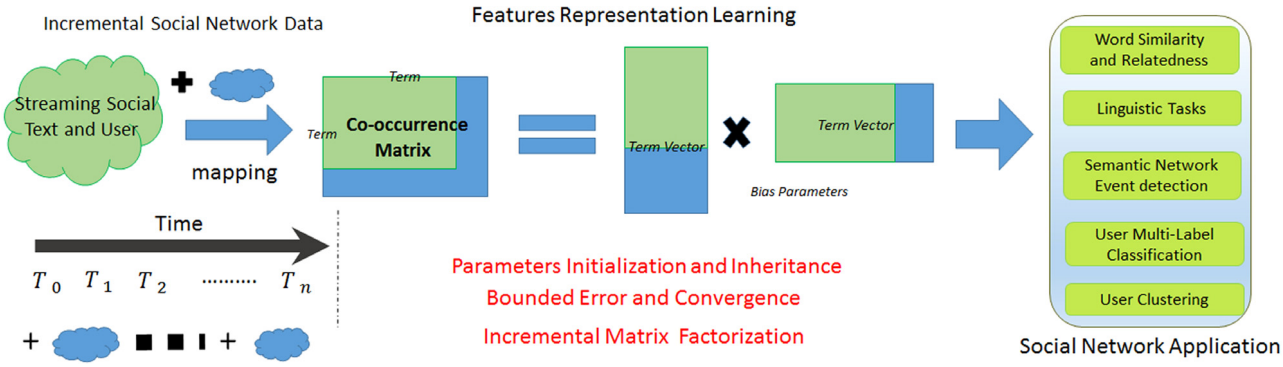


Fig. 1. Illustration of incremental term representation learning in social network data.

Table 1
Symbol notations.

Symbol	Explanation
$\mathcal{W}'$	The renewed social network data
$\Delta\mathcal{W}$	The incremental social network data
X_{ij}	The renewed term w_i and w_j co-occurrence matrix value
ΔX_{ij}	The incremental term w_i and w_j co-occurrence matrix value
W_i'	The renewed term w_i representation vectors
W_j'	The renewed term w_j representation vectors
b_i	The renewed bias of term parameter of w_i
b_j'	The renewed bias of term parameter of w_j
J'	The renewed object function in social network data $\mathcal{W}'$
x'_{max}	The renewed maximum threshold
$f_1(X_{ij}')$	The renewed weighting function of X_{ij} in co-occurrence matrix value

To train a new set of term vectors, it is necessary to re-scan and re-train the whole data $\mathcal{W}' = \mathcal{W} \cup \Delta\mathcal{W}$ based on stochastic gradient descent in the first place. Given the above factorization analysis of the objective function, it can be inferred that the following approach can be applied to the old data $\mathcal{W}$ in order to save training time. Our goal is to find a new set of (local) optimal term vectors W_i and bias parameters b_i to approximate the re-training results. We first assume that all term vectors W_i are already (local) optimal and can thus further calibrate them. Then stochastic gradient based optimization based on $\mathcal{M}$ and $\Delta\mathcal{M}$ respectively is performed. During the training of the incremental matrix $\Delta\mathcal{M}$, all vectors and bias parameters can be updated according to the following formulas:

$$\begin{aligned}
W_i'^T &:= W_i'^T - \eta' f(X_{ij}) M \tilde{W}_j' + \eta' f(X_{ij}) L(X_{ij}) \tilde{W}_j' \\
&\quad - \eta' \hbar(X_{ij} - L(X_{ij})) \tilde{W}_j' \\
\tilde{W}_j' &:= \tilde{W}_j' - \eta' f(X_{ij}) M W_i'^T + \eta' f(X_{ij}) L(X_{ij}) W_i'^T \\
&\quad - \eta' \hbar(X_{ij} - L(X_{ij})) W_i'^T \\
b_i' &:= b_i' - \eta' f(X_{ij}) M + \eta' f(X_{ij}) L(X_{ij}) - \eta' \hbar(X_{ij} - L(X_{ij})) \\
\tilde{b}_j' &:= \tilde{b}_j' - \eta' f(X_{ij}) M + \eta' f(X_{ij}) L(X_{ij}) - \eta' \hbar(X_{ij} - L(X_{ij}))
\end{aligned} \quad . \tag{8}$$

where η' is a new incremental degenerative learning rate.

In increment training, the following iterative formula for term vectors and bias parameters is performed.

$$\begin{aligned}
W_i'^T &:= W_i^T + \eta' (f(X_{ij}) L(X_{ij}) - \hbar(X_{ij} - L(X_{ij}))) \tilde{W}_j' \\
\tilde{W}_j' &:= \tilde{W}_j + \eta' (f(X_{ij}) L(X_{ij}) - \hbar(X_{ij} - L(X_{ij}))) W_i'^T \\
b_i' &:= b_i + \eta' (f(X_{ij}) L(X_{ij}) - \hbar(X_{ij} - L(X_{ij}))) \\
\tilde{b}_j' &:= \tilde{b}_j + \eta' (f(X_{ij}) L(X_{ij}) - \hbar(X_{ij} - L(X_{ij}))).
\end{aligned} \tag{9}$$

The procession of adding social network data into W′, updating term co-occurrence frequency X′ in matrix M′, and correctly factorizing the objective function J' and conduct incremental iterations is indicated in Algorithm 1, which includes all parameters initialization and inheritances, as well as updating function model and incremental parameter iterations.

Algorithm 1 Incremental GloVe

Input: For each data drawn from $\Delta\mathcal{W}$

1: Sequence of w_i training examples

2: Original term co-occurrence matrix X_{ij};

3: Weak learning rate η and the threshold of x_{max}

4: Save all the term vector and bias parameters W and b.

Output: Incremental parameters W, $\tilde{W}$ and b.

5: **function** Incremental GloVe($\Delta\mathcal{W}, X_{ij}, \eta, W, b$)

6: Count all co-occurrence value in $\Delta\mathcal{W}$, set $X_{ij} = X_{ij} + \Delta X_{ij}$, $i, j = 1 \cdots V$;

7: Inherit all term and bias vector, W and b, and random initialized new term and bias vector.

8: Update the threshold from x_{max} to x'_{max}, so set $x'_{max} = x_{max} + \Delta x_{max}$

9: Decompose the weight function $f_1(X_{ij} + \Delta X_{ij}) = f_0(X_{ij}) + \hbar$

10: Factorize the object function $\mathcal{J}' = \mathcal{J} + \Delta J$

11: Decompose the incremental object function $\Delta J = F(M, W', b', \widetilde{W}', \tilde{b}')$

12: Call the stochastic gradient descent on $\Delta J = argminF(M, W', b', \widetilde{W}', \tilde{b}')$

13: Conduct the stochastic gradient descent for $W', b', \widetilde{W}', \tilde{b}'$

14: **return** $W', b', \widetilde{W}', \tilde{b}'$

15: **end function**

4.4. Convergence analysis

This section presents the theoretical analysis of the incremental learning algorithm by using the GloVe model. The log-likelihood functions in Eqs. (6) and (7) are positive. Thus, the minimizing of objective is limited by zero. However, since the objective function involves the dot product of term vectors and summation of bias parameters and matrix values, it is a non-convex optimization problem. By using an alternative optimization to alter term vectors and bias parameters, fixing one and optimizing the other is a convex problem. In this incremental decomposition, the convergence of optimizing over the updated data $\Delta\mathcal{W}$ is the same as it is in the original GloVe model. When optimizing old data, we assume term vectors are already (local) optimal, and can thus optimize parameters over all data on this foundation. That means we have:

$$\nabla^2_{W_i'^T} = \sum_{i,j=1}^{V} 2f_1(X_{ij} + \Delta X_{i,j})(\tilde{W}_j' \cdot \tilde{W}_j'^T) \tag{10}$$

where $f_1 \in [0, 1]$ and $\tilde{W}_j' \cdot \tilde{W}_j'^T \geq 0$ by checking the second order derivative of the term vectors and bias parameters in incremental GloVe model.

Compared with the original second order derivative over the old data, variation of $f_0(X_{ij})$ by $f_1(X_{ij} + \Delta X_{ij})$ is added. This is guaranteed by the use of stochastic gradient descent in Eqs. (8) and (9), thus leading the process towards another local optimum.

$$\nabla^2_{\tilde{W}_j'} := \sum_{i,j=1}^{V} 2f_1(X_{ij} + \Delta X_{i,j})(\tilde{W}_i'^T \cdot \tilde{W}_i') \tag{11}$$

$$\nabla^2_{b_i} := \nabla^2_{\tilde{b}_j} = 2.$$

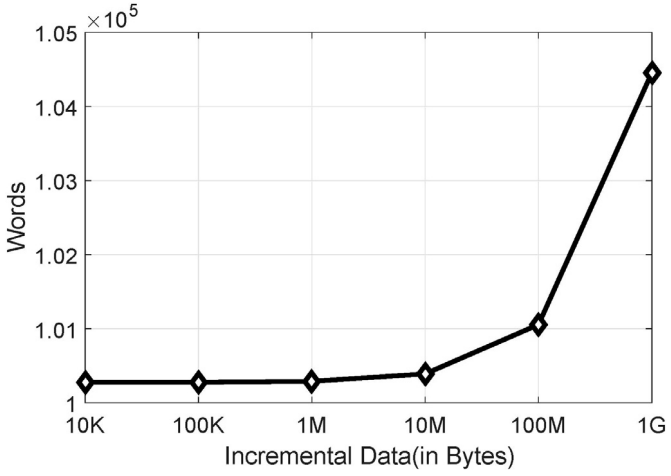
Hence the $\nabla^2_{\tilde{W}_j'}$, $\nabla^2_{b_i}$ and $\nabla^2_{\tilde{b}_j}$ have similar convergence properties.

5. Experiments

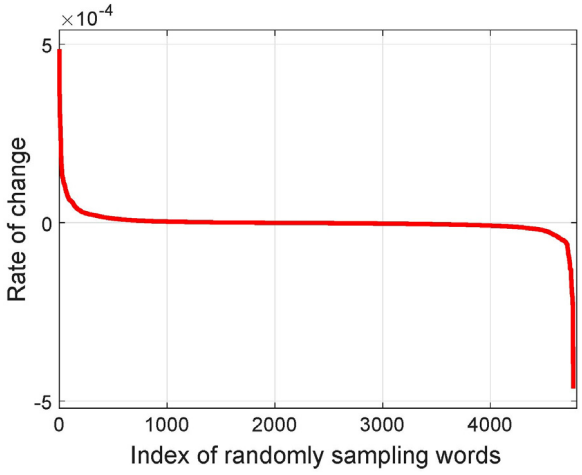
Experiments are conducted in this section to demonstrate the effectiveness and efficiency of the incremental factorizing for word representations and user representations in network big data analysis tasks. We first evaluate time and quality of the vectors by comparing global factorization with incremental factorization then text and network user analysis tasks, i.e., word similarity/relatedness, linguistic tasks, social event detection, multi-label classification and user clustering to evaluate the term vectors.

5.1. Training time and quality

In order to build term co-occurrence matrices, streaming social text data is extracted as the source and the data is divided into several temporal data sets. Then we use 2 GB data set as the initial data, which is the old one in previous sections. The 2 GB data contains 474,746,098 tokens and 100,278 unique words. Next, we select 10 KB, 100 KB, 1 MB, 10 MB, 100 MB, and 1 GB data as new update data to compare the performances of algorithms. We fix the size of co-occurrence matrix by 104740 × 104740, and initialize it by zero. The number of words arising with new updated data is shown in Fig. 2(a). In a limited range, arising words emerge log-scale in accordance with the log-scale of incremental streaming data. Then, we check the change of the distribution of weight function $\hbar = f_1(X_{ij} + \Delta X_{ij}) - f_0(X_{ij})$ is checked. Fig. 2(b) shows the changing rate of the distribution of weight function $\hbar$ that are affected by the incremental data when 2 GB data has 1 GB incremental data and x_{max} increases from 40 to 49. It should be noted that x-axis represents the index of randomly sampling words and y-axis represents $\hbar$. The change of the distribution of weight function shows the tendency of both ascent and descent. In addition, it is a normal phenomenon that increase and decrease in the change of weight function are balanced.



(a) The changing quantity of words affected by the incremental data.

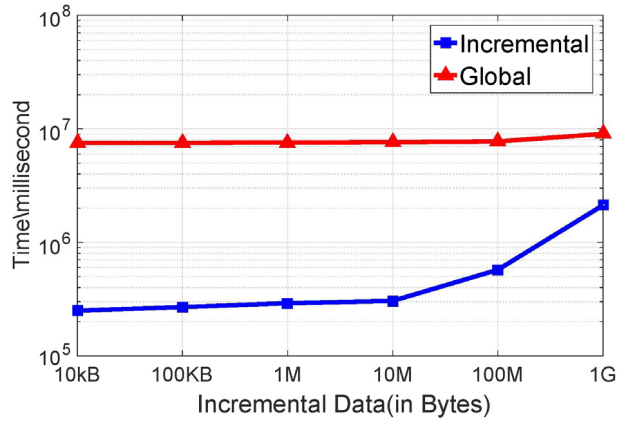


(b) The change of weighting affected by the incremental data.

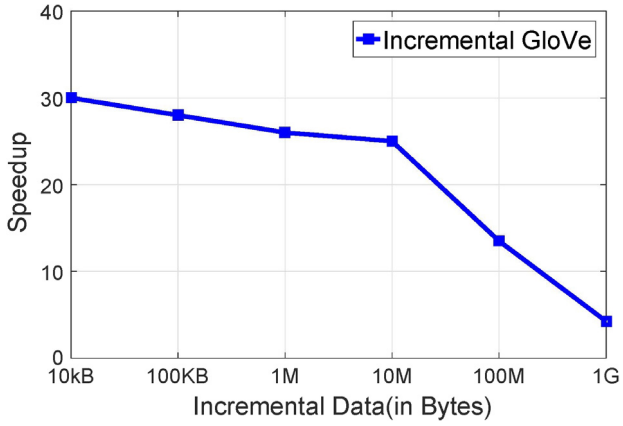
Fig. 2. Words change with incremental corpus.

For original global matrix factorizing, old data with new data is combined and run by the original the GloVe model. For incremental training, we use the model trained based on the 2 GB initial training corpus, and then run our algorithm to update the words co-occurrence as well as the word vectors and bias parameters. All experiments mentioned above are run with 10 CPU threads and generate word embeddings with 300 dimensions.

Then we check the training time and speedup by using our incremental factorizing algorithm. The results of the training time can be seen in Fig. 3(a). It indicates that the time curve of GloVe is linear with the training size. Since the adding of data from 10 KB to 100 MB is relatively small compared with the original training size 2 GB, the time curve for GloVe with global training is flat until 1 GB of additional training data is added. Furthermore, incremental training for GloVe benefits from the algorithm and is faster than that of the global training. Again, the scale is linear with the number of additional updated data. The results of speedup are shown in Fig. 3(b). It can be seen that the speedup is more significant in the case of smaller update corpus. The GloVe model can achieve up to 30 times speedup with this incremental training algorithm.

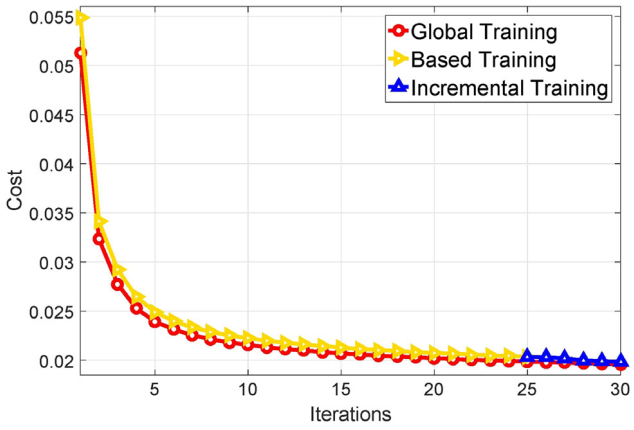


(a) Time performance by comparing global and incremental factorizing.

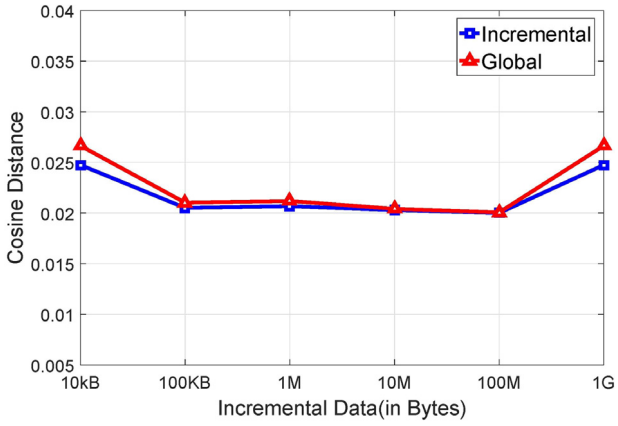


(b) Speedup performance by comparing global and incremental factorizing.

Fig. 3. Training performance of global and incremental training.



(a) Cost comparison between global and incremental factorizing.



(b) Cosine distance between global and incremental factorizing.

Fig. 4. Training performance and cosine distance of global and incremental word representations.

To further understand the speedup of increment, the cost function at each epochs between global training and incremental training is shown in Fig. 4(a) when scanning the 2 GB old data and new increased 1 GB of data. It can be seen that the number of updates exponentially decreases with the increase of epochs. As to the top incremental training, the extension curve indicates that there is indeed significant speedup by inheriting from previous learned term and bias vectors. It can be observed that incremental factorizing can be convergent after iterating five times, while the global training has to iterate 30 times to achieve same performance. The *x*-axis represents training time, and *y*-axis represents cost of objection function (1 6). The convergence of cost almost reaches 0.02. There is no obvious cost boundary between incremental and global training for future training indicating that the former is faster than the latter.

We also randomly select 5000 word vectors to test the differences among word vectors. Furthermore, cosine similarity is used to evaluate differences between these two sets of word vectors. For global training, the same algorithm is run twice, using different random initializations. As to the incremental training, the results of incremental training are compared with those of global training. It can be observed from Fig. 4(b) that the cosine distance between global training and incremental training is approximate and comparable. Both curves are extremely close, and the differences between incremental training vector and global training vector belong to the scale of 10^{-3} in cosine distance.

5.2. Word similarity/relatedness

As mentioned in previous sections, word similarity/relatedness evaluation is used as a benchmark to evaluate the correctness of our incremental training algorithm. Specifically, we use the data set collected by Faruqui and Dyer [34] which includes MC-30, TR-3k, MTurk-287, MTurk-771, RG-65, RW-STANFORD (RW), SIMLEX-999, VERB-143, WS-353-ALL, WS-353-REL, WS-353-SIM, and YP-130.[3] Cosine value is employed to compute the similarities between words, as well as to rank the words similar/related to each other. The Spearman's rank correlation coefficient [35] is used to check the correlation of ranks between human annotations and computed similarities. Due to limited space, only the results trained over 10 KB, 100 KB, …, 1 GB update data are shown. From Fig. 5, it can be observed that the results of incremental training are comparable and sometimes better than the results of global training.

5.3. Linguistic tasks

Other downstream linguistic tasks [36,37] are also tested. Different from word similarity/relatedness tasks, linguistic tasks consist of questions, for example, "a is to b as c is to ?", whose correlations are measured by human ratings. Thus, it is more

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

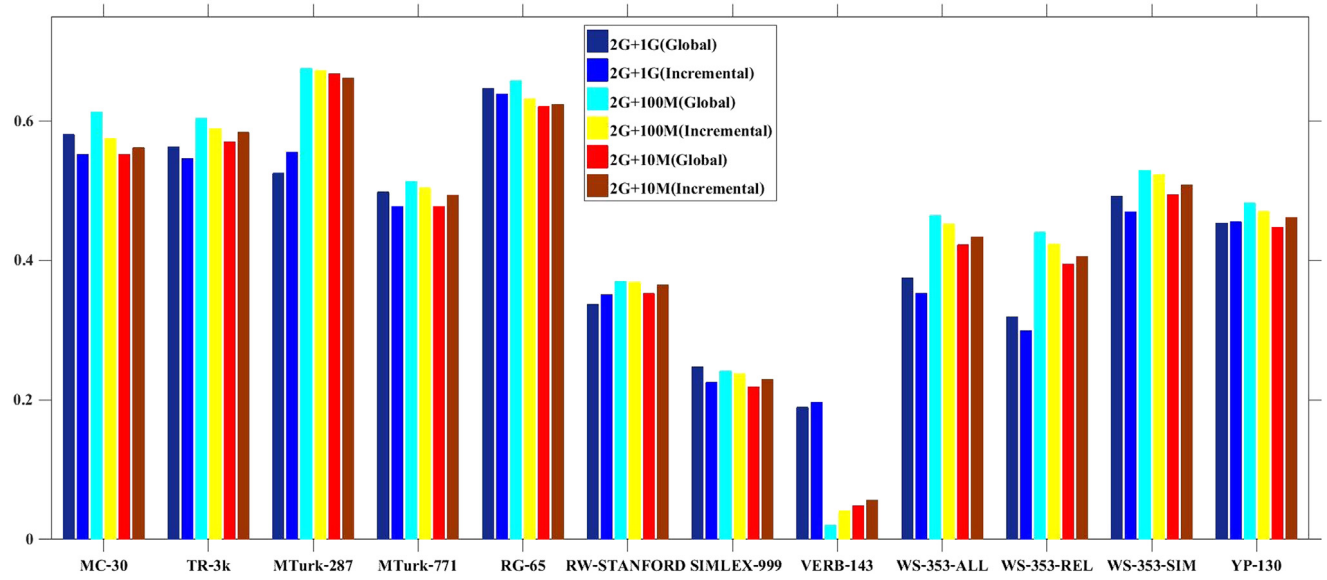


Fig. 5. Training performance and cosine distance of global and incremental word representations.

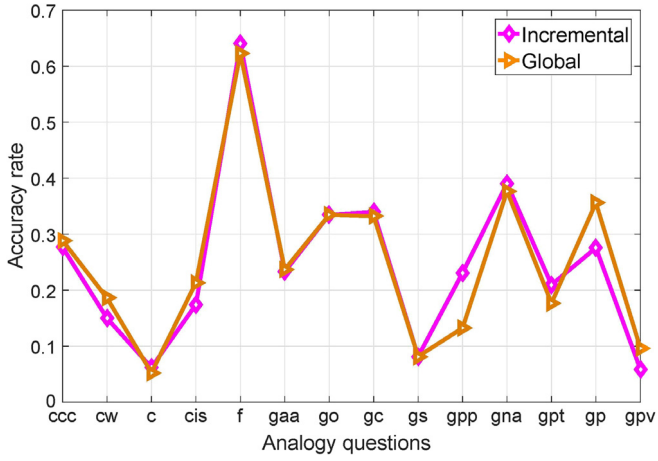


Fig. 6. Linguistic tasks comparison of global and incremental training in word representations.

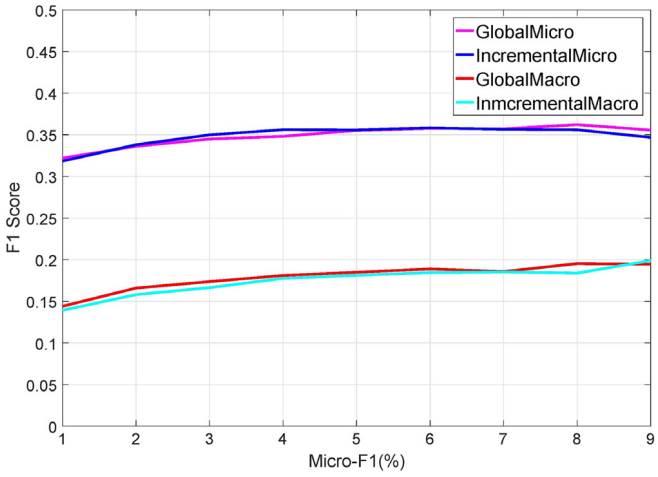


Fig. 8. Multi-Label classification comparison of global and incremental training in user representations.

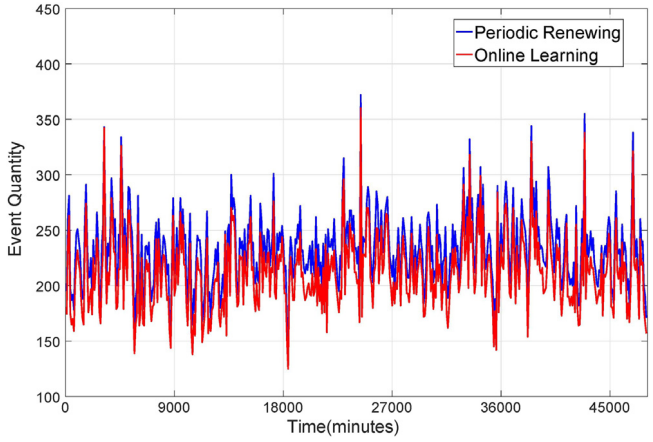
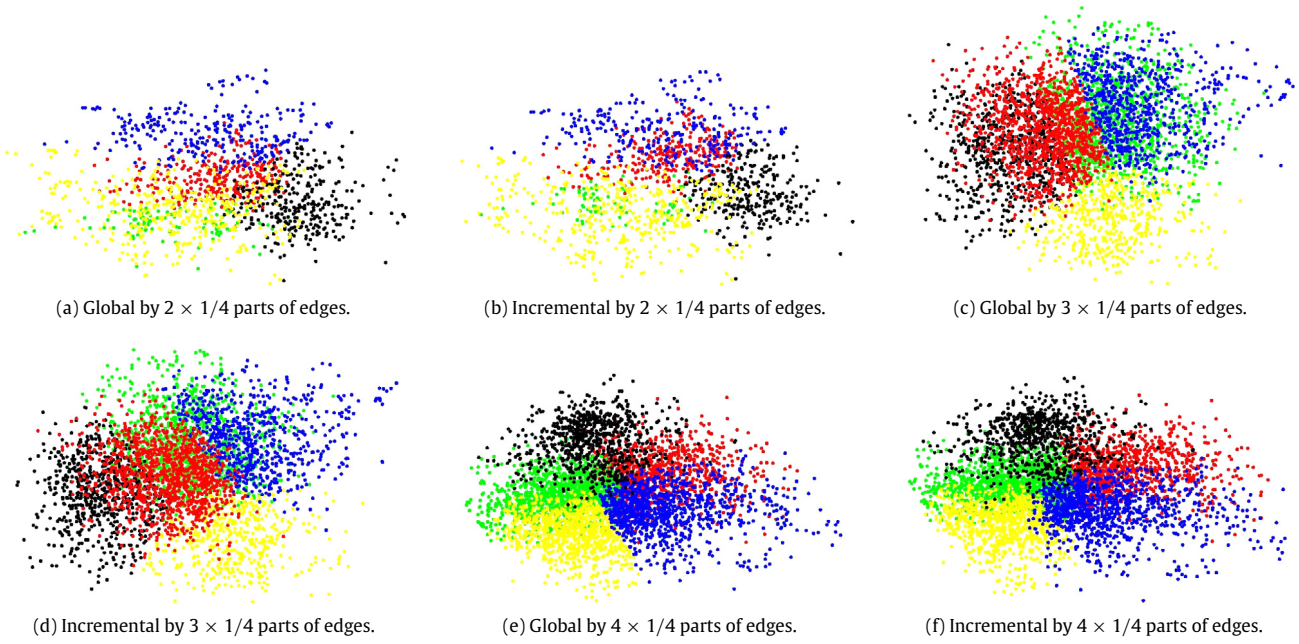


Fig. 7. Semantic event detection comparison of global and incremental training in word representations.

complicated than simply comparing similarity/relatedness between words. We conduct linguistic tasks' experiments based on Syntactic data set [6] containing 19544 semantic and syntactic subset questions, to answer corresponding words based on the word embeddings produced by our experiments. The analogy questions include capital-common-countries(ccc), capital-world (cw), currency(c), city-in-state(cis), family(f), gram1-adjective-to-adverb(gaa), gram2-opposite(go), gram3-comparative(gc), gram4-superlative(gs), gram5-present-participle(gpp), gram6-nationality-adjective(gna), gram7-past-tense(gpt), gram8-plural (gp), gram9-plural-verbs(gpv), which are answered by using Levy and Goldbergs similarity multiplication method [4]. Following [3,4], the incremental processes is tested. The results are shown in Fig. 6. It can be seen that the incremental training and global training show similar results, and incremental training possesses a slightly better performance than global training. This demonstrates again that the performance of incremental training is comparable to global training. Furthermore, we remark that the performance of our incremental method continuously improves when more training data is incorporated.



(a) Global by 2 × 1/4 parts of edges. (b) Incremental by 2 × 1/4 parts of edges. (c) Global by 3 × 1/4 parts of edges.

(d) Incremental by 3 × 1/4 parts of edges. (e) Global by 4 × 1/4 parts of edges. (f) Incremental by 4 × 1/4 parts of edges.

Fig. 9. Complementary visualizations p2p network user clustering comparison of global and incremental matrix factorization for user representations. Note that we first cluster user network then reduce user features dimension from 64 to 3, and the rotation effects are produced by dimension reduction. As the edges increase, the effect of user clustering becomes more intensive and clearly separated.

5.4. Semantic network event detection

Word vectors carrying semantics can be applied in social network big data mining areas. Social streaming process and mining for big data computing are both issues of great importance. The streaming social short text from Open Sina Weibo[34] can be used to detect abnormity and burst of social network events [38–42]. The Open Sina Weibo API produces more than fifty million streaming data every day, containing about 30000 public burst social events.

Experimental data set from Sina Weibo Open Platform can be collected by the API provided by Sina Weibo. First, the raw text is retrieved by using word-split tools like NLPIR[4] or BosonNLP[5] from NLP. Then the co-occurrence of words is counted. By extracting words whose co-occurrence exceeds a certain threshold, the event graph could be constructed. It is an undirected graph whose node represents each single word. The weight value on the edge is the co-occurrence, which can measure the heat of the event. However, this is only the discovery from physical phenomena based on emergence of word and abnormal sub-graph of words. But social network events are the basis of semantic understanding influenced by different group of social users. By using NLP and data mining technology for abnormal graph of words, interpretable event representation [42–44] contains event time, place, participant, emotion, key words, title and related weibos, etc. Finally, both the graph of words and interpretable event representations can be used to represent social event.

Traditional physical phenomena and semantics based methods do not consider the changes in semantics of word over time. However, when combining abnormal sub-graph with incremental word semantics, if the semantics of the words in the graph become closer, nodes representing these words can be merged. In this way, the number of the nodes and the scale of graph used to merge event can be reduced, thus lower the impact by noise.

On this basis, the word vectors are trained respectively by the one-month periodic renewing and online incremental social data. These word vectors are used to measure the semantic distance between words among abnormal graphs of words. Cosine distance of word vectors is adopted to merge nodes that are semantically similar among graphs. Compared the number of events detected by word vectors trained from the one-month periodic renewing and the streaming online incremental social data in 48000 min, the results are shown in Fig. 7. It can be observed that online learning word vector based event detection method fell at the rate of 200 events/100 min on average.[6]

5.5. User multi-label classification

This section presents an experimental analysis of the incremental representations in network user multi-label classification [45]. Network user vectors are extracted from DeepWalk, a novel approach for learning latent representations of vertices in a network [5], in which the Skip-gram model is substituted by matrix factorization model for word representations. The multi-label classification task by user representations in network is evaluated in this section. The training data on the BlogCatalog [46] network is then increased from 10% to 90%. In BlogCatalog network data set, there are 10312 nodes and 333983 edges. The results are presented in Fig. 8, where X-axis refers to the percentage of incremental network data, and Y-axis refers to Micro-F1 score and Macro-F1 score. The performance of incremental and global matrix factorization are similar. Moreover, the speedup of training time conforms to the laws in Fig. 3(b).

5.6. User clustering

This section shows another experimental analysis of the incremental representations in P2P network user complementary visualizations clustering. The Gnutella peer-to-peer network data set [47] contains 6301 nodes and 20777 edges. The network data set is evenly divided into 4 parts by edges. We use one of the 4 parts as base training data, and take another 3 parts as incremental

4 http://open.weibo.com
5 http://bosonnlp.com
6 http://ring.cnbigdata.org/

training data one by one. The network user vectors are extracted from node2vec, an algorithmic framework for learning continuous feature representations for nodes in networks [6], where matrix factorization model for word representations is adopted. Then we cluster feature representations by k-means [48]. For more distinct user feature, each network in 2-D plane with nodes assigned colors based on their clusters is visualized. The performance is presented in Fig. 9. It can be concluded that the comparison between incremental and global matrix factorization for user feature representations cluster are close, and incremental training does a slightly better job than global training in some situations. Besides, the speedup of training time conforms to the laws in Fig. 3(b).

6. Conclusion

This paper proposes an incremental term representation learning method for word and user representations for social network analysis. In order to support social network analysis, we develop a model based on the GloVe, which is one of the most popular unsupervised learning algorithm for term representation. To demonstrate its effectiveness and efficiency, a systematic evaluation of both the training time and the quality of vectors is conducted. Our results are also evaluated in the following social network analysis applications, including word similarity/relatedness, linguistic tasks, semantic event detection, user multi-label classification and user clustering. Experimental results demonstrated that the incremental training significantly outperforms global training in processing speed as well as quality performance on various tasks. The smaller the updated network data is, the faster the update factorizing process can, even up to 30 times faster than in certain cases. In addition, theoretical analysis can also help in better understanding the performance of the incremental model. For future work, we plan to extend our approach to other advanced feature learning models beyond GloVe, such as Neural network language models [49], dependency RNN [50], LSTM and deeper RNN models [30].

Acknowledgments

The corresponding author is Jianxin Li. This work is supported by China 973 Fundamental R&D Program (No. 2014CB340300), NSFC program (No. 61472022, 61421003), SKLSDE-2016ZX-11, and partly by the Beijing Advanced Innovation Center for Big Data and Brain Computing. We thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions.

References

[1] A. Marin, B. Wellman, Social network analysis: an introduction, in: The SAGE Handbook of Social Network Analysis, 2011, pp. 11–25.
[2] J. Scott, Social network analysis, Sage, 2012.
[3] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: EMNLP, vol. 14, 2014, pp. 1532–1543.
[4] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, in: Advances in Neural Information Processing Systems, 2014, pp. 2177–2185.
[5] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 701–710.
[6] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks.
[7] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NIPS, 2013, pp. 3111–3119.
[8] H. Peng, J. Li, Y. Song, Y. Liu, Incrementally Learning the Hierarchical Softmax Function for Neural Language Models, AAAI, 2017.
[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ICLR, URL http://arxiv.org/abs/13013781.
[10] A. Mnih, G.E. Hinton, A scalable hierarchical distributed language model, in: NIPS, 2008, pp. 1081–1088.
[11] H. Peng, Y. Song, D. Roth, Event detection and co-reference with minimal supervision, EMNLP, 2016.
[12] X. Feng, L. Huang, D. Tang, B. Qin, H. Ji, T. Liu, A language-independent neural network for event detection, in: The 54th Annual Meeting of the Association for Computational Linguistics, 2016, p. 66.
[13] L. Hu, C. Shao, J. Li, H. Ji, Incremental learning from news events, Knowl.-Based Syst. 89 (2015) 618–626.
[14] J. Turian, L.-A. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: ACL, 2010, pp. 384–394.
[15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P.P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[16] R. Socher, J. Bauer, C.D. Manning, A.Y. Ng, Parsing with compositional vector grammars, in: ACL (1), 2013, pp. 455–465.
[17] Anonymous, Cross-lingual dataless classification for many languages, in: ACL Accompany Paper As Supplementary Material, 2016.
[18] H. Luo, Z. Liu, H. Luan, M. Sun, Online learning of interpretable word embeddings, in: Proceedings of EMNLP, 2015, pp. 1687–1692.
[19] S. Li, J. Zhu, C. Miao, A generative word embedding model and its low rank positive semidefinite solution, arXiv preprint arXiv:1508.03826.
[20] K. Cao, M. Rei, A joint model for word embedding and word morphology, arXiv preprint arXiv:1606.02601.
[21] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, in: NIPS, 2014, pp. 2177–2185.
[22] W. Chen, D. Grangier, M. Auli, Strategies for training large vocabulary neural language models, arXiv preprint arXiv:1512.04906.
[23] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, TACL 3 (2015) 211–225.
[24] X. Geng, K. Smith-Miles, Incremental learning, Encyclopedia Biometrics (2015) 912–917.
[25] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. C 31 (4) (2001) 497–508.
[26] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[27] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowl.-Based Syst. 27 (2012) 271–280.
[28] Matthew Brand, Incremental singular value decomposition of uncertain data with missing values, in: European Conference on Computer Vision, Springer, Berlin, Heidelberg, 2002, pp. 707–720.
[29] M. Gutmann, A. Hyvärinen, Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics, J. Mach. Learn. Res. 13 (2012) 307–361.
[30] D. Renshaw, K.B. Hall, Long short-term memory language models with additive morphological features for automatic speech recognition, in: ICASSP, 2015, pp. 5246–5250.
[31] C. Hegde, P. Indyk, L. Schmidt, A nearly-linear time framework for graph-structured sparsity, in: ICML, 2015, pp. 928–937.
[32] E.H. Duchi, John Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159.
[33] Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.
[34] M. Faruqui, C. Dyer, Improving vector space word representations using multilingual correlation, in: EACL, 2014, pp. 462–471.
[35] J.L. Myers, A.D. Well, Research Design & Statistical Analysisn, Routledge, 1995.
[36] R. Barac, E. Bialystok, Bilingual effects on cognitive and linguistic development: role of language, cultural background, and education, Child Develop. 83 (2) (2012) 413–422.
[37] C.T. Schütze, The Empirical Base of Linguistics: Grammaticality Judgments and Linguistic Methodology, Language Science Press, 2016.
[38] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 851–860.
[39] H. Sayyadi, M. Hurst, A. Maykov, Event detection and tracking in social streams, in: Icwsm, 2009.
[40] R. Lee, K. Sumiya, Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection, in: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, ACM, 2010, pp. 1–10.
[41] Q. Zhao, P. Mitra, B. Chen, Temporal and information flow based event detection from social text streams, in: AAAI, vol. 7, 2007, pp. 1501–1506.
[42] W. Yu, J. Li, M.Z.A. Bhuiyan, R. Zhang, J. Huai, Ring: real-time emerging anomaly monitoring system over text streams, IEEE Trans. Big Data (2017). http://dx.doi.org/10.1109/TBDATA.2017.2672672.
[43] L. Huang, X.F. Taylor Cassidy, H. Ji, C.R. Voss, J. Han, A. Sil, Liberal event extraction and event schema induction.
[44] D. Yu, H. Ji, Unsupervised person slot filling based on graph mining.
[45] G. Tsoumakas, I. Katakis, Multi-Label Classification: An Overview, Dept. of Informatics, Aristotle University of Thessaloniki, Greece.

[46] L. Tang, H. Liu, Relational learning via latent social dimensions, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 817–826.
[47] M. Ripeanu, I. Foster, A. Iamnitchi, Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design, arXiv preprint cs/0209028.
[48] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[49] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606.
[50] P. Mirowski, A. Vlachos, Dependency recurrent neural language models for sentence completion, in: ACL, 2015, pp. 511–517.

Hao Peng, is currently a Ph.D. candidate at the School of Computer Science and Engineering in Beihang University (BUAA), China. His research interests include deep learning, representation learning, big data computing, social network analysis, distributed computing.

Mengjiao Bao, is currently a master candidate at the School of Computer Science and Engineering in Beihang University (BUAA), China. His research interests include machine learning, big data computing, social network analysis.

Jianxin Li is a professor at the School of Computer Science and Engineering, Beihang University, China. He received his Ph.D. degree from Beihang University in 2008. He was a visiting scholar in machine learning department of Carnegie Mellon University, USA in 2015, and a visiting researcher of MSRA in 2011. His current research interests include data analysis and processing, distributed systems, and system virtualization. The corresponding author.

Md Zakirul Alam Bhuiyan, received the Ph.D. degree and the M. Eng. degree from Central South University, China, in 2009 and 2013 respectively, and the B.Sc. degree from International Islamic University Chittagong, Bangladesh, in 2005, all in Computer Science and Technology. He is currently an assistant professor of the Department of Computer and Information Sciences at the Fordham University. Earlier, he worked as an assistant professor (research) at the Temple University and a post-doctoral fellow at the Central South University, China, a research assistant at the Hong Kong PolyU, and a software engineer in industries. His research focuses on dependable cyber–physical systems, WSN applications, network security, and sensor-cloud computing.

Yaopeng Liu, is currently a master candidate at the School of Computer Science and Engineering in Beihang University (BUAA), China. His research interests include machine learning, cloud computing, distributed systems.

Yu He, is currently a master candidate at the School of Computer Science and Engineering in Beihang University (BUAA), China. His research interests include machine learning, cloud computing, distributed systems.

Erica Yang, is a senior computer scientist with Scientific Computing Department, STFC Rutherford Appleton Laboratory. Her current research interests include high throughput systems, data management, high dimensional visualization, semantics analytics and big data mining.