Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective

Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong and Enhong Chen
University of Science and Technology of China
Hefei, Anhui, China
etali@mail.ustc.edu.cn, linlixu@ustc.edu.cn, {tianfei, jal, xwzhong}@mail.ustc.edu.cn,
cheneh@ustc.edu.cn

Abstract

Recently significant advances have been witnessed in the area of distributed word representations based on neural networks, which are also known as word embeddings. Among the new word embedding models, skip-gram negative sampling (SGNS) in the *word2vec* toolbox has attracted much attention due to its simplicity and effectiveness. However, the principles of SGNS remain not well understood, except for a recent work that explains SGNS as an implicit matrix factorization of the pointwise mutual information (PMI) matrix. In this paper, we provide a new perspective for further understanding SGNS. We point out that SGNS is essentially a representation learning method, which learns to represent the co-occurrence vector for a word. Based on the representation learning view, SGNS is in fact an explicit matrix factorization (EMF) of the words' co-occurrence matrix. Furthermore, extended supervised word embedding can be established based on our proposed representation learning view.

1 Introduction

Distributed word representations, i.e., word embeddings, have been widely applied in various text mining and natural language processing tasks. Different to the traditional one-hot representation that represents a word with a large vector, distributed word representation embeds every word into a low dimensional continuous space. Such representations are assumed to convey semantic and syntactic information of words.

The most widely adopted discipline to learn word embedding vectors is to maximize the corpus likelihood by neural network training [Bengio *et al.*, 2006; Mnih and Hinton, 2007; 2009; Collobert *et al.*, 2011; Huang *et al.*, 2012; Le and Mikolov, 2014; Huang *et al.*, 2013; Pennington *et al.*, 2014; Kiros *et al.*, 2014]. Among all these methods, CBOW and the skip-gram model in the well-known *word2vec* toolbox [Mikolov *et al.*, 2013a; 2013b] have attracted tremendous attention from both academia and industry due to their effectiveness and efficiency. Remarkably, it is shown that the embedding vectors produced by these models preserve the syntactic and semantic relations between words under simple linear operations. For example, relations are captured as *Madrid - Spain* $\approx$ *Paris - France*, *good - best* $\approx$ *great - greatest*.

We focus on the skip-gram negative sampling (SGNS) model in this paper. Despite of its significant success, the underlying theoretical principles of the SGNS model are not clear enough yet. The SGNS model follows the routine of maximizing the conditional probability of the observed contexts given the current word when scanning through the corpus, however it is not clear what information the embedding vectors really convey.

The first attempt to answer this question is proposed in [Omer and Yoav, 2014], where the authors show that SGNS is an implicit matrix factorization (IMF) that factorizes an implicit word-context matrix where the value of each entry indicates the strength of association between the corresponding word-context pair. Specifically, the implicit word-context matrix that SGNS is factorizing is known as the **pointwise mutual information (PMI) matrix** constructed from the raw **co-occurrence matrix**. The authors demonstrate that in the ideal case, i.e., the embedding size of word vectors be infinity, SGNS perfectly reconstructs the PMI matrix.

In this paper, we provide a new perspective for the nature of the SGNS model. To be more specific, our main contribution is that we reformulate the objective of SGNS as a representation learning objective that has never been discovered before. In our representation learning view, the embedding vector learned for a word is a hidden representation for occurrences of the corresponding contexts under a softmax loss. Based on the representation learning view, we further show that SGNS is in fact an explicit matrix factorization (EMF), where the matrix to be factorized is the co-occurrence matrix. From this representation view, the extended task of supervised word embedding will have a clear definition. Compared with the existing IMF analysis of SGNS, our EMF formulation differs in the matrix to be factorized and the reconstruction loss. Furthermore, based on our formulation, the convergence property of the proposed algorithm will be much easier to analyze.

More importantly, once the equivalence of SGNS, representation learning and EMF is established, the new perspective will provide a solid basis for further natural extensions and generalizations of SGNS.

The remainder of the paper is organized as follows. We first review related background including the co-occurrence matrix, PMI, SGNS and general matrix factorization in Sec-

tion 2, then present the perspective of representation learning and EMF for SGNS in Section 3. The optimization algorithm is designed in Section 4, followed with insight from our proposed perspective and the extended approach of supervised explicit matrix factorization. We conduct experimental investigation in Section 6 and demonstrate that the algorithm based on our formulation performs as well as the SGNS in the *word2vec* toolbox.

2 Background

Our explanation of word embedding focuses on skip-gram negative sampling (SGNS) from *word2vec*. In the following we give a brief review of the co-occurrence matrix, PMI, SGNS and general matrix factorization.

2.1 Co-occurrence Matrix and Pointwise Mutual Information

Given a training corpus $\mathcal{D}$ and a word w, several words are selected as the context words for w according to certain strategies, e.g., those neighboring words falling in all the fix-sized windows centered at w. We denote the number of times that a context word c appears in w's contexts as $\#(w,c)$ which is also called the co-occurrence count, and we have the following:

$$\begin{aligned} \#(w) = \sum_{c\in V_C} \#(w,c), \quad \#(c) = \sum_{w\in V_W} \#(w,c) \\ |\mathcal{D}| = \sum_{w\in V_W, c\in V_C} \#(w,c) \end{aligned},$$

where V_W and V_C denote the word and context vocabularies in a text corpus $\mathcal{D}$. The **co-occurrence matrix** is denoted as $\mathbf{D}$, where the entry in the c^{th} row and w^{th} column is $\#(w,c)$. A column of $\mathbf{D}$ can be regarded as an explicit representation for corresponding w denoted as the explicit word vector $\mathbf{d}_w$ in the rest of the paper. Based on $\mathbf{D}$, the PMI matrix $\mathbf{M}$ can be constructed such that $\mathbf{M}_{w,c} = \log(\frac{\#(w,c)|\mathcal{D}|}{\#(w)\#(c)})$.

2.2 Skip-Gram Negative Sampling

For a word $w \in V_W$ and a context word $c \in V_C$, their embedding vectors are represented as column vectors $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{c} \in \mathbb{R}^d$ respectively, where d is the embedding's dimensionality. The embedding vectors of the words in V_W and the context words in V_C constitute the columns of the word and context embedding matrices $\mathbf{W}$ and $\mathbf{C}$.

The general skip-gram model is a simplified statistical language model that aims to predict context words given a central word w. The conditional probability is defined in a softmax form:

$$P(c|w) = \frac{e^{\mathbf{w}^T\mathbf{c}}}{\sum_{c'\in V_C} e^{\mathbf{w}^T\mathbf{c}'}} \tag{1}$$

The context embedding matrix $\mathbf{C}$ and word embedding matrix $\mathbf{W}$ can then be learned through optimizing the following:

$$\max_{\mathbf{W},\mathbf{C}} \sum_{w\in V_W}\sum_{c\in V_C} \#(w,c)\log\, P(c|w) \tag{2}$$

However, it is difficult to calculate the partition function $\sum_{c'\in V_C} e^{\mathbf{w}^T\mathbf{c}'}$ in the denominator of (1). Therefore, it is proposed in [Mikolov *et al.*, 2013b; Mnih and Kavukcuoglu, 2013] to maximize an alternative likelihood exploiting negative sampling

$$\log\, \sigma(\mathbf{w}^T\mathbf{c}) + k\, \mathbb{E}_{c_N\sim P_\mathcal{D}}(\log\, \sigma(-\mathbf{w}^T\mathbf{c}_N)), \tag{3}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function and $P_\mathcal{D}$ is a probability measure taken on all words to sample false context words and k is a hyper parameter specifying the number of such words for each w. This formulation is easy to compute and is named as Skip-Gram Negative Sampling (SGNS). SGNS is applied to all the selected word-context pairs by skip-gram, leading to the following objective:

$$\begin{aligned} &\max_{\mathbf{W},\mathbf{C}} \quad \mathbf{L}(\mathbf{W},\mathbf{C}) \\ &= \sum_{w\in V_W}\sum_{c\in V_C} \#(w,c)\big[\log\, \sigma(\mathbf{w}^T\mathbf{c}) \\ &\qquad + k\mathbb{E}_{c'\sim P_\mathcal{D}}\left(\log\, \sigma(-\mathbf{w}^T\mathbf{c}')\right)\big] \\ &= \sum_{w\in V_W} l(\mathbf{w},\mathbf{C}) \end{aligned} \tag{4}$$

To simplify analysis, following [Omer and Yoav, 2014], $P_\mathcal{D}$ is set to the empirical unigram distribution $P_\mathcal{D}(c) = \frac{\#(c)}{|\mathcal{D}|}$.

2.3 Matrix Factorization and Representation Learning

We revisit matrix factorization and representation learning in this subsection. Given a matrix $\mathbf{X}$, matrix factorization aims to factorize $\mathbf{X}$ into two matrices $\mathbf{Y}$ and $\mathbf{Z}$ such that $\mathbf{X} \approx \mathbf{YZ}$. The objective of matrix factorization can be written as:

$$\min_{\mathbf{Y},\mathbf{Z}} \mathcal{L}(\mathbf{X},\mathbf{YZ}), \tag{5}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function that measures the distance between two matrices with the same size.

Different $\mathcal{L}(\cdot,\cdot)$ leads to different matrix factorization models, including non-negative matrix factorization [Lee and Seung, 2001], probabilistic matrix factorization [Mnih and Salakhutdinov, 2007], max-margin matrix factorization [Srebro *et al.*, 2004]. All of these models can be formulated as special cases of (5).

We can also consider matrix factorization as **Representation Learning**. Under certain circumstance, the objective (5) can be re-written as:

$$\min_{\mathbf{Y},\mathbf{Z}} \Sigma_i \mathcal{L}(\mathbf{x}_i, \mathbf{Yz}_i). \tag{6}$$

where $\mathcal{L}(\cdot,\cdot)$ evaluates the distance between the i^{th} column of $\mathbf{X}$ and $\mathbf{YZ}$, and $\mathcal{L}$ is called the representation loss. From this point of view, one can regard that $\mathbf{z}_i$ is the code and a hidden representation of the observed instance vector $\mathbf{x}_i$ and $\mathbf{Y}$ is regarded as the representation dictionary, therefore the objective of matrix factorization can be viewed as learning a representation for each sample $\mathbf{x}_i$. Sparse coding [Lee *et al.*, 2006] is a well-known representative of representation learning models.

3 SGNS as Representation Learning and Explicit Matrix Factorization

In this section, we provide a novel view for SGNS in the perspective of representation learning and explicit matrix factorization (EMF) that has never been discovered before. To be concrete, we will prove that the objective of SGNS (4) is a special case of (5) and (6), and define the specific loss function $\mathcal{L}(.,.)$. As a consequence, we can view SGNS as a representation learning as well as a matrix factorization model. From the representation learning view, the instance vector to be represented here is the explicit word vector $\mathbf{d}_w$ and the representation dictionary is the context embedding matrix $\mathbf{C}$. From the matrix factorization view, the matrix to be factorized is the words' co-occurrence matrix $\mathbf{D}$.

3.1 Representation Loss for Explicit Word Vector

Before proceeding to our main results, we define the explicit word vector, and the specific representation loss $\mathcal{L}(\cdot,\cdot)$ in (6) for our problem.

Definition 1. *For a word w, its explicit word vector $\mathbf{d}_w \in \mathbb{R}^{|V_C|}$ is defined as: $d_{w,c} = \#(w,c)$, where $d_{w,c}$ is the c^{th} element of vector $\mathbf{d}_w$ and $\#(w,c)$ is the co-occurrence count between word w and c in the corpus.*

We also define the candidate set $\mathcal{S}_w$ of possible explicit word vectors for word w: $\mathcal{S}_w = \mathcal{S}_{w,1} \times \mathcal{S}_{w,2} \times ...\mathcal{S}_{w,c} \times ...\mathcal{S}_{w,|V_c|}$ is the Cartesian product of $|V_C|$ subsets and each subset is defined as $\mathcal{S}_{w,c} = \{0, 1, \cdots, Q_{w,c}\}$, where $Q_{w,c}$ is a pre-defined upper bound for the possible co-occurrence count between word w and c. That is, we assume that all the possible explicit word vectors for word w are elements of the set $\mathcal{S}_w$. The detailed value for $Q_{w,c}$ is specified in the later section and we guarantee that $\#(w,c) \leq Q_{w,c}$, thus $\mathbf{d}_w \in \mathcal{S}_w$.

Given the above definition, we introduce the concept of *Representation Loss for Explicit Word Vector*.

Definition 2. Representation Loss for Explicit Word Vector *is defined as the negative log probability of observing the explicit word vector $\mathbf{d}_w$ given $\mathbf{w}$ and $\mathbf{C}$. To be more concrete,*

$$\begin{aligned}\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w}) &= -\log \frac{e^{\mathbf{d}_w^T\mathbf{C}^T\mathbf{w}}}{\sum_{\mathbf{d}'_w \in \mathcal{S}_w} e^{\mathbf{d}'^T_w\mathbf{C}^T\mathbf{w}}} \\ &\triangleq -\log\ P(\mathbf{d}_w|\mathbf{C}^T\mathbf{w}) \\ &= -\log \prod_{c\in V_C} \frac{e^{d_{w,c}\mathbf{C}_c^T\mathbf{w}}}{\sum_{d'_{w,c}\in\mathcal{S}_{w,c}} e^{d'_{w,c}\mathbf{C}_c^T\mathbf{w}}} \\ &= -\sum_{c\in V_C} \log P(d_{w,c}|\mathbf{C}_c^T\mathbf{w})\end{aligned}, \quad (7)$$

where $\mathcal{L}_S$ serves as the representation loss and $\mathbf{C}_c$ denotes the c^{th} column of $\mathbf{C}$. It should be mentioned that the summation $\sum_{d'_{w,c}\in\mathcal{S}_{w,c}}$ is a summation in Hamming space that will be defined below in the proof of Theorem 1.

The negative representation loss $\mathcal{L}_S$ is inspired from replicated softmax [Hinton and Salakhutdinov, 2009] in which a generative model for representing documents is proposed, and $P(\mathbf{d}_w|\mathbf{C}^T\mathbf{w})$ here is the conditional distribution of the generative model in replicated softmax.

3.2 Equivalence of SGNS and EMF

Theorem 1. *For a word w, when $Q_{w,c}$ is set to $k\frac{\#(w)\#(c)}{|\mathcal{D}|} + \#(w,c)$, $\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w})$ is equivalent to $-l(\mathbf{w},\mathbf{C})$, where $l(\mathbf{w},\mathbf{C})$ is the loss term for w in (4).*

We defer the detailed proof for *Theorem 1* to the end of this subsection. To summarize, *Theorem 1* guarantees that for word w, $\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w})$ is equivalent to $-l(\mathbf{w},\mathbf{C})$ in (4), if the *pseudo context length* $Q_{w,c}$ is set to an appropriate value. Therefore, the objective of SGNS (4) is equivalent to $\min_{\mathbf{w},\mathbf{C}} \sum_{w\in V_W} \mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w})$.

According to general representation learning in Section 2.3, the equivalence shown in *Theorem 1* implies that the embedding vector $\mathbf{w}$ learned in SGNS is a hidden representation vector for each explicit vector $\mathbf{d}_\mathbf{w}$ under the representation loss (7).

As the main corollary derived from *Theorem 1*, we point out that the objective of SGNS (4) is equivalent to explicit matrix factorization (EMF) of matrix $\mathbf{D}$:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{C}} \quad & -\sum_{w\in V_W} l(\mathbf{w},\mathbf{C}) \\ &= \sum_{w\in V_W} \mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w}) \\ &\triangleq \mathbf{MF}(\mathbf{D}, \mathbf{C}^T\mathbf{W}) \\ &= -\operatorname{tr}(\mathbf{D}^T\mathbf{C}^T\mathbf{W}) + \sum_{w\in V_W} \log(\sum_{\mathbf{d}'_w\in\mathcal{S}_w} e^{\mathbf{d}'^T_w\mathbf{C}^T\mathbf{w}}),\end{aligned} \quad (8)$$

where $\mathbf{MF}$ denotes the loss of matrix factorization and corresponds to $\mathcal{L}(\mathbf{X}, \mathbf{YZ})$ in (5). Each column of $\mathbf{C}^T\mathbf{W}$ is supposed to approximate the corresponding $\mathbf{d}_w$ vector under the representation loss (7). In addition, from the last line of (8), we can note that $\mathbf{MF}$ intends to minimize an inner product loss with a soft maximum regularization, and the objective is convex over $\mathbf{C}^T\mathbf{W}$, though it is not jointly convex over $\mathbf{C}$ and $\mathbf{W}$.

Proof of Theorem 1. First the partition function of $P(d_{w,c}|\mathbf{w},\mathbf{C}_c)$ in (7) can be factorized as follow:

$$\begin{aligned}&\sum_{d'_{w,c}\in\mathcal{S}_{w,c}} e^{d'_{w,c}\mathbf{C}_c^T\mathbf{w}} \\ =& \sum_{d'_{w,c,1},d'_{w,c,2}\cdots,d'_{w,c,Q_{w,c}}\in\{0,1\}^{Q_{w,c}}} e^{\sum_{q=1}^{Q_{w,c}} d'_{w,c,q}\mathbf{C}_c^T\mathbf{w}} \\ =& \sum_{d'_{w,c,1},d'_{w,c,2}\cdots,d'_{w,c,Q_{w,c}}\in\{0,1\}^{Q_{w,c}}} \prod_{q=1}^{Q_{w,c}} e^{d'_{w,c,q}\mathbf{C}_c^T\mathbf{w}} \\ =& \prod_{q=1}^{Q_{w,c}} \sum_{d'_{w,c,q}\in\{0,1\}} e^{d'_{w,c,q}\mathbf{C}_c^T\mathbf{w}},\end{aligned}$$

so we reformulate $P(d_{w,c}|\mathbf{w}, \mathbf{C}_c)$ as:

$$\begin{aligned} P(d_{w,c}|\mathbf{C}^T\mathbf{w}) &= \frac{e^{d_{w,c}\mathbf{C}_c^T\mathbf{w}}}{\sum_{d'_{w,c}\in\mathcal{S}_{w,c}} e^{d'_{w,c}\mathbf{C}_c^T\mathbf{w}}} \\ &= \frac{\prod_{q=1}^{Q_{w,c}} e^{d_{w,c,q}\mathbf{C}_c^T\mathbf{w}}}{\prod_{q=1}^{Q_{w,c}}\sum_{d'_{w,c,q}\in\{0,1\}} e^{d'_{w,c,q}\mathbf{C}_c^T\mathbf{w}}} \\ &= \prod_{q=1}^{d_{w,c}} \sigma(\mathbf{C}_c^T\mathbf{w}) \prod_{q=d_{w,c}+1}^{Q_{w,c}} \sigma(-\mathbf{C}_c^T\mathbf{w}) \\ &= \sigma(\mathbf{C}_c^T\mathbf{w})^{d_{w,c}}\sigma(-\mathbf{C}_c^T\mathbf{w})^{Q_{w,c}-d_{w,c}}. \end{aligned} \tag{9}$$

Given the factorization of $P(d_{w,c}|\mathbf{w}, \mathbf{C})$ above, with $\mathbf{C}_c = \mathbf{c}$ and $d_{w,c} = \#(w, c)$, $\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w})$ can be reformulated as:

$$\begin{aligned} &\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w}) \\ =& -\log(P(\mathbf{d}_w|\mathbf{C}^T\mathbf{w})) \\ =& -\sum_{c\in V_C}\left[d_{w,c}\log\,\sigma(\mathbf{w}^T\mathbf{c}) + (Q_{w,c}-d_{w,c})\log\,\sigma(-\mathbf{w}^T\mathbf{c})\right] \\ =& -\sum_{c\in V_C}\big[\#(w,c)\log\,\sigma(\mathbf{w}^T\mathbf{c}) \\ &+ (Q_{w,c}-\#(w,c))\log\,\sigma(-\mathbf{w}^T\mathbf{c})\big] \end{aligned} \tag{10}$$

By substituting $Q_{w,c} = k\frac{\#(w)\#(c)}{|D|} + \#(w, c)$ into (10) above and following the recipe in [Omer and Yoav, 2014], we get

$$\begin{aligned} &\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w}) \\ =& -\sum_{c\in V_C}\left[\#(w,c)\log\sigma(\mathbf{w}^T\mathbf{c}) + k\frac{\#(w)\#(c)}{|\mathcal{D}|}\log\,\sigma(-\mathbf{w}^T\mathbf{c})\right] \\ =& -\sum_{c\in V_C}\#(w,c)\log\,\sigma(\mathbf{w}^T\mathbf{c}) \\ &-\sum_{c_N\in V_C} k\frac{\#(w)\#(c_N)}{|\mathcal{D}|}\log\,\sigma(-\mathbf{w}^T\mathbf{c}_N) \\ =& -\sum_{c\in V_C}\#(w,c)\log\sigma(\mathbf{w}^T\mathbf{c}) - \#(w)k\mathbb{E}_{c_N\sim P_\mathcal{D}}\log\sigma(-\mathbf{w}^T\mathbf{c}_N) \\ =& -\sum_{c\in V_C}\#(w,c)\left[\log\sigma(\mathbf{w}^T\mathbf{c}) + k\mathbb{E}_{c_N\sim P_\mathcal{D}}\left(\log\sigma(-\mathbf{w}^T\mathbf{c}_N)\right)\right] \\ =& -l(\mathbf{w}, \mathbf{C}) \end{aligned} \tag{11}$$

Therefore, given appropriate $Q_{w,c}$, the negative representation loss $\mathcal{L}_S$ (7) is equivalent to $-l(\mathbf{w}, \mathbf{C})$ in (4). ∎

4 Optimization Algorithm

Given the equivalence of SGNS, representation learning and EMF established above, we propose a new optimization algorithm to train word embeddings based on the matrix factorization formulation. Specifically, we leverage the alternating minimization scheme for optimization, which is an effective and widely adopted method in the matrix factorization literature. Before we formally describe the algorithm, we derive the gradients of (8) that will be used in the algorithm:

$$\begin{aligned} \frac{\partial\mathbf{MF}(\mathbf{D}, \mathbf{C}^T\mathbf{W})}{\partial\mathbf{C}} &= \sum_{w\in V_w} -\frac{\partial\mathcal{L}_S(\mathbf{d}_w, \mathbf{C}^T\mathbf{w})}{\partial\mathbf{C}} \\ &= \sum_{w\in V_w} -\mathbf{d}_w\mathbf{w}^T + \mathbb{E}_{\mathbf{d}'_w|\mathbf{C}^T\mathbf{w}}\left[\mathbf{d}'_w\right]\mathbf{w}^T \\ &= \left(\mathbb{E}_{\mathbf{D}'|\mathbf{C}^T\mathbf{W}}\mathbf{D}' - \mathbf{D}\right)\mathbf{W}^T \\ \frac{\partial\mathbf{MF}(\mathbf{D}, \mathbf{C}^T\mathbf{W})}{\partial\mathbf{W}} &= \mathbf{C}\left(\mathbb{E}_{\mathbf{D}'|\mathbf{C}^T\mathbf{W}}\mathbf{D}' - \mathbf{D}\right), \end{aligned} \tag{12}$$

where $\mathbb{E}_{\mathbf{d}'_w|\mathbf{w},\mathbf{C}}$ denotes the conditional expectation taken on $\mathbf{d}'_w$ with respect to the distribution $P(\mathbf{d}_w|\mathbf{C}^T\mathbf{w})$ in (7) and $\mathbb{E}_{\mathbf{D}'|\mathbf{C}^T\mathbf{W}}$ denotes the concatenation of all $\mathbb{E}_{\mathbf{d}'_w|\mathbf{C}^T\mathbf{w}}$ from $w \in V_W$. The expectation $\mathbb{E}_{\mathbf{d}'_w|\mathbf{C}^T\mathbf{w}}\left[\mathbf{d}'_w\right]$ can be computed in a closed form. According to (7) and (9), we have:

$$\mathbb{E}_{d'_{w,c}|\mathbf{C}_c^T\mathbf{w}}\left[d'_{w,c}\right] = Q_{w,c}\sigma(\mathbf{C}_c^T\mathbf{w}), \tag{13}$$

where $\mathbb{E}_{d'_{w,c}|\mathbf{C}_c^T\mathbf{w}}\left[d'_{w,c}\right]$ is the c^{th} entry in the vector $\mathbb{E}_{\mathbf{d}'_w|\mathbf{C}^T\mathbf{w}}\left[\mathbf{d}'_w\right]$ and is the expectation of a binomial distribution given the probability $\sigma(\mathbf{C}_c\mathbf{w})$ and number of trials $Q_{w,c}$.

The details of the Alternating Minimization for Explicit Matrix Factorization (AMEMF) algorithm are summarized in *Algorithm 1*. In the main loop, there are two minimization steps (starting from line 4 and line 9 respectively), which minimize the objective with regard to W and C alternatively. For each minimization step, we employ the gradient descent scheme based on the gradients in (12).

Algorithm 1: Alternating minimization for explicit matrix factorization

Input: Co-occurrence matrix $\mathbf{D}$, step-size of gradient descent η, maximum number of iterations K
Output: $\mathbf{C}_K$, $\mathbf{W}_K$

1 initialize $\mathbf{C}_i$ and $\mathbf{W}_i$ randomly, i = 1;
2 **while** $i \leq K$ **do**
3 $\mathbf{W}_i = \mathbf{W}_{i-1}$;
4 *//minimize over* $\mathbf{W}$;
5 **repeat**
6 $\mathbf{W}_i = \mathbf{W}_i - \eta\,\mathbf{C}_{i-1}\left(\mathbb{E}_{\mathbf{D}'|\mathbf{W}_i,\mathbf{C}_{i-1}}\mathbf{D}' - \mathbf{D}\right)$;
7 **until** *Convergence*;
8 $\mathbf{C}_i = \mathbf{C}_{i-1}$;
9 *//minimize over* $\mathbf{C}$;
10 **repeat**
11 $\mathbf{C}_i = \mathbf{C}_i - \eta\left(\mathbb{E}_{\mathbf{D}'|\mathbf{W}_i,\mathbf{C}}\mathbf{D}' - \mathbf{D}\right)\mathbf{W}_i^T$;
12 **until** *Convergence*;
13 $i = i + 1$;

The main difference between AMEMF and the algorithm for SGNS in *word2vec* is that AMEMF is a batch alternating minimization algorithm, while *word2vec* scans through the corpus and updates $\mathbf{w}$ and $\mathbf{c}$ in a stochastic mode. Subtle differences in implementation include whether a negative sampling procedure is used (SGNS) or not (AMEMF); and the

way to tune the learning rates: whether it is a linearly dropping learning rate to guarantee the convergence of stochastic gradient descent (SGD) in SGNS or a constant learning rate in AMEMF.

Despite of the differences, in the experiments in Section 6, we will demonstrate that the optimization algorithm AMEMF derived directly from the matrix factorization formulation (8) performs comparably with the SGD algorithm adopted in *word2vec*, which further verifies the equivalence between SGNS and EMF.

4.1 Convergence Analysis

Here we analyze the convergence property of the proposed algorithm AMEMF. Firstly, note that the objective in each minimization subprocedure (line 4 and 9 in *Algorithm 1*) is convex, which guarantees the optimal solution of each subprocedure can be reached with sublinear convergence rate [Nesterov, 2004] when the step-size is chosen properly. The whole minimization procedure can be summarized as an iteration over the following two steps: $\mathbf{W}_i = \arg\min_{\mathbf{W}} \mathbf{MF}(\mathbf{D}, \mathbf{C}_{i-1}^T \mathbf{W})$ and $\mathbf{C}_i = \arg\min_{\mathbf{C}} \mathbf{MF}(\mathbf{D}, \mathbf{C}^T \mathbf{W}_i)$, which implies $\mathbf{MF}(\mathbf{D}, \mathbf{C}_{i-1}^T \mathbf{W}_{i-1}) \geq \mathbf{MF}(\mathbf{D}, \mathbf{C}_{i-1}^T \mathbf{W}_i) \geq \mathbf{MF}(\mathbf{D}, \mathbf{C}_i^T \mathbf{W}_i)$. As a consequence, it is guaranteed that the objective (8) descents monotonically and the algorithm will converge due to the lower bounded objective.

5 Insight from the Equivalence

We have presented the main results of this paper by establishing the equivalence between SGNS and representation learning as well as matrix factorization. Meanwhile, the contribution of this work is beyond the equivalence itself in the sense that we can now derive natural extensions of word embedding by leveraging the equivalence. To be more concrete, we propose an extended task *Supervised Word Embedding*, which is of great practical importance and can be conducted much more naturally within the framework of representation learning and matrix factorization than the skip-gram model. This enhances the significance of the equivalence which not only helps understanding the skip-gram model, but also benefits real world tasks.

5.1 Supervised Word Embedding

Before describing the *Supervised Word Embedding* task, we first review the **analogical reasoning task** proposed in [Le and Mikolov, 2014], which serves as the basis for our task. The analogical reasoning task aims to find the most proper answer *d* in a query *a* - *b* ≈ *c* - *d* where *d* is not known, such as *Madrid* - *Spain* ≈ *Paris* - *France* and *good* - *best* ≈ *great* - *greatest*. Given the corresponding embedding vectors $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$, we find the most proper word d through finding the most similar vector for $\mathbf{w}_b - \mathbf{w}_a + \mathbf{w}_c$ under cosine distance. The percentage of these answers that we predict accurately in queries is taken as the evaluation metric for this task.

In practice, the queries in the analogical reasoning task can not only act as the benchmark for evaluating word embedding, but also provide additional side information to guide the embedding process and boost the performance. We refer this task of leveraging the information from the analogical reasoning queries to improve word embedding as *Supervised Word Embedding*. Although our proposed EMF is an unsupervised model as SGNS, the formulation (8) can be generalized to a supervised model naturally to incorporate supervised information based on the representation learning view, given the training corpus and analogical reasoning queries. Specifically, a query in the analogical reasoning task has four components that are denoted as "a", "b", "c" and the accurate answer "d", and the corresponding co-occurrence vectors and embedding vectors are denoted as $\mathbf{d}_a$, $\mathbf{d}_b$, $\mathbf{d}_c$, $\mathbf{d}_d$ and $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$, $\mathbf{w}_d$ respectively. The objective of supervised explicit matrix factorization (SEMF) can then be written as:

$$\min_{\mathbf{W},\mathbf{C}} \mathbf{MF}(\mathbf{D}, \mathbf{C}^T\mathbf{W}) + \lambda\, \mathbf{MF}(\mathbf{D}_d, \mathbf{C}^T(\mathbf{W}_b - \mathbf{W}_a + \mathbf{W}_c)), \tag{14}$$

where matrices $\mathbf{D}_d, \mathbf{W}_a, \mathbf{W}_b, \mathbf{W}_c$ consist of $\mathbf{d}_d, \mathbf{w}_a, \mathbf{w}_b, \mathbf{w}_c$ as columns respectively, and provide supervised information. λ is a hyper-parameter controlling the degree of supervision. In this model, $\mathbf{w}_d$ is a representation for $\mathbf{d}_d$ due to the first term and $\mathbf{w}_b - \mathbf{w}_a + \mathbf{w}_c$ also corresponds to a representation for $\mathbf{d}_d$ due to the second term , and thus $\mathbf{w}_d$ is supposed to approximate $\mathbf{w}_b - \mathbf{w}_a + \mathbf{w}_c$. In this way, side information can be leveraged to guide the embedding process. we can observe that the proposed supervised model is solely based on our $\mathbf{MF}$ matrix factorization loss and is a natural extension of EMF.

The supervised model (14) can be solved similarly with the alternating minimization framework as in *Algorithm 1*.

6 Experiments

In this section, we conduct several experiments to verify the effectiveness of the AMEMF algorithm in the EMF framework. The experiments consist of two parts: comparison of word embedding methods and evaluation of supervised word embedding. They are both evaluated by the analogical reasoning task. To compare the methods, we will evaluate the performance of SGNS, SPPMI, IMF and EMF, and verify the equivalence of SGNS and EMF. Here SPPMI indicates the Shifted Positive Pointwise Mutual Information (PMI) matrix $\mathbf{M}^{SPPMI}$, which is a variant of PMI as $\mathbf{M}_{w,c}^{SPPMI} = \max(\mathbf{M}_{w,c} - \log k, 0)$ and has better performance than the original PMI in the analogical reasoning task according to [Omer and Yoav, 2014]. IMF here represents the word embedding approach based on Singular Value Decomposition (SVD) of the SPPMI matrix. In the supervised word embedding task, we will compare EMF and SEMF, and justify the superiority of SEMF by leveraging side information.

Datasets

We use a publicly accessible dataset Enwik9[1] as our training corpus. Enwik9 contains about 124 million tokens. We adopt the original dataset of analogical reasoning queries used in [Mikolov *et al.*, 2013a] for the analogical reasoning task, which contains 19544 queries called *Google query*

[1] http://mattmahoney.net/dc/textdata.html

k	Method	mini-count			
		3000	4000	5000	6000
1	SPPMI	66.43%	66.43%	60.86%	61.94%
2	SGNS	73.73%	75.66%	70.43%	69.78%
	IMF	54.05%	57.38%	55.14%	55.22%
	EMF	**75.18**%	**76.02**%	**70.57**%	**71.08**%
4	SGNS	74.02%	75.02%	**71.71**%	**71.64**%
	IMF	37.55%	40.91%	34.43%	38.06%
	EMF	**74.82**%	**75.57**%	68.57%	**71.64**%
6	SGNS	74.82%	**77.38**%	**72.86**%	**72.39**%
	IMF	31.69%	30.41%	31.86%	31.72%
	EMF	**75.04**%	75.57%	71.14%	71.64%

Table 1: Comparison of SGNS, PMI, IMF and EMF in the analogical reasoning task

Method	λ	Training/Test set ratio			
		10/90%	30/70%	50/50%	70/30%
EMF	/	62.97%	62.97%	62.12%	62.21%
SEMF	0.025	64.92%	70.50%	74.69%	79.55%
	0.05	67.52%	76.56%	81.66%	**86.78**%
	0.075	69.06%	79.75%	**84.70**%	82.59%
	0.1	70.82%	**80.53**%	76.12%	77.46%
	0.125	**71.95**%	79.50%	72.04%	57.59%

Table 2: Comparison of EMF and SEMF in terms of accuracy with different training ratios and λ values

dataset. Vocabulary size is controlled by a hyper-parameter $\mathrm{mini\text{-}count}$ that filters low frequency (less than $\mathrm{mini\text{-}count}$) words out and different vocabulary sizes result in different sizes of query dataset correspondingly. The negative sampling parameter is denoted as k as in (4).

Experimental Setup

The dimensionality of all embedding vectors is set to 200. For comparison of word embedding methods, we compare the embedding vectors produced by SGNS, SPPMI, IMF with SVD and EMF with the AMEMF algorithm under different k and $\mathrm{mini\text{-}count}$ values. The step-size of AMEMF is set to $6e-7$. To keep the settings of AMEMF and SGNS as consistent as possible, they use the same co-occurrence matrix produced by the skip-gram strategy with window size 5. The step-size of SGNS is set according to the default in the *word2vec* toolbox, $P_{\mathcal{D}}$ is set to the unigram distribution, and the number of iterations K in AMEMF is set to 200.

For the *supervised word embedding* task, we partition the Google query dataset into a training query set and a test query set, and then train the supervised model (SEMF) on the training query set and test its embedding vectors in the analogical reasoning task. We compare SEMF and the unsupervised model (EMF) with different K values and training ratios. The $\mathrm{mini\text{-}count}$ value is fixed to 1000, the step-size is set to $5e{-}7$, and the number of iterations K in AMEMF is set to 100.

6.1 Experimental Results and Analysis

We evaluate SGNS, SPPMI, IMF and EMF in the analogical reasoning task. The comparison of these four models in terms of the analogical reasoning accuracy is shown in *Table 1* with different values of the negative sampling parameter k and $\mathrm{mini\text{-}count}$. For each cell corresponding to a combination of k and $\mathrm{mini\text{-}count}$ values, we highlight the best performance. The SPPMI model achieves its best performance when k is set to 1 and the its accuracy decreases rapidly when k increases, therefore, we evaluate it independently. With $k = 1$, one can observe that the accuracy of SPPMI is not comparable to the best performance in *Table 1*.

In the rest 12 cells of *Table 1*, EMF performs best in 8 cells, while SGNS performs best in 5 cells. Though it appears that EMF is subtly better than SGNS, the accuracy of EMF and SGNS are very close actually. Generally speaking, EMF and SGNS perform quite similarly for all the different k and $\mathrm{mini\text{-}count}$ parameter pairs and are significantly superior to the other models. This experimental result empirically serves as a strong evidence of the equivalence between EMF and SGNS. Meanwhile, the IMF model does not perform well in the analogical reasoning task which matches the experimental result from [Omer and Yoav, 2014], and this also demonstrates that EMF could be a more reasonable connection between SGNS and matrix factorization than IMF.

Next the unsupervised model EMF and supervised model SEMF are evaluated with analogical reasoning test accuracy in *Table 2*. There is no natural way to incorporate supervised information for the IMF model, which is another advantage of EMF over IMF. *Table 2* shows that SEMF significantly outperforms EMF, verifying that incorporating supervised information improves the quality of word embeddings. Given the similar performance of EMF and SGNS in the analogical reasoning task, the superiority of SEMF over the original *word2vec* is obvious. As to the effects of different hyper parameters, one can observe that the performance of the unsupervised EMF remains stable with different training ratios. In the meantime, it can be observed that the performance of SEMF grows with the training ratio and λ impacts the degree of supervision significantly. This experiment not only verifies the generalization ability of SEMF, but also inspires that we can equip embedding vectors with intended properties through supervision.

7 Conclusion

We revisit the skip-gram negative sampling (SGNS) model in the popular toolbox *word2vec*, and prove that intrinsically SGNS is a representation learning method, as well as an explicit matrix factorization (EMF) of the co-occurrence matrix that is directly obtained from corpus. Different to implicit matrix factorization (IMF) [Omer and Yoav, 2014], our objective is explicitly equivalent to SGNS, based on which the task of supervised word embedding can be conducted naturally. Experimental results justify the equivalence between SGNS and EMF, as well as the validity of supervised word embedding.

Acknowledgments

We would like to thank Prof. Qing Ling for his advice on algorithm design. This work is supported by the National Natural Science Foundation of China (No. 61375060), and the National Science Foundation for Distinguished Young Scholars of China (No. 61325010).

References

[Bengio *et al.*, 2006] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[Hinton and Salakhutdinov, 2009] Geoffrey E Hinton and Ruslan Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[Huang *et al.*, 2012] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[Huang *et al.*, 2013] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2333–2338, 2013.

[Kiros *et al.*, 2014] Ryan Kiros, Richard S. Zemel, and Ruslan R. Salakhutdinov. A multiplicative model for learning distributed text-based attribute representations. In *Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada*, pages 2348–2356, 2014.

[Le and Mikolov, 2014] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China*, pages 1188–1196, 2014.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[Lee *et al.*, 2006] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[Mnih and Hinton, 2007] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM, 2007.

[Mnih and Hinton, 2009] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

[Mnih and Kavukcuoglu, 2013] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273, 2013.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[Nesterov, 2004] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

[Omer and Yoav, 2014] Levy Omer and Goldberg Yoav. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, 2014.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12, 2014.

[Srebro *et al.*, 2004] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.