Report on: Neural Word Embedding as Implicit Matrix Factorization

Shashank Rangarajan
sr87317@usc.edu
Role: TEACHER 1

1 Summary

The paper (Levy and Goldberg, 2014) analyzes the skip-gram with negative-sampling (SGNS) word embedding method and shows that it implicitly factorizes a word-context matrix based on pointwise mutual information (PMI) of word-context pairs shifted by a constant. The paper also examines another embedding method called noise-contrastive estimation (NCE), which is implicitly factorizing a similar matrix based on log conditional probability of word given its context. The authors show that representing words with a sparse Shifted Positive PMI word-context matrix improves results on word similarity tasks and one of two analogy tasks. The paper compares SGNS and exact factorization with SVD for word similarity tasks and finds that SVD achieves similar results. However, SGNS remains superior to SVD for analogy tasks, possibly due to the weighted nature of SGNS's factorization.

This document is structured as follows: Section 2 presents significant insights from my viewpoint. Lastly, Section 3 outlines industrial applications of SGNS and their implications.

2 Key Insights from the Paper

The three main takeaways from the paper are - the implicit factorization in SGNS, the comparision of direct factorization to SGNS, and the effect of negative sampling on direct factorization v/s SGNS.

2.1 SGNS as Implicit Matrix Factorization

The main idea presented in the paper is that SGNS can be thought of as an instance of matrix factorization. In SGNS, we embed words and contexts into a low-dimensional space R^d, which gives us matrices $W \in R^{(|V_W| \times d)}$ and $C \in R^{(|V_c| \times d)}$ where the rows correspond to the embedding of each word and context respectively. We can think of W and C as a factorization of a matrix $M = W \cdot C^T$, $M \in R^{(|V_W| \times |V_C|)}$, where every cell of the matrix M corresponds to some association between the words w and contexts c, which we can denote as $f(w, c)$.

Solving for $f(w, c)$ by assuming a perfect reconstruction of M (as derived in the paper), we see that the association can be expressed as:

$$f(w, c) = \log\left(\frac{\#(w, c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

which is equivalent to the PMI shifted by a constant.

$$M_{i,j} = f(w_i, c_i) = \mathit{PMI}(w_i, c_i) - \log k$$

where k denotes negative sampling value.

Furthermore, when we relax the assumption of perfect reconstruction, the optimization becomes a weighted matrix factorization where the objective is to seek the optimal d-dimensional factorization of the matrix $M - \log k$ under a metric which pays more for deviations on frequent (w, c) pairs than deviations on infrequent ones.

2.2 Comparision of SGNS and Singular Value Decomposition (SVD)

In the paper, Spectral Dimensionality Reduction (SDR) is introduced as a method for word embedding based on the SVD factorization of a shifted PPMI matrix. Table 1 shows the pros and cons of each approach.

2.3 Effect of negative sampling value k over SVD and SGNS

In the paper, the empirical results (Figure 1) show the percentage of deviation from the optimal objective value for various algorithms and values of $k \in \{1, 5, 15\}$.

An important observation: SVD becomes very erroneous as k increases. This is the result of increasing sparsity in the matrix. As sparsity increases in the matrix, SVD can become inaccurate and prefer factorizations close to the zero matrix.

SGNS performs better at higher values of k by giving more weight to frequent pairs during training, while SVD treats all matrix cells equally.

Other observations: a) Shifted PPMI is indeed a near-perfect approximation of the optimal solution, even though it discards a lot of information considering only positive cells; b) SVD is slightly better than SGNS at optimizing the objective for $d \leq 500$ and $k = 1$. However, while SGNS is able to leverage higher dimensions and reduce its error significantly, SVD fails to do so.

3 Application: Recommender Systems

The winners of the Netflix Prize (Prize, 2006) employed a matrix factorization approach to tackle the collaborative filtering problem by breaking down the user-item matrix into low-rank matrices that capture user and item preferences. Recently, SGNS has emerged as a potential solution for recommender systems, such as in (Ozsoy, 2016), where the aim is to find embeddings for users and items based on co-occurrence of items in the user's rating or activity history.

An important problem that SGNS can address is obtaining embeddings for different types of content or items. For example, in a music streaming application, the SGNS objective can be employed to embed songs into vectors based on user metrics such as how often songs are played in sequence. The key advantage of using SGNS in this context is that it helps solve the cold-start problem. When a new song is added to the service, we can use averages or other aggregations of songs in the same genre as the initial vectors for the new song. Similarly, we can use SGNS to get embeddings of users and group users into similar clusters, which is useful in providing recommendations to new users using similar aggregations.

Another formulation involves using SGNS to sample the top-k recommendations from a user-item PMI. For example, we can create a skip-gram based model that employs users as input vectors (words) to predict the most liked movies (context). Instead of using the embeddings, we can use the distribution to sample the top-k movies to be recommended to the user. This formulation, however, presents two challenges. Firstly, due to the weighted nature of the inherent objective function in SGNS, the user may only be recommended movies that are very similar to each other, which fails to account for diversity in the recommendations. Secondly, this formulation limits scalability to new users as the model needs to be retrained every time a new user is added.

To overcome the second challenge, an alternative formulation can be employed, where the user-item interaction itself is embedded to find similar interactions. This is more feasible using autoencoder models, which also have implicit matrix factorizations built into them.

4 Tables

SGNS	
Pros	- SGNS is weighted, and hence prefers correct values for frequent pair (w, c) and allows more error for non-frequent ones - SGNS distinguishes between observed and unobserved events - SGNS does not require the underlying matrix to be sparse one, and this enables optimization of dense matrix like shifted PMI matrix
Cons	- SGNS is a gradient based method hence the solution is not exact - SGNS training procedures are not feasible for large corpora as it requires each observation of (w, c) to be presented separately.
SVD	
Pros	- Exact solution - hence SVD does not require hyperparameter tuning - Can be easily trained on count-aggregated data, making it scalable to large corpora
Cons	- Sparsity is bad, SVD suffers from unobserved values, which are very common, and result in zero-matrix factorizations - SVD is unweighted, and solving for exact weighted SVD is a computationally hard problem - SVD does not optimize on shifted PMI directly as it is infeasible

Table 1: Pros and Cons of SGNS, SVD

5 Figures

Method	PMI$-\log k$	SPPMI	SVD			SGNS		
			$d = 100$	$d = 500$	$d = 1000$	$d = 100$	$d = 500$	$d = 1000$
$k = 1$	0%	0.00009%	26.1%	25.2%	24.2%	31.4%	29.4%	7.40%
$k = 5$	0%	0.00004%	95.8%	95.1%	94.9%	39.3%	36.0%	7.13%
$k = 15$	0%	0.00002%	266%	266%	265%	7.80%	6.37%	5.97%

Figure 1: Percentage of deviation from the optimal objective value (lower values are better).

6 Key points from the discussion

- The paper serves as a valuable reference point for assessing the interpretability of deep learning models via mathematical deduction.
- This paper likely inspired exploration of biases in optimizers like SGD and Gradient descent, and investigate why they favor certain solutions.

References

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Makbule Gulcin Ozsoy. 2016. From word embeddings to item recommendation. *CoRR*, abs/1601.01356.

Netflix Prize. 2006. Netflix prize.