Low-Rank Approximation of Matrices for PMI-based Word Embeddings

Alena Sorokina, Aidana Karipbayeva, and Zhenisbek Assylbekov

Department of Mathematics, Nazarbayev University
{alena.sorokina, aidana.karipbayeva, zhassylbekov}@nu.edu.kz

Abstract. We perform an empirical evaluation of several methods of low-rank approximation in the problem of obtaining PMI-based word embeddings. All word vectors were trained on parts of a large corpus extracted from English Wikipedia (enwik9) which was divided into two equal-sized datasets, from which PMI matrices were obtained. A repeated measures design was used in assigning a method of low-rank approximation (SVD, NMF, QR) and a dimensionality of the vectors (250, 500) to each of the PMI matrix replicates. Our experiments show that word vectors obtained from the truncated SVD achieve the best performance on two downstream tasks, similarity and analogy, compare to the other two low-rank approximation methods.

Keywords: natural language processing · pointwise mutual information · matrix factorization · low-rank approximation · word vectors

1 Introduction

Today word embeddings play an important role in many natural language processing tasks, from predictive language models and machine translation to image annotation and question answering, where they are usually 'plugged in' to a larger model. An understanding of their properties is of interest as it may allow the development of better performing embeddings and improved interpretability of models using them. One of the widely-used word embedding models is the Skip-gram with negative sampling (SGNS) of Mikolov et al. (2013). Levy and Goldberg (2014) showed that the SGNS is implicitly factorizing a pointwise mutual information (PMI) matrix shifted by a global constant. They also showed that a low-rank approximation of the PMI matrix by truncated singular-value decomposition (SVD) can produce word vectors that are comparable to those of SGNS. However, truncated SVD is not the only way of finding a low-rank approximation of a matrix. It is optimal in the sense that it minimizes the approximation error in the Frobenius and the 2- norms, but this does not mean that it produces optimal word embeddings, which are usually evaluated in downstream NLP tasks. The question is: Is there any other method of low-rank matrix approximation that produces word embeddings better than the truncated SVD factorization? Our experiments show that the truncated SVD is actually a strong baseline which we failed to beat by another two widely-used low-rank approximation methods.

2 Low-Rank Approximations of the PMI-matrix

The simplest version of a PMI matrix is a symmetric matrix with each row and column indexed by words[1], and with elements defined as

$$\mathrm{PMI}(i,j)=\log\frac{p(i,j)}{p(i)p(j)}, \tag{1}$$

where $p(i,j)$ is the probability that the words i, j appear within a window of a certain size in a large corpus, and $p(i)$ is the unigram probability for the word i. For computational purposes, Levy and Goldberg (2014) suggest using a positive PMI (PPMI), defined as

$$\mathrm{PPMI}(i,j)=\max(\mathrm{PMI}(i,j),0). \tag{2}$$

They also show empirically that the low-rank SVD of the PPMI produces word vectors which are comparable in quality to those of the SGNS.

The low-rank matrix approximation is approximating a matrix by one whose rank is less than that of the original matrix. The goal of this is to obtain a more compact representation of the data with a limited loss of information. In what follows we give a brief overview of the low-rank approximation methods used in our work. Since both PMI (1) and PPMI (2) are square matrices, we will consider approximation of square matrices. For a thorough and up-to-date review of low-rank approximation methods see the paper by Kishore Kumar and Schneider (2017).

Singular Value Decomposition (SVD) factorizes $\mathbf{A}\in\mathbb{R}^{n\times n}$, into the matrices $\mathbf{U}\in\mathbb{R}^{n\times n}$, $\mathbf{S}\in\mathbb{R}^{n\times n}$ and $\mathbf{V}^\top\in\mathbb{R}^{n\times n}$:

$$\mathbf{A}=\mathbf{U}\mathbf{S}\mathbf{V}^\top,$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{S}$ is a rectangular diagonal matrix whose entries are in descending order, $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_n\geq 0$, along the main diagonal, and are known as the singular values of $\mathbf{A}$. The rank d approximation (also called *truncated* or *partial SVD*) of $\mathbf{A}$, $\mathbf{A}_d$ where $d<\operatorname{rank}\mathbf{A}$, is given by zeroing out the $n-d$ trailing singular values of $\mathbf{A}$, that is[2]

$$\mathbf{A}_d=\mathbf{U}_{1:n,1:d}\mathbf{S}_{1:d,1:d}\mathbf{V}^\top_{1:d,1:n}.$$

By the Eckart-Young theorem (Eckart and Young, 1936), A_d is the closest rank-d matrix to A in Frobenius norm, i.e. $\|\mathbf{A}-\mathbf{A}_d\|_F\leq\|\mathbf{A}-\mathbf{B}\|_F$, $\forall\mathbf{B}\in\mathbb{R}^{n\times n}:\operatorname{rank}(\mathbf{B})=d$. Levy and Goldberg (2014) suggest factorizing the PPMI matrix with truncated SVD, and then taking the rows of $\mathbf{U}_{1:n,1:d}\mathbf{S}^{1/2}_{1:d,1:d}$ as word vectors, and we follow their approach.

[1] Assume that words have already been converted into integer indices.

[2] $\mathbf{A}_{a:b,c:d}$ is a submatrix located at the intersection of rows $a,a+1,\dots,b$ and columns $c,c+1,\dots,d$ of a matrix $\mathbf{A}$.

QR decomposition with column pivoting of $\mathbf{A} \in \mathbb{R}^{n \times n}$ has the form $\mathbf{AP} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is orthogonal, $\mathbf{R} \in \mathbb{R}^{n \times n}$ is upper triangular and $\mathbf{P} \in \mathbb{R}^{n \times n}$ is a permutation matrix. The rank d approximation to $\mathbf{A}$ is then

$$\mathbf{A}_d = \mathbf{Q}_{1:n,1:d}[\mathbf{RP}^\top]_{1:d,1:n}$$

which is called *truncated QR decomposition* of $\mathbf{A}$. After factorizing the PPMI matrix with this method we suggest taking the rows of $\mathbf{Q}_{1:n,1:d}$ as word vectors.

However, we suspect that a valuable information could be left in the $\mathbf{R}$ matrix. A promising alternative to SVD is a Rank Reveling QR decomposition (RRQR). Assume the QR factorization of the matrix $\mathbf{A}$:

$$\mathbf{AP} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix}$$

where $\mathbf{R}_{11} \in \mathbb{R}^{d \times d}$, $\mathbf{R}_{12} \in \mathbb{R}^{d \times (n-d)}$, $\mathbf{R}_{22} \in \mathbb{R}^{(n-d) \times (n-d)}$. For RRQR factorization, the following condition should be satisfied:

$$\sigma_{\min}(\mathbf{R}_{11}) = \Theta(\sigma_d(\mathbf{A}))$$
$$\sigma_{\max}(\mathbf{R}_{22}) = \Theta(\sigma_{d+1}(\mathbf{A}))$$

which suggests that the most significant entries are in $\mathbf{R}_{11}$, and the least important are in $\mathbf{R}_{22}$. Thus, we also suggest taking the columns of $[\mathbf{RP}^\top]_{1:d,1:n}$ as word vectors.

Non negative matrix factorization (NMF). Given a non negative matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a positive integer $d < n$, NMF finds non negative matrices $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{H} \in \mathbb{R}^{d \times n}$ which minimize (locally) the functional $f(\mathbf{W}, \mathbf{H}) = \|\mathbf{A} - \mathbf{WH}\|_F^2$. The rank d approximation of $\mathbf{A}$ is simply

$$\mathbf{A}_d = \mathbf{WH}.$$

When factorizing the PPMI matrix with NMF, we suggest taking the rows of $\mathbf{W}$ as word vectors.

3 Experimental Setup

3.1 Corpus

All word vectors were trained on the `enwik9` dataset[3] which was divided into two equal-sized splits. The PMI matrices on these splits were obtained using the `hypewords` tool of Levy et al. (2015). All corpora were pre-processed by removing non-textual elements, sentence splitting, and tokenization. PMI matrices were derived using a window of two tokens to each side of the focus word, ignoring words that appeared less than 300 times in the corpus, resulting in vocabulary sizes of roughly 13000 for both words and contexts. A repeated measures design was used for assigning the method of factorization (SVD, QR, NMF) and dimensionality of the vectors (250, 500) to each PMI matrix replicate. We used two replicates per each level combination.

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

3.2 Training

Low-rank approximations were performed using the following open-source implementations:

- Sparse SVD from SciPy (Jones et al., 2014),
- Sparse RRQR from SuiteSparse (Davis and Hu, 2011), and
- NMF from scikit-learn (Pedregosa et al., 2011).

For NMF we used the nonnegative double SVD initialization. We trained 250 and 500 dimensional word vectors with each method.

3.3 Evaluation

We evaluate word vectors on two tasks: similarity and analogy. A similarity is tested using the WordSim353 dataset of Finkelstein et al. (2002), containing word pairs with human-assigned similarity scores. Each word pair is ranked by cosine similarity and the evaluation is the Spearman correlation between those rankings and human ratings. Analogies are tested using Mixed dataset of 19544 questions such as "*a* is to *b* as *c* is to *d*", where *d* is hidden and must be guessed from the entire vocabulary. We filter questions with out of vocabulary words, as standard. Accuracy is computed by comparing $\arg\min_d \|\mathbf{b} - \mathbf{a} + \mathbf{c} - \mathbf{d}\|$ to the labelled answer.

4 Results

The results of evaluation are provided in Table 1, which we analyze using the two-factor ANOVA with factors being (1) low-rank approximation method and (2) dimensinality of word vectors, and response being the performance in similarity or analogy task. We analyze the tasks separately.

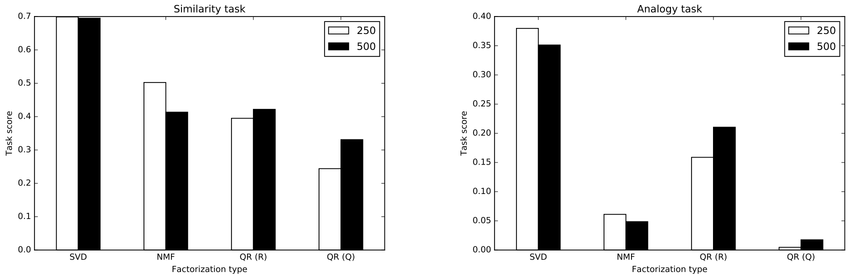
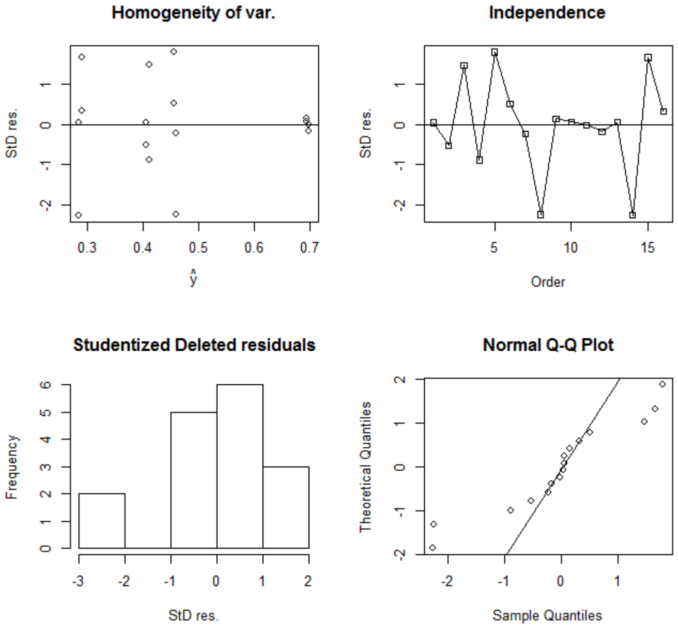


Fig. 1. Test scores for different factorization methods on Similarity and Analogy tasks.

Table 1. Results

Method of low-rank approximation	Dimensionality of vectors	Replicate #	Similarity task	Analogy task
SVD	250	1	0.7010	0.3778
SVD	250	2	0.6969	0.3817
SVD	500	1	0.6989	0.3568
SVD	500	2	0.6914	0.3458
NMF	250	1	0.5265	0.0660
NMF	250	2	0.4780	0.0563
NMF	500	1	0.4499	0.0486
NMF	500	2	0.3769	0.0487
QR (R)	250	1	0.4077	0.1644
QR (R)	250	2	0.3822	0.1533
QR (R)	500	1	0.4717	0.2284
QR (R)	500	2	0.3719	0.1925
QR (Q)	250	1	0.2870	0.0034
QR (Q)	250	2	0.2009	0.0059
QR (Q)	500	1	0.3573	0.0165
QR (Q)	500	2	0.3048	0.0186


Homogeneity of var.

Independence

Studentized Deleted residuals

Normal Q-Q Plot

Fig. 2. ANOVA residuals for the results on Similarity task.

4.1 Similarity task

The standard residual analysis is used to check whether the ANOVA assumptions are satisfied. From Figure 2 we see that the residuals have constant variability around zero, are independent and normally distributed. The normality is confirmed using Shapiro-Wilk test, p-value = 0.7923.

Table 2. ANOVA table for the similarity task results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	0.37055017	0.05293574	29.68	< .0001
Error	8	0.01426728	0.00178341		
Corrected total	15	0.38481745			
	R-Square	Coeff Var	Root MSE	Score Mean	
	0.962925	9.126819	0.042230	0.462707	

Table 3. Main and Interaction Effects in the Similarity task

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Factorization	3	0.35433596	0.11811199	66.23	< .0001
Dimension	1	0.00011159	0.00011159	0.06	0.8088
Interaction	3	0.01610263	0.00536754	3.01	0.0945

Table 4. ANOVA Table for the Analogy task results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	0.30745304	0.04392186	424.61	< .0001
Error	8	0.00082753	0.00010344		
Corrected total	15	0.30828057			
	R-Square	Coeff Var	Root MSE	Score Mean	
	0.997316	6.602449	0.010171	0.154043	

The SAS package was used to obtain ANOVA table (Table 2), which shows the effects of the factors on the similarity score. F-test for equality of the factor level means was conducted, $F = 29.68$ and p-value < 0.0001. Hence, it can

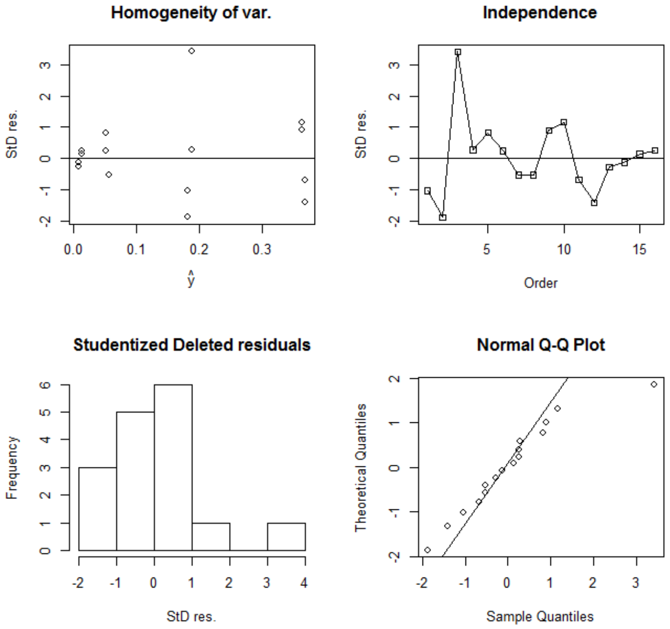


Fig. 3. ANOVA Residuals for the Analogy task results

Table 5. Main and Interaction Effects in the Analogy task

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Factorization	3	0.30365768	0.10121923	978.52	< .0001
Dimension	1	0.00013820	0.00013820	1.34	0.2811
Interaction	3	0.00365715	0.00121905	11.78	0.0026

be concluded that at least one factor level mean is different from the others. $R^2 = 0.962925$ shows that more than 96% of variation in the similarity score is explained by the factors considered.

Proceeding with analysis of main and interaction effects, one can conduct F-test for each of the factors and the interaction between them. From Table 3, we see that the method of low-rank approximation affects the performance of words vectors in the similarity task, $F = 66.23$, p-value < 0.0001. The dimensionality of word vectors has no effect on the performance in the similarity task, $F = 0.06$ with p-value > 0.8. Also, there is no interaction between the method of factorization and the dimensionality of word vectors, $F = 3.01$ with p-value 0.0945. Thus, SVD significantly outperforms the other factorization methods.

4.2 Analogy task

Again, we first need to check whether the ANOVA assumptions are satisfied. From Figure 3 we see that the residuals have constant variability around zero, are independent and normally distributed. The normality is confirmed using Shapiro-Wilk test, p-value = 0.112. The ANOVA Table (Table 4) shows that at least one level mean is different from the others. R^2 is 0.997316, thus, 99% of variation in the analogy score is explained by the considered factors.

We proceed to the analysis of main and interaction effects. The method of low-rank approximation affects the performance of word vectors in the analogy task, $F = 978.52$ with p-value < 0.0001. The dimensionality of word vectors has no effect on the performance in the analogy task, $F = 1.34$ with p-value > 0.2. Unlike the similarity task, there is an interaction effect between the two factors, $F = 11.78$ with p-value $= 0.0026$.

5 Discussion

Why dimensionality is critical in similarity task for NMF? We obtained the highest results in the similarity task using the SVD-based low-rank approximation, for which the dimensionality of word vectors did not influence the performance much. On the contrary, the performance in similarity task using the NMF method of factorization is significantly affected by the dimension of the word vector: 250-dimensional word vectors give significantly better results than 500-dimensional ones. This can be explained by the specific characteristics of the NMF method of factorization. When we look at the word vectors produced by NMF, we can see that they contain many zeros. Hence, an increase in the dimensionality makes them even sparser. Similarity task is based on finding the cosine of the angle between two word vectors. Therefore, when the vectors become sparser, the result of element-wise multiplication, which is necessary for obtaining cosine, becomes smaller. Thus, there is a much higher possibility that the cosine similarity score between two vectors, containing many zeros, will give a number closer to zero than to 1. This, as a result, leads to the worse performance in the similarity task. Our suggestion is to decrease the dimensionality of the NMF method to 100. We expect that this may give better results.

Why NMF performs poorly in the analogy task? We provide a theoretical analysis of the poor performance of the NMF in the analogy task. We model word vectors produced by the NMF as independent and identically distributed random vectors from an isotropic multivariate Gaussian distribution $\mathcal{N}(\mathbf{4.5}, \mathbf{I})^4$, since for a 500-dimensional $\mathbf{v} \sim \mathcal{N}(\mathbf{4.5}, \mathbf{I})$ there is a big chance that it is nonnegative:

$$\Pr(\mathbf{v} \in [0, +\infty)^{500}) = [\Pr(4.5 + Z > 0)]^{500} \approx 0.9983,$$

where $Z \sim \mathcal{N}(0, 1)$ is a standard normal random variable. For a triplet of word vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ we have $\mathbf{b} - \mathbf{a} + \mathbf{c} \sim \mathcal{N}(\mathbf{4.5}, 3\mathbf{I})$, and therefore

$$\begin{aligned} &\Pr(\mathbf{b} - \mathbf{a} + \mathbf{c} \in [0, +\infty)^d) = [\Pr(3 + \sqrt{3}Z \geq 0)]^d \\ &= [\Pr(Z \geq -4.5/\sqrt{3})]^d < [0.9953]^d. \end{aligned}$$

When $d = 500$, this probability is ≈ 0.1, i.e. there is a small chance that $\mathbf{b} - \mathbf{a} + \mathbf{c}$ is non negative, and thus we will likely not find a non-negative $\mathbf{d}$ when we minimize $\|\mathbf{b} - \mathbf{a} + \mathbf{c} - \mathbf{d}\|$. This is confirmed empirically: for *all* word triplets (a, b, c) from the analogy task, the vector $\mathbf{b} - \mathbf{a} + \mathbf{c}$ has at least one negative component.

Why using R is better than using Q in the QR decomposition? The $\mathbf{Q}$ matrix from QR factorization gives the worst results in the similarity task, and it does not depend on the dimensionality of the vector. The reason is that the necessary information is left in the $\mathbf{R}$ matrix. Truncation of $\mathbf{R}\mathbf{P}^\top$ gives better approximation to the original matrix than the truncated $\mathbf{Q}$, because the most significant entries of $\mathbf{R}\mathbf{P}^\top$ are in the top left quarter and remain after truncation.

6 Conclusion

We analyzed the performance of the word vectors obtained from a word-word PMI matrix by different low-rank approximation methods. As it was expected, the truncated SVD provides a far better solution than the NMF and the truncated QR in both similarity and analogy tasks. While the performance of the NMF is relatively good in the similarity task, it is significantly worse in the analogy task. NMF produces only non-negative sparse vectors and we showed how this deteriorates the performance in both tasks. $\mathbf{R}\mathbf{P}^\top$ matrix from QR factorization with column pivoting gives better word embedding than $\mathbf{Q}$ matrix in both tasks.

Acknowledgement

The work of Zhenisbek Assylbekov has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan, contract # 346/018-2018/33-28, IRN AP05133700.

[4] The isotropy is motivated by the work of Arora et al. (2016); **4.5** is a vector with all elements equal to 4.5.

Bibliography

[1] Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: A latent variable model approach to pmi-based word embeddings. Transactions of the Association for Computational Linguistics 4, 385–399 (2016)

[2] Davis, T.A., Hu, Y.: The university of florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS) 38(1), 1 (2011)

[3] Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika 1(3), 211–218 (1936)

[4] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. ACM Transactions on information systems 20(1), 116–131 (2002)

[5] Jones, E., Oliphant, T., Peterson, P.: {SciPy}: open source scientific tools for {Python} (2014)

[6] Kishore Kumar, N., Schneider, J.: Literature survey on low rank approximation of matrices. Linear and Multilinear Algebra 65(11), 2212–2244 (2017)

[7] Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems. pp. 2177–2185 (2014)

[8] Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. TACL 3, 211–225 (2015)

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research 12(Oct), 2825–2830 (2011)