

# Word embedding in form of symmetric and skew-symmetric operator

Koshchenko Ekaterina  
National Research University  
Higher School of Economics in Saint-Petersburg  
Saint-Petersburg, Russia  
catherine.pths@gmail.com

Kuralenok Igor  
JetBrains Research  
Saint-Petersburg, Russia  
ikuralenok@gmail.com

**Abstract**—Existing word embedding models represent each word with two real-valued vectors: central and context. This happens because of words relations asymmetric nature and requires more time and data for training. We introduce a new approach based on asymmetric relations that uses the advantages of global vectors model. Due to the reduction of asymmetric information impact on resulting words representations, our model converges faster and outperforms existing models on words analogies tasks.

**Index Terms**—SSDE, word embedding, matrix decomposition

## I. INTRODUCTION

Understanding words relations in the context of natural language is an easy task for human but not for computer. We need to teach computers how words are related and what meanings they have, depending on the context. To make it possible for a machine to process words, they have to be presented in digitized format. This leads to the idea of real-valued vector representations — word embeddings.

Most works on word embeddings focus their attention on preserving two words properties in their representations. The first property is that words relations and similarities can be described using distances and angles between word vectors. For example, closer-further feature: “yellow” is closer to “red” than to “smart”. In vector form it can be presented as

$$Yellow - Red < Yellow - Smart$$

This property is widely used for synonyms search. Another property is words analogies. The corresponding feature was introduced by Mikolov et al. [1], designed to learn words similarities. For example, “Paris” and “France” has the same connection as “Budapest” and “Hungary”. In vectors we can present it as

$$France - Paris = Hungary - Budapest.$$

This approach benefits models creating meaning based word vectors, while the closer-further feature is more practical and can be applied to clustering and classification tasks.

Word embeddings were originally created to be used in Natural Language Processing tasks. For example, one

of the feature extraction techniques used for document indexing is latent semantic indexing [2]. Latent semantic indexing is a precursor for word embeddings embodying the same principles and ideas. Another task is sentiment analysis. One of the solutions for this problem is SentProp framework [3], it combines label propagation method with word embeddings to learn sentiment lexicons on domain-specific corpora. Another way to solve some of the Natural Language Processing tasks are Language Models. Nowadays state of the art decisions for Language Modeling are ELMO [4] and BERT [5]. Each of these methods uses prebuilt word embeddings as input data and can benefit from better embedding models. Therefore, creating better embedding models is still a relevant task.

There are three most popular and used word embedding models. Word2Vec is a local window-based method presented by Mikolov et al. [6]. It preserves words analogies feature, bringing closer vectors of words appearing in a similar context. Another approach is GloVe [7] which is trained on word-word co-occurrence counts. Authors noticed that to understand the relation of two words you can examine the ratio of their co-occurrence probabilities with various probe words, thus deploying words analogies feature. Third model – FastText [8] – is focused on distances/angles property. FastText uses character  $n$ -grams to enrich word vectors with subword information. This approach allows to use morphology information, therefore, choosing better vectors for sparse words and makes it possible to learn something for non-vocabulary words.

Words relations are often asymmetrical. For example, "New York" is a common combination of words meaning the name of the city in the USA. However, "York New" is a quite rare combination and does not mean anything specific. In all mentioned models words interaction is expressed in terms of the dot product of their vectors, that leads to a generation of two vectors for each word: central and context. For that reason, twice more parameters should be computed and, consequently, more time is required for learning. To solve this problem asymmetrical relations between word representation can be used instead of central and context vectors dot product.

In this work, we propose a Symmetric Skew-symmetric

Decomposition based model. We demonstrate that our method outperforms GloVe approach on its words analogies metrics.

## II. RELATED WORK

There are many word embedding models known from the literature. But most of them were based on three principle approaches: Word2Vec [6], GloVe [7] and FastText [8]. All three models are widely used in language models and Natural Language Processing applications.

### A. Word2Vec

Word2Vec is an approach introduced by Mikolov et al. [6] that preserves words analogies property. It suggests two language models: Skip-gram and CBOW. Both methods represent words relationships with the dot product of their vectors. As it was described in the introduction, relations can be asymmetrical, which leads to two vectors per word usage: central and context. Skip-gram and CBOW scan corpus with a sliding window. All words inside the window are considered to be in the same context, i.e. connected to each other. In both models all words inside one window get the same co-occurrence weight, i.e. are equal. We call this type of window "constant window".

Continuous Bag of Words (CBOW) is a model trained with "predict middle-word if you know surrounding context" task. The method tries to choose words central and context vectors, so that probability to predict the word in the middle of the sliding window, based on the rest of the window, would be high. The second model is called Skip-gram and is trained on the inverse problem: predict context with just one word in the center of the sliding window.

For each training step for each word, both methods should count the probability of using window middle-word in context with any other word from the vocabulary. It makes computational complexity too high. In later article [9] this problem was solved for Skip-gram model with Negative Sampling. Negative Sampling suggests counting the probability of middle-word being in the same context only with a constant number of positive and negative samples. Positive samples are words that often appear in one window with middle-word, they can be found before the training process. Negative samples are words that are unlikely to appear in context with middle-word. Mikolov et al. suggest getting negative samples from uniform distribution raised to 3/4rd power. This approach allows accelerating Skip-gram model calculations while being of the same quality.

Results of experiments have shown that Skip-gram method performs better on semantic tasks and their syntactic tasks results are very similar. Since Skip-gram can be trained easier than CBOW with same or even better results, later models use Skip-gram.

Skip-gram and CBOW models have several drawbacks. First, training time depends on the corpus size. Second,

there are two vectors generated for each word, which requires more time and input data for training.

### B. GloVe

GloVe model, for Global Vectors, suggested by Manning et al. also aims to preserve words analogies. The relationship of two words can be learned by examining their relations with other words. In this approach words relationships are represented with a matrix of their co-occurrences  $X$ , where  $x_{ij}$  is how many times word  $w_i$  was in the context with word  $w_j$ . This matrix should be constructed before the training process with one scan of the corpus. On each learning step we iterate through co-occurrences matrix and for each non-zero co-occurrence  $x_{ij}$  calculate central and context vectors for corresponding  $w_i$ , according to value and direction of target function gradient.

In GloVe each word is presented with two vectors, similar to Word2Vec. A sliding window is also used to scan the corpus for co-occurrences matrix construction. Unlike the Word2Vec "constant" window, GloVe uses "shrinking" window. The weight of co-occurrence in the window linearly decreases with distance increasing. Authors did not explore how window type affects experiments results and did not give any details on such a choice.

### C. FastText

FastText model, in contrast to Word2Vec and GloVe, was built to preserve words property of representing words relations in distances and angles between their vectors. This change allows the model to perform better on text classification tasks. Similar to two previous methods, FastText generates central and context vectors for each word and uses a sliding window to scan the corpus.

The main idea of this approach is to use character  $n$ -grams to build central vectors. During the vocabulary construction, each word is saved with its  $n$ -grams. For example, for the word "pencil" we also remember 3-grams "pe", "pen", "enc", "nci", "cil" and "il" in addition to the whole word sequence. 3-gram "pen" corresponding to the word "pencil" is different from the word "pen". After that, during the training process, each sequence gets its own vector and resulting central vector is a sum of all  $n$ -gram vectors and whole word vector.

As it was mentioned, FastText has great results on text classification tasks but Word2Vec and GloVe outperform it on words analogies tasks.

## III. THE SSDE MODEL

Words relations have asymmetric nature, for that reason all three approaches above generate two vectors for each word. The question is how to apply these central and context vectors. In GloVe, for example, there are several modes for what to use as a resulting vector. The default mode is a sum of central and context vectors. There was no intuition for this choice, although our experiments have

shown that the default mode indeed performs best. It is possible that Word2Vec, GloVe and FastText use more parameters than they really need, which means more time and input data is required for training. The subject of our research was to find out if words asymmetric information is really necessary to include into the resulting vector. To do that we introduce a Symmetric Skew-symmetric Decomposition Embedding (SSDE). It is based on GloVe model, mainly because it is faster than other existing models and performs better on word analogies metrics.

#### A. GloVe model analysis

The main idea of GloVe model: words  $w_i$  and  $w_j$  relation can be found by studying the ratio of their co-occurrence probabilities with various probe words –  $P(w_i, w_k)/P(w_j, w_k)$ , where  $w_k$  is a probe word. So, general model can be written as

$$F((u_i - u_j)^T v_k) = \frac{P(w_i, w_k)}{P(w_j, w_k)}. \quad (1)$$

Authors say that due to exchangeability of words and context words function  $F$  should be a homomorphism:

$$F((u_i - u_j)^T v_k) = \frac{F(u_i^T v_k)}{F(u_j^T v_k)}. \quad (2)$$

This formula gives an idea that model  $F$  is exponential, which in combination with Eqn. (1) leads to:

$$u_i^T v_k = \log P_{ik} = \log X_{ik} - \log X_i. \quad (3)$$

After that GloVe brings biases to the formula.  $\log X_i$  does not depend on probe word  $k$  and is replaced with bias  $b_i^u$ . For word-context exchange symmetry context bias  $b_k^v$  is also included:

$$u_i^T v_k + b_i^u + b_k^v = \log X_{ik}. \quad (4)$$

In this equation, right-hand side is what information model has to learn and left-hand side is how GloVe preserves it. This is optimized with weighted least squares regression model. As a result, GloVe model target function is

$$J = \sum_{i,j=1}^{|V|} f(X_{ij}) \cdot (u_i^T v_j + b_i^u + b_j^v - \log X_{ij})^2, \quad (5)$$

where

- $X$  – co-occurrences matrix,
- $|V|$  – vocabulary size,
- $u_i$  and  $b_i^u$  – central vector and bias for word  $w_i$ ,
- $v_j$  and  $b_j^v$  – context vector and bias for word  $w_j$ .

Introduction of encoding and decoding biases is a moment that has no mathematical demonstration in the article, but our experiments have shown that the model does not work without their usage. We explained this with target function similarity with mutual information formula:

$$D_{KL} = \sum_{i,j} p(w_i, w_j) \cdot \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \quad (6)$$

We are actually looking for embedding that will preserve the ratio in logarithmic part of Eqn. (6). The ratio represents how much more often a combination of words  $x$  and  $y$  occurs in corpus than each of them individually. Information that the model encodes is a conditional probability given model  $F$ :

$$I = \sum_{i,j} p(w_i, w_j) \cdot \log \frac{p(w_i, w_j|F)}{p(w_i|F)p(w_j|F)}. \quad (7)$$

The result of rewriting Eqn. (6) and Eqn. (7) in GloVe notation and combining with the weighted least squares method will be very similar to GloVe target function:

$$\begin{aligned} \frac{p(w_i, w_j|F)}{p(w_i|F)p(w_j|F)} &\Rightarrow e^{u_i^T v_j} \\ \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} &\Rightarrow \log X_{ij} - b_i^u - b_j^v \\ J^* &= \sum_{i,j} p(w_i, w_j) \cdot (u_i^T v_j + \log p(w_i) + \log p(w_j) - \\ &\quad - \log p(w_i, w_j))^2, \end{aligned} \quad (8)$$

Joint probability of words  $w_i$  and  $w_j$  are what in GloVe model is designed as co-occurrences matrix  $X_{ij}$  and prior probabilities of words are designed as biases  $b_i^u$  and  $b_j^v$ . In our experiments we tried both ways and obtained similar results for biases and probabilities usage. For that reason, we continued using prior probabilities in SSDE to decrease computational complexity.

#### B. Our model

From Eqn. (5) we see that GloVe represents words relations with dot product of their central and context vectors:  $u^T v$ . This is done to consider the asymmetry property that we want to remove. Central and context vectors dot product is equal to corresponding one-hot encoder vectors multiplication to central and context matrices product. Central and context matrices product can be considered as linear operator, and any linear operator can be decomposed to sum of symmetric and skew-symmetric matrices [10]:

$$\begin{aligned} u_i^T v_j &= h_i U^T V h_j \\ L &= U^T V = S + K \end{aligned} \quad (9)$$

After that symmetric matrix  $S$  (according to the property of symmetric matrices) can be written as a product of some low-rank matrix and its transpose. The same transformation can be used for the skew-symmetric matrix  $K$  with multiplying lower-diagonal part to  $-1$ .

$$l_{ij} = s_{ij} + k_{ij} = a_i^T a_j + \xi_{ij} \cdot c_i^T c_j, \quad (10)$$

The size of a matrix  $A$  is  $|V| \cdot l$  where  $|V|$  - size of vocabulary,  $l$  - word symmetric representation size. The size of a matrix  $C$  is  $|V| \cdot m$  where  $m$  - word asymmetric representation size. Balancing between symmetric and skew-symmetric sizes we control the information distribution

the way we need. For example, to reduce the influence of asymmetric information on resulting word representation we make constant  $m$  much smaller than  $l$ .

In total, after rewriting GloVe target function (5) with Eqn. (10) and using the prior probabilities instead of biases, we get SSDE model target function:

$$Q = \sum_{i,j=1}^{|V|} f(p_{ij}) \cdot (a_i^T a_j + \xi_{ij} \cdot c_i^T c_j + \log p_i + \log p_j - \log p_{ij})^2, \quad (11)$$

- $p_{ij} = p(w_i, w_j)$  and  $p_i = p(w_i)$  – are counted from the input corpus before the training process
- $\xi_{ij} = -1$ , if  $i > j$ , otherwise  $\xi_{ij} = 1$

On each training step we iterate through word-word co-occurrences matrix  $X$ . Each co-occurrence  $x_{ij}$  shows how many times word  $w_i$  was in the context with word  $w_j$ . We compute gradients for symmetric vectors and skew-symmetric vectors and update them according to the gradients.

Resulting word embeddings are vectors of symmetric matrix  $A$ . Since we wanted to remove asymmetric information influence on resulting word representations, vectors  $c_i$  are only used for training. However, their properties worth further studying.

There are two ways to optimize function (11): 1) gradient descent, 2) stochastic gradient descent. The advantage of gradient descent is that it will eventually converge to better results. Though stochastic gradient has several methods that achieve reasonable results much faster than gradient-descent. Since we wanted to reduce training time, we decided to use GloVe’s approach using adaptive gradient descent. GloVe authors also noticed that values slightly change on each stochastic gradient iteration which means computations can be done in parallel.

GloVe model shuffles whole co-occurrences matrix on each step of stochastic gradient descent.

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} f(X_{ij}) \cdot (u_i^T v_j + b_i^u + b_j^v - \log X_{ij})^2 \quad (12)$$

$$= \mathbb{E}_{i,j \sim U(X)} f(X_{ij}) \cdot (u_i^T v_j + b_i^u + b_j^v - \log X_{ij})^2.$$

In SSDE model we shuffle only lines of co-occurrences matrix.

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} f(p_{ij}) \cdot (a_i^T a_j + \xi_{ij} \cdot c_i^T c_j + \log p_i + \log p_j - \log p_{ij})^2$$

$$= \mathbb{E}_i \sum_{j=1}^{|V|} f(p_{ij}) \cdot (a_i^T a_j + \xi_{ij} \cdot c_i^T c_j + \log p_i + \log p_j - \log p_{ij})^2. \quad (13)$$

Lines shuffle without columns shuffle makes computations cash-friendly, reducing cash-miss rate. This change allowed

us to optimize model performance while quality remained the same.

## IV. EXPERIMENTS

### A. Evaluation

To compare SSDE with GloVe we used metrics suggested in GloVe article. All the metrics are based on word analogies property. There are four words  $w_1, w_2, w_3, w_4$ , all associated with one topic and can be described as “ $w_1$  is related to  $w_2$  the same way  $w_3$  is related to  $w_4$ ”. This can be presented in vectors terms as

$$w_2 - w_1 = w_4 - w_3.$$

According to the arithmetics law this can be rewritten as

$$w_2 - w_1 + w_3 = w_4 (*).$$

Testing algorithm is: 1) get first three input words and count left part of (\*) 2) among all vectors of our vocabulary find the closest vector  $v$  to the previous step result (using cosine similarity) 3) if word corresponding to  $v$  is equal to  $w_4$ , then this experiment was successful, otherwise it failed.

We do not provide a comparison with CBOW or Skip-gram models, but, as it is shown in the article [7], GloVe performs better than the other baselines.

‘Tab. I’ shows all metrics that were used to evaluate both GloVe and SSDE models. Five of these metrics have semantic nature, for example,

$$”King” - ”Man” + ”Woman” = ”Queen”.$$

While the other nine are syntactic, for example,

$$”Dangerous” - ”Danger” + ”Beauty” = ”Beautiful”.$$

### B. Results

We compared GloVe and SSDE models on corpus composed of 100Mb of articles from English Wikipedia. For corpus scanning we used symmetric shrinking window of size 30. All models were trained up to convergence. Studying of the constant window and asymmetric window results will be completed in future work.

Tab. II shows the performance of GloVe and SSDE models with an equal number of parameters trained. Our approach significantly improves scores both for semantic and syntactic tasks.

Tab. III shows results of GloVe and SSDE models with equal sizes of word embeddings vectors. As it was mentioned, GloVe model uses a sum of central and context vectors as the resulting representation and SSDE model uses only a symmetric vector. Similar or even higher scores can be obtained with SSDE model with the same representation size as GloVe, but almost twice a smaller number of parameters.

All the results were obtained on Inter Core i7 processor, 8GB, DDR4 memory type.

Table I  
GLOVE WORD ANALOGIES METRICS

Type	Name	Example	Size
Semantic	Capital common countries	Greece to Athens as Iraq to Baghdad	506
Semantic	Capital world	Nigeria to Abuja as Ghana to Accra	4525
Semantic	City in state	Illinois to Chicago as Texas Houston	2467
Semantic	Currency	Dinar to Algeria as Kwanza to Angola	866
Semantic	Family	Brother to Boy as Sister to Girl	506
Syntactic	Adjective to adverb	Calm to calmly as Happy to Happily	992
Syntactic	Opposite	Aware to Unaware as Clear to Unclear	812
Syntactic	Comparative	Worse to Bad as Bigger to Big	1332
Syntactic	Superlative	Worst to Bad as Biggest to Big	1122
Syntactic	Present Participle	Coding to Code as Dancing to Dance	1056
Syntactic	Nationality Adjective	China to Chinese as Poland to Polish	1599
Syntactic	Past Tense	Danced to Dancing as Flew to Flying	1560
Syntactic	Plural	Bananas to Banana as Birds to Bird	1332
Syntactic	Plural Verbs	Eats to Eat as Says to Say	870
Total			19545

Table II  
EXPERIMENTS WITH EQUAL PARAMETERS NUMBER

Model	Average sem score	Average synt score	Average total score	Sec. per iter.
GloVe-25	12.9%	12.3%	12.5%	13
SSDE-40-5	19.1%	22.5%	20.4%	11
SSDE-40-10	19.6%	22.6%	20.8%	12
GloVe-50	20.3%	29.4%	23.9%	18
SSDE-80-5	24.3%	39.8%	30.6%	17
SSDE-80-10	25.1%	39.7%	31.0%	18
SSDE-80-20	25.5%	41.4%	31.9%	20
GloVe-80	25.3%	40.2%	31.3%	24
SSDE-120-10	25.3%	48.1%	34.5%	25
SSDE-120-20	26.3%	47.8%	<b>35.1%</b>	27

<sup>a</sup>"GloVe- $n$ " – vector size  $n$ .

<sup>b</sup>"SSDE- $m$ - $l$ " – symmetric size  $m$ , asymmetric size  $l$ .

We demonstrated that our approach outperforms GloVe model on word analogies metrics while calculating a twice smaller number of parameters. This fact proves our initial assumption that asymmetric information influence on word embeddings can be significantly reduced, thus, optimizing time required for training of the model.

## V. CONCLUSION

### A. Achievements

In this paper, we studied the necessity of word relationships asymmetric information for word embeddings.

Table III  
EXPERIMENTS WITH EQUAL REPRESENTATIONS SIZE

Model	Average sem score	Average synt score	Average total score	Sec. per iter.
GloVe-50	20.3%	29.4%	23.9%	18
SSDE-50-5	22.5%	28.7%	25.0%	12
SSDE-50-10	21.8%	28.3%	24.3%	14
SSDE-50-20	21.7%	28.5%	24.5%	15
GloVe-80	25.3%	40.2%	31.3%	24
SSDE-80-5	24.3%	39.8%	30.6%	17
SSDE-80-10	25.1%	39.7%	31.0%	18
SSDE-80-20	25.5%	41.4%	31.9%	20
GloVe-120	27.6%	44.0%	34.2%	34
SSDE-120-10	25.3%	48.1%	34.5%	25
SSDE-120-20	26.3%	47.8%	<b>35.1%</b>	27
GLOVE-150	27.8%	45.2%	34.8%	79

<sup>a</sup>"GloVe- $n$ " – vector size  $n$ .

<sup>b</sup>"SSDE- $m$ - $l$ " – symmetric size  $m$ , asymmetric size  $l$ .

We showed that it is possible to train high-quality word vectors using a little information on the asymmetry of relations, comparing to the popular word embedding model with highest scores on word analogies tasks – GloVe. Since our approach computes a twice smaller number of parameters, it requires less time to train the model.

We analyzed GloVe model and introduced a new model – SSDE – that combines the advantages of GloVe with our ideas on asymmetric relations. Comparison of SSDE with GloVe has shown that our model outperforms GloVe on word analogies metrics, while GloVe, according to the article [7], outperforms CBOW and Skip-gram models.

### B. Future work

SSDE model, similar to GloVe and Word2Vec, uses a sliding window to scan the corpus. We assume that depending on the type of the window used, results may be different for metrics of different types. Constant windows might perform better on synonyms search tasks, while the shrinking window could be a good choice for word analogies tasks. So, in future work, we will examine window type influence on different metrics types.

Currently, we only use vectors with symmetric information for resulting word embeddings. However, there might be some interesting information encoded in asymmetric vectors. For example, L1-regularization turn most of the skew-symmetric vectors to zero. There might be some connection between those words which corresponding skew-symmetric vectors are not zero. In future work, we will study the asymmetric component of SSDE and analyze if there is any pattern that might increase performance on some tasks.

Window size and symmetry influence on model performance is another aspect that was not examined. Importance of asymmetric information might increase for highly asymmetric windows.

## REFERENCES

- [1] T. Mikolov, W.-t. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations.” in *HLT-NAACL*, 2013, pp. 746–751.
- [2] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002. [Online]. Available: <http://nmis.isti.cnr.it/sebastiani/Publications/ACMCS02.pdf>
- [3] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, “Inducing domain-specific sentiment lexicons from unlabeled corpora,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 595–605. [Online]. Available: <http://aclweb.org/anthology/D16-1057>
- [4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proc. of NAACL*, 2018.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>
- [7] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: <http://www.aclweb.org/anthology/D14-1162>
- [8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.
- [10] F. Gantmacher, *The theory of matrices*, ser. The Theory of Matrices. Chelsea Pub. Co., 1960, no. т. 1. [Online]. Available: <https://books.google.ru/books?id=GOdQAAAAMAAJ>