A New Algorithm of Word Embedding Via PSD-matrix Factorization

Tikhonova Maria

Higher School of Economics, Sberbank

m_tikhonova94@mail.ru

Introduction

Word embeddings are an unsupervised Natural Language Processing technique.

The task is to derive a set of vectors corresponding to words.

Two categories of word embeddings methods

Neural word embedding models
- Bengio Y. et al., 2003
- Mikolov T. et al., 2013a
- Mikolov T. et al., 2013b

Low-rank matrix factorization based methods
- Levy O., Goldberg Y., 2014
- Dhillon P. S. et al., 2015
- Li S. et al., 2015

Matrix factorization methods reconstruct bigram statistics matrix X as

$$\hat{X} = VC^T$$

$\hat{X}$ – reconstruction of the statistics matrix

V – word matrix

C – context matrix

Shortcomings of matrix factorization based methods:

- different matrices for words and their contexts: $V \neq C$
- freedom of choice: $VC = (VS)(S^{-1}C)^T = \tilde{V}\tilde{C}^T$

Idea: choose Positive Semidefinite matrix (PSD-matrix) for X so

$$\hat{X} = VV^T$$

Result: $V = C$ **and** NO freedom of choice

Positive SemiDefinite Spectral (PSDS) embeddings

- matrix factorization method
- eigendecomposition of a PSD-matrix matrix X

Algorithm PSDS

Step 1. Enumerate words; for each word define its neighbourhood

Step 2. Define $n \times N$ matrix A

$$A_{ij} = \begin{cases} 1, & if\ i-th\ word\ belongs\ to\ the\ j-th\ neighbourhood \\ 0, & otherwise \end{cases}$$

n – vocabulary size

N – number of all the words

Step 3. Define matrix X

$$X := AA^T \text{ - PSD by construction}$$

Step 4. Find k maximal eigenvalues $\lambda_1, \ldots \lambda_k$ of X and corresponding eigenvectors $v_1, \ldots v_k$

Step 5. Define $n \times k$ word embedding matrix

$$V := \left(\sqrt{\lambda_1}v_1, \ldots, \sqrt{\lambda_k}v_k\right)$$

***i*-th row of *V* is the *k*-dimensional word embedding of the *i*-th word.**

Algorithm's modification PSDS-zeroed

Additional step 3.5. Diagonal of X is zeroed

$$X_{ii} = 0$$

Experiments

Corpus: dump of SimpleWikipedia.

3 vocabulary sizes: 10000, 50000 and 100000 words.

Algorithms: PSDS, PSDS-zeroed, Word2Vec SGNS.

Metric: *Mean Average Precision (map@K)* for ranking word pairs in *Simlex-999*.

Word pairs in SimLex-999

Word 1	Word 2	Similarity rating
new	old	1.58
hard	difficult	8.77

100-dimensional word embedding

Algorithm	10 K	50 K	100 K
PSDS	0.657	0.657	0.657
PSDS-zeroed	**0.6793**	**0.6788**	**0.6788**
Word2Vec	0.6381	0.6285	0.623

200-dimensional word embedding

Algorithm	10 K	50 K	100 K
PSDS	0.6719	0.6719	0.6719
PSDS-zeroed	**0.6891**	**0.6891**	**0.6891**
Word2Vec	0.6553	0.6512	0.6451

Conclusions

- PSDS-zeroed outperforms other methods
- basic version of PSDS works well
- PSDS and PSDS-zeroed are stable to vocabulary size change
- good word embeddings with only 10000 words

Closest words

PSDS maps similar words into close vectors (by cosine similarity).

Word	Closest words
Russia	Britain, Poland, Austria, Scotland, Egypt
algebra	arithmetic, formulas, differential, topology, cryptography
football	league, hockey, played, club, professional
dog	cat, horse, self, bear, wild

References

Bengio Y. et al., 2003 -- neural probabilistic language model //Journal of machine learning research. – 2003. – Т. 3. – №. Feb. – С. 1137-1155.

Dhillon P. S. et al., 2015 -- Eigenwords: Spectral word embeddings //The Journal of Machine Learning Research. – 2015. – Т. 16. – №. 1. – С. 3035-3078.

Levy O., Goldberg Y., 2014 -- Neural word embedding as implicit matrix factorization //Advances in neural information processing systems. – 2014. – С. 2177-2185.

Mikolov T. et al., 2013a -- Efficient estimation of word representations in vector space //arXiv preprint arXiv:1301.3781. – 2013.

Mikolov T. et al., 2013b -- Distributed representations of words and phrases and their compositionality //Advances in neural information processing systems. – 2013. – С. 3111-3119.

Li S. et al., 2015 -- A generative word embedding model and its low rank positive semidefinite solution //arXiv preprint arXiv:1508.03826. – 2015.