Optimization Task in Equivalent to word2vec Matrix Factorization

Victor Kantor[1,2]

[1] Moscow Institute of Physics and Technologies
[2] Yandex
Moscow, Russia
viktor.kantor@phystech.edu

Abstract. Omer Levy and Yoav Goldberg have shown in their paper at NIPS 2014 that word2vec is equivalent to the factorization of shifted PMI matrix. The question which was not discussed in this work and later papers is the right choice of the norm for an approximation of the matrix. Authors also presented the results of the experiments with SVD approximating the matrix with respect to Frobenius norm. In this work we show that weighted Frobenius norm could be the reasonable choice, but weights shouldn't be equal to one as in Levy and Goldberg experiments. We conjecture that the right choice of weights could help to improve matrix factorization results on analogy questions, where skip-gram with negative sampling (SGNS) remains superior to SVD.

Keywords: word2vec, SGNS, matrix factorizations, SVD, distributional semantics

1 Introduction

Word2vec is a powerful and popular natural language processing technique proposed by Mikolov et al. in [1]. It allows to get word representations with some useful properties. Dot product of word2vec vectors is a good similarity measure and arithmetical operations with vectors help to solve some analogy tasks (popular example: "queen − woman + man ≈ king").

In [3] it was shown that word2vec is similar to matrix factorization technique well-known in NLP and collaborative filtering. And factorizing matrix is almost the PMI-matrix which is also common object in NLP and CF.

The main difference between matrix factorization in [3] and common use of matrix factorization techniques is that Levy et al. result is valid for exact factorization of the matrix but usually we work with approximate factorization in NLP and CF applications.

The default choice of norm for approximating the initial matrix in matrix factorization is Frobenius norm which leads to the quadratic loss:

$$||A||_F = \sqrt{\sum_{i,j} a_{ij}^2} \tag{1}$$

$$||X - UV||_F^2 = \sum_{i,j}(x_{ij} - u_i^T v_j)^2 \tag{2}$$

Experiments from [3] and [4] were considered with Singular Value Decomposition (SVD) which gives the best approximation according to Frobenius norm and subsequently quadratic loss. This choice was motivated by popularity of SVD and Frobenius norm as a default variant. In this work we get the theoretical motivation for quadratic loss using second order Taylor series approximation for objective function. An interesting conclusion is that classic SVD optimizing simple quadratic loss isn't a good choice in this case. Our future work includes experiments which could complement this conclusion with practical results.

2 Skip-Gram with Negative Sampling (SGNS)

In this section we provide a brief review of the result from [3].

2.1 Setting and Notation

The skip-gram model assumes a corpus of words $w \in V_w$ and their contexts $c \in V_c$, where V_w and V_c are sets of words and contexts. We denote the collection of observed words and contexts pair as D. To denote the length of D (the number of words in collection) we use $|D|$. Note that $|D|$ differs from $|V_w|$. We also use $\#(w, c)$ to denote the number of times the pair (w, c) appear in D. Notation $\#(w)$ and $\#(c)$ has the similar meaning.

Let d be the embedding dimensionality. Each word $w \in V_w$ is associated with a vector $w \in \mathbf{R}^d$ and each context $c \in V_c$ is associated with a vector $c \in \mathbf{R}^d$.

2.2 SGNS as Matrix Factorization

As it was shown in [3], if k is the number of negative samples, SGNS optimization task is as follows:

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w, c) \left(\log \sigma(< w, c >) + k \cdot \mathbf{E}_{c_N \sim P_D} \log \sigma(- < w, c_N >)\right) \tag{3}$$

The expectation term:

$$\mathbf{E}_{c_N \sim P_D} \log \sigma(- < w, c_N >) = \sum_{c_N \in V_c} \frac{\#(c_N)}{|D|} \log \sigma(- < w, c_N >) \tag{4}$$

Substituting this expression to 5 we get the local objective for a specific (w, c) pair:

$$\ell(w, c) = \log \sigma(< w, c >) + k \cdot \#(w) \cdot \frac{\#(c)}{|D|} \log \sigma(- < w, c >) \tag{5}$$

Comparing the derivative $\frac{\partial \ell(w,c)}{\partial < w,c >}$ to zero we get:

$$< w, c >= \log \left(\frac{\#(w,c) \cdot |D|}{\#(w)\#(c)} \right) - \log k \tag{6}$$

The first term is well-known as pointwise mutual information (PMI) of (w, c). We denote $PMI_k(w,c) = PMI(w,c) - \log k$, so SGNS is factorizing shifted PMI matrix M^{PMI_k}: $M_{ij}^{PMI_k} = PMI_k(w_i, c_j)$

3 Weighted Frobenius Norm approximation

If we are going to use quadratic loss for this matrix factorization, the reasonable way to approximate the objective function with such loss is to use the second order Taylor series expansion. In this case we get the weighted quadratic loss, so the main question is the values of the weights. If the weights are the same for every (w, c) pair, we can use standard SVD approximating initial matrix with respect to Frobenius norm. The following theorem shows that weights could be different for different pairs (w, c):

Theorem 1. *Assume* $< w, c > \approx PMI_k(w,c)$. *Then:*

$$\ell \approx \sum_{w \in V_w} \sum_{c \in V_c} const(< w, c >) + \alpha_{wc} \left(< w, c > - PMI_k(w,c) \right)^2 \tag{7}$$

where $const(< w, c >)$ *doesn't depend of* $< w, c >$ *and:*

$$\alpha_{wc} = \frac{1}{|D|} \cdot \frac{(k\#(w)\#(c))^2}{\#(w,c)|D| - k\#(w)\#(c)} \tag{8}$$

Proof. Taylor series expansion for $\ell(w,c)$ in point $< w, c >= PMI_k(w,c)$:

$$\ell(w,c) = \log \sigma(PMI_k(w,c)) + k\#(w)\frac{\#(c)}{|D|} \log \sigma(-PMI_k(w,c)) +$$

$$+0 \cdot (< w, c > - PMI_k(w,c)) + \frac{\partial^2 \ell(w,c)}{\partial < w, c >^2} (< w, c > - PMI_k(w,c))^2 +$$

$$+o\left((< w, c > - PMI_k(w,c))^2 \right)$$

Here the first term is $const(< w, c >)$, the second term is equal to zero because it's Taylor series expansion in the extremum point, and the third term leads to quadratic loss. From this equation we almost get the statement of the theorem:

$$\ell \approx \sum_{w \in V_w} \sum_{c \in V_c} const(< w, c >) + \#(w,c) \frac{\partial^2 \ell(w,c)}{\partial < w, c >^2} \left(< w, c > - PMI_k(w,c) \right)^2 \tag{9}$$

The last step is to get $\frac{\partial^2 \ell(w,c)}{\partial < w,c >^2}$. Let $< w, c >= x$, then:

$$\frac{\partial \ell(w,c)}{\partial x} = \#(w,c)\sigma(-x) - k\frac{\#(w)\#(c)}{|D|}\sigma(x) \tag{10}$$

$$\frac{\partial^2 \ell(w,c)}{\partial x^2} = -\sigma(x)\sigma(-x)\left[\#(w,c) + k\frac{\#(w)\#(c)}{|D|}\right] \tag{11}$$

Substituting $PMI_k(w,c) = PMI(w,c) - \log k = \log\left(\frac{\#(w,c)\cdot|D|}{\#(w)\#(c)}\right) - \log k =$
$= \log\left(\frac{\#(w,c)\cdot|D|}{k\#(w)\#(c)}\right)$ for $x =< w, c >$ we get:

$$\sigma\left(\log\left(\frac{\#(w,c)\cdot|D|}{k\#(w)\#(c)}\right)\right) = \left(1 - \frac{\#(w,c)\cdot|D|}{k\#(w)\#(c)}\right)^{-1}$$

$$\sigma\left(-\log\left(\frac{\#(w,c)\cdot|D|}{k\#(w)\#(c)}\right)\right) = \left(1 + \frac{\#(w,c)\cdot|D|}{k\#(w)\#(c)}\right)^{-1}$$

$$\alpha_{wc} = \#(w,c)\frac{\partial^2 \ell(w,c)}{\partial x^2} =$$

$$= \frac{1}{|D|}\cdot\frac{(k\#(w)\#(c))^2}{\#(w,c)|D| - k\#(w)\#(c)}$$

□

4 Fitting parameters

The comparison of word2vec and SVD was presented in [4]. However SVD is a matrix factorization optimizing quadratic loss (with the same weights of terms). Also these results are based on classic SVD calculation method. In this section we propose the iterative matrix factorization techniques frequently used in recommender systems. This method makes parameters fitting process closer to original word2vec parameters fitting.

In both suggested methods we consider the following optimization task:

$$\tilde{\ell} = \sum_{w\in V_w}\sum_{c\in V_c}\alpha_{wc}\left(< w,c > - PMI_k(w,c)\right)^2 \to \min_{w,c} \tag{12}$$

4.1 Stochastic Gradient Decent (SGD)

The derivatives of objective functions are as follows:

$$\frac{\partial \tilde{\ell}}{\partial w} = 2\sum_{c\in V_c}\alpha_{wc}\left(< w,c > - PMI_k(w,c)\right)c \tag{13}$$

$$\frac{\partial \tilde{\ell}}{\partial c} = 2 \sum_{w \in V_w} \alpha_{wc} \left(< w, c > - PMI_k(w, c) \right) w \tag{14}$$

In stochastic gradient decent we choose random terms from sums over $w \in V_w$ and $c \in V_c$:

$$w^{k+1} = w^k - \gamma_k \alpha_{wc} \left(< w^k, c > \right) c \tag{15}$$

$$c^{k+1} = c^k - \eta_k \alpha_{wc} \left(< w, c^k > \right) w \tag{16}$$

Here γ_k and η_k are step sizes. A simple way γ_k and η_k to define is just to use small constant values. Another variant is to use popular heuristics for SGD step size.

4.2 Alternating Least Squares (ALS)

The main problem of matrix factorizations via SGD is a low convergence rate. Sometimes this problem could be solved with Alternating Least Squares method. The idea is to get iteratively w as a solution of the equation $\frac{\partial \tilde{\ell}}{\partial w} = 0$ and c as a solution of the equation $\frac{\partial \tilde{\ell}}{\partial c} = 0$. From the expression for objective function gradient one can conclude that to use ALS in our task we just need to solve following linear systems iteratively up to convergence:

$$\left(\sum_{c \in V_c} \alpha_{wc} c c^T \right) w = \sum_{c \in V_c} \alpha_{wc} c \tag{17}$$

$$\left(\sum_{w \in V_w} \alpha_{wc} w w^T \right) c = \sum_{w \in V_w} \alpha_{wc} w \tag{18}$$

5 Conclusion and future research

Theorem 1 from section 3 shows that matrix factorization with weighted quadratic loss is close to initial optimization task. Also we get the weights and see that previous experiments with quadratic loss without weights are less motivated than experiments with weighted one.

In [3] authors have also mentioned that skip-gram with negative sampling (SGNS) remains superior to SVD on analogy questions and this could stem from the weighted nature of SGNS's factorization. We suppose that experiments with weighted quadratic loss could improve matrix factorization results in this task up to word2vec results. Also the objective function cold be modified with penalties of baseline predictors as it's common in collaborative filtering and it could slightly improve results too. Future work includes such experiments with analogy questions task and objective function modification.

References

1. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, United States, 3111–3119 (2013)
2. Levy, O., Goldberg, Y.: word2vec explained: deriving Mikolov et al.s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)
3. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada, 2177–2185 (2014)
4. Levy, O., Goldberg Y., Dagan, I.: Improving Distributional Similarity with Lessons Learned from Word Embeddings, Transactions of the Association for Computational Linguistics, 3, 211-225 (2015)