Interpreting Word Embeddings with Generalized Low Rank Models

Abstract

Glove and Skip-gram word embedding methods learn word vectors by decomposing a matrix of word co-occurrences into a product of low-rank matrices. In this work, we propose an iterative algorithm for computing word embeddings based on constructing word occurrence matrices via Generalized Low Rank Models. Our algorithm generalizes both Skip-gram and GloVe as well as giving rise to other embedding methods based on the specified occurrence matrix distribution and the number of iterations in the iterative algorithm. In particular, using a Tweedie distribution with one-step in the iterative framework results in the GloVe embedding whereas a multinomial distribution with full-convergence mode results in the skip-gram model. Experimental results demonstrate improved results of our model over the GloVe method on a word analogy similarity task.

1 Introduction

Word embeddings are low dimensional vector representations of words or phrases. They are applied to word analogy tasks and used as feature vectors in numerous tasks within natural language processing, computational linguistics, and machine learning. They are constructed by various methods which rely on the distributional hypothesis popularized by Firth: "words are characterized by the company they keep" [Firth, 1962]. Two seminal methodological approaches to finding word embeddings are Skip-gram [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014]. Both methods input a corpus $\mathcal{D}$, process it into a word co-occurence matrix Y, then output word vectors with some dimension d.

Skip-gram processes a corpus with w words and c contexts into a co-occurence matrix $Y \in \mathbb{R}^{w \times c}$, where y_{ij} is the number of times word w_i is used in context c_j in the corpus. Next, Skip-gram [Pennington et al., 2014, Section 3.1] estimates

$$(\hat{U}, \hat{V}) = \arg \min_{U \in \mathbb{R}^{w \times d}, V \in \mathbb{R}^{c \times d}} \sum_{i=1}^{w} \sum_{j=1}^{c} y_{ij} \log \frac{\exp\left(\boldsymbol{u}_i^T \boldsymbol{v}_j\right)}{\sum_{k=1}^{c} \exp\left(\boldsymbol{u}_i^T \boldsymbol{v}_k\right)} \tag{1}$$

then defines the word vectors to be the rows of $\hat{U}$ and the context vectors are the rows of $\hat{V}$.

GloVe processes a corpus with w words into a co-occurence matrix $Y \in \mathbb{R}^{w \times w}$ where y_{ij} is the harmonic sum of the number of tokens between words w_i and w_j for each co-occurrences. That is,

$$y_{ij} = \sum_{p_1 < p_2, |p_1 - p_2| \leq l_c, \mathcal{D}(p_1) = w_i, \mathcal{D}(p_2) = w_j} \frac{1}{|p_1 - p_2|},$$

Preprint. Work in progress.

where $\mathcal{D}(p_1)$ is the p_1^{th} word in the corpus and l_c is the length of the *context window*. Next, GloVe estimates

$$(\hat{U}, \hat{V}, \hat{\boldsymbol{\psi}}, \hat{\boldsymbol{\omega}}) = \arg \min_{U,V \in \mathbb{R}^{w \times d}; \boldsymbol{\psi}, \boldsymbol{\omega} \in \mathbb{R}^w} \sum_{i,j=1}^{w} (\min\{y_{ij}, y_{\max}\})^{.75} \left(\boldsymbol{u}_i^T \boldsymbol{v}_j + \psi_i + \omega_j - \log y_{ij}\right)^2, \tag{2}$$

where $\boldsymbol{u}_i$ is the i^{th} row of the matrix U, $\boldsymbol{v}_j$ is the j^{th} row of the matrix V, ψ_i and ω_j are bias terms, and $y_{\max}$ is some prespecified cutoff. GloVe then defines the estimated word vectors to be the rows of $\frac{1}{2}\hat{U} + \frac{1}{2}\hat{V}$.

In both Skip-gram and GloVe, a matrix of co-occurences Y is introduced by processing the corpus, and an objective function is introduced to find a low rank factorization related to the co-occurences Y. In this paper, we derive the corpus processing techniques and loss functions from principled model-based assumptions. We introduce a model-based iterative algorithm, and show that (1) results from running the iterative algorithm until completion for a Multinomial model and (2) is a one step of the iterative algorithm for a Tweedie model. This algorithms allows us to introduce methods intimately related to Skip-gram and GloVe and to introduce altogether new methods.

2 Related Work

In the introduction, we saw that Skip-gram and GloVe compute a co-occurence matrix Y which results from processing the corpus $\mathcal{D}$ and an objective function $\mathcal{L}$ to relate the matrix Y to a product of low rank matrices. Many existing approaches for explaining word embedding methods do so by identifying or deriving the co-occurence matrix Y or the objective function $\mathcal{L}$. In this section, we review relevant work in this area, which helps frame our approach discussed in Section 4.

Much of the related work involves using the co-occurence matrix from Skip-gram, where the rows represent words, the columns represent context, and the entry of the matrix represents the frequency of the word-context pair: Define Y to be this matrix.

Early approaches to finding low-dimensional embeddings of words relied on the singular value decomposition [Landauer et al., 1998, Turney and Pantel, 2010]. These methods would truncate the singular value decomposition by zeroing out the small singular values. Eckart and Young [1936] show that this is equivalent to using a loss function $\mathcal{L}$ which is invariant to orthogonal transformation. For simplicity, these early approaches find

$$\arg \min_{U \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{c \times d}} \|UV^T - Y\|_F^2,$$

where the Frobenius norm is the choice of loss function.

The co-occurence matrix and the loss function for Skip-gram can be read off from equation (1). Cotterell et al. [2017] find a probabilistic interpretation of this loss function related to a Binomial distribution.

Mikolov et al. [2013b] introduce Skip-gram with negative sampling (SGNS), a variant of Skip-gram. If we view Skip-gram as maximizing the true positive rate of predicting a word will be in a given context, we can view SNGS as maximizing the true positive rate plus k times an approximation of the true negative rate. When $k = 0$, Skip-gram and SGNS coincide.

Levy and Goldberg [2014] use a heuristic argument to interpret SGNS as using a co-occurence matrix which is a shifted PMI. Define the total number of times word w_i appears $t_i = \sum_{\tilde{j}=1}^{c} y_{i\tilde{j}}$, the total number of times context c_j appears $t_j = \sum_{\tilde{i}=1}^{w} y_{\tilde{i}j}$, and the total number of words $t = \sum_{\tilde{i},\tilde{j}=1}^{w,c} y_{\tilde{i}\tilde{j}}$. The shifted PMI matrix has has entries $\log \frac{y_{ij}t}{t_i t_j} - \log k$. However, they were unable to determine the loss function. Later, Li et al. [2015] and Landgraf and Bellay [2017] were able to explicitly identify both the loss function and the co-occurence matrix. They find a different co-occurence matrix than Levy and Goldberg [2014], one that does not depend on k, while their loss function does depend on k. Surprisingly, they argue that SGNS is using Y, the same matrix that Skip-gram uses. The loss function is

$$\sum_{i,j=1}^{w,c} y_{ij}(\boldsymbol{u}_i^T \boldsymbol{v}_j) - (y_{ij} + k\frac{t_i t_j}{t}) \log\left(1 + \exp(\boldsymbol{u}_i^T \boldsymbol{v}_j)\right).$$

Landgraf and Bellay [2017] explain that this loss function has a probabilistic interpretation, and they use that interpretation to recover the shifted PMI matrix as a plug in estimator within their model.

The approach in this paper will be to view the co-occurence matrix as having each entry be a random variable and connect the loss function to the likelihood of the random variable. This approach was shown to be effective by Cotterell et al. [2017] and Landgraf and Bellay [2017]. Before this, we need to review some relevant modeling background.

3 Generalized Low Rank Models

Generalized linear models model a response vector y as estimated given an input matrix X and an assumed link-linear relationship with the mean of y given X. A brief introduction to generalized linear models is given in the Appendix in Section 7.2. We now consider two important generalizations of generalized linear models: (1) extending the response to a matrix $Y \in \mathbb{R}^{n \times m}$, and (2) an unsupervised low-rank relationship for the systematic component η.

Principal components analysis (PCA) Jolliffe [2011], is one well-known method for computing a low-rank approximation to a matrix Y which satisfies both (1) and (2). In PCA, we write $y_{ij} \sim \text{Normal}(\boldsymbol{u}_i^T \boldsymbol{v}_j, \sigma^2)$ where each y_{ij} is independent, and $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^d$ for some dimension $d \ll n, m$. The maximum likelihood estimator for $\boldsymbol{u}_i$ is taken to be a low-dimensional embedding of the i^{th} row of the response Y. A similar extension from linear models to generalized linear models allows us to generalize from PCA to generalized low rank models [Udell et al., 2014] allowing us to estimate model-based low-dimensional embeddings of non-normal data.

Definition 1 *For some exponential dispersion family* $\text{ED}(\mu, \varphi)$ *with mean parameter* μ *and dispersion parameter* φ*, the model for* $Y \in \mathbb{R}^{n \times m}$ *is a* generalized low rank model *with* link function g *when*

$$\begin{cases} y_{ij} \overset{\text{ind}}{\sim} \text{ED}(\mu_{ij}, \varphi) \\ g(\mu_{ij}) = \eta_{ij}, \\ \eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j + \psi_i + \omega_j, \end{cases}$$

where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^d$ *are the rows of matrices* $U \in \mathbb{R}^{n \times d}$ *and* $V \in \mathbb{R}^{m \times d}$*, respectively, and* $\boldsymbol{\psi}, \boldsymbol{\omega} \in \mathbb{R}^d$ *are offset or bias terms.*

The difference between the generalized low rank model, and the generalized linear model from Definition (3) is in the systematic component (third line of Definition (4)). In generalized low rank models, the link-transformed mean is modeled as a low rank matrix which is the product of two matrices with small number of columns d. In other words, generalized low rank models find low-dimensional embeddings of the mean of the response on a scale determined by the link function.

In Section 4 we present an iterative algorithm inspired by IRLS which to estimate the maximum likelihood estimator of a generalized low rank model. In Section 5, we show that this method allows us to compute word embeddings.

4 Methodology

In Section 7.2 of the Appendix following our introduction of generalized linear models, we introduced the IRLS method for computing the MLE $\hat{\boldsymbol{\beta}}$, and in Section 3 we introduced generalized low rank models. We now present Algorithm 1 for performing an algorithm analogous to IRLS on a co-occurrence matrix Y to produce embedding matrices $\hat{U}$ and $\hat{V}$. This algorithm gives us an effective procedure for constructing word embeddings $\hat{U}$ and $\hat{V}$ from a corpus D.

4.1 Our Proposed Method

Our proposed method has three steps:

1. Choose a co-occurence matrix $Y \in \mathbb{R}^{w \times c}$ to summarize the document. (Note, in some cases that $c = w$.)

2. Choose a plausible exponential dispersion family to model the entries of the co-occurence matrix

3. Run IRLS using the chosen exponential dispersion family to output word vectors

We will in practice always make a default choice of the link function that only depends on the distribution. This makes it so that there is no need to input the link function into the algorithm.

Data: Co-occurrence Matrix $Y \in \mathbb{R}^{w \times c}$
Require: Distribution $ED(x; \mu, \varphi)$, number of steps T, Dimension d
Result: $\hat{U} \in \mathbb{R}^{w \times d}, \hat{V} \in \mathbb{R}^{c \times d}, \hat{\boldsymbol{\psi}} \in \mathbb{R}^{w}, \hat{\boldsymbol{\omega}} \in \mathbb{R}^{c}$
Initialize $\mu^{(0)} = Y$;
Compute $W^{(0)}$ and $Z^{(0)}$ according to Equations (8) and (9);
for $t = 0 : T$ **do**
    Evaluate the least squares problem

$$\arg \min_{U \in \mathbb{R}^{w \times d}, V \in \mathbb{R}^{c \times d}, \boldsymbol{\psi} \in \mathbb{R}^{w}, \boldsymbol{\omega} \in \mathbb{R}^{w}} \sum_{i,j=1}^{w,c} W_{ij}^{(t)} \left(\boldsymbol{u}_i^T \boldsymbol{v}_j + \omega_i + \psi_j - z_{ij}^{(t)} \right)^2 ;$$

    Update $\mu^{(t+1)}$ according to $\mu_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j + \omega_i + \psi_j$;
    Update $W^{(t+1)}$ according to Equation (8);
    Update $Z^{(t+1)}$ according to Equation (9);
end
return $\hat{U}^{(T)}, \hat{V}^{(T)}, \hat{\boldsymbol{\psi}}^{(T)}, \hat{\boldsymbol{\omega}}^{(T)}$

Algorithm 1: IRLS Algorithm for Generalized Low Rank Models

We regularize the word vectors by including the penalty

$$\frac{\lambda}{2} \left(\|U\|_F^2 + \|V\|_F^2 \right) \tag{3}$$

with $\lambda = .002$ for two reasons. One is to reduce some noise in the estimation of the word vectors. Udell et al. [2014, Lemma 7.3] show that penalizing by (3) is equivalent to penalizing by $\lambda \|UV^T\|_*$, the nuclear norm of UV^T. Since penalizing the nuclear norm UV^T shrinks the dimension of the embedding and larger dimensional embeddings tend to be better [Melamud et al., 2016], we choose a small tuning parameter to reduce noise while still preserving the dimension.

Another reason is to symmetrically distribute the singular values of $\hat{U}\hat{V}^T$ to both matrices $\hat{U}$ and $\hat{V}$. Write the singular value decomposition $\hat{U}\hat{V}^T = \overline{U}\Sigma\overline{V}^T$, for $\overline{U}$ and $\overline{V}$ orthogonal and Σ diagonal. Mu et al. [2018, Theorem 1] shows that using penalty (3) results in having $\hat{U} = \overline{U}\Sigma^{1/2}Q$ and $\hat{V} = \overline{V}\Sigma^{1/2}Q$ for some orthogonal matrix Q. This is desirable since it was argued by Levy et al. [2015] that it has empirically been shown that a symmetric distribution of singular values works optimally on semantic tasks.

4.2 Examples

4.2.1 First example: GloVe

The first step of our algorithm is to pick a co-occurence matrix $Y \in \mathbb{R}^{w \times w}$ that summarizes the corpus. We will pick the one used by GloVe, so that $y_{ij} = \sum_{p_1 < p_2, |p_1 - p_2| \leq l_c, \mathcal{D}(p_1) = w_i, \mathcal{D}(p_2) = w_j} \frac{1}{|p_1 - p_2|}$.

Now we must determine a plausible distribution for the co-occurence matrix that lies within the exponential dispersion family. Recall that the Tweedie distribution has the property mentioned in equation (5) that it is a sum of Poisson many independent Gamma distributions. An informal way to

write this is that

$$\begin{aligned}\text{Tweedie} &= \sum_{i=1}^{\text{Poisson}} \text{Gamma}_i \\ &= \sum_{i=1}^{\text{Poisson}} \frac{1}{\text{InvGamma}_i},\end{aligned}$$

where the InvGamma distribution is defined to be the reciprocal of the Gamma distribution. The InvGamma distribution is supposed on the positive real line.

We will argue that the Tweedie distribution is a reasonable distribution for the entries of the co-occurence matrix. We will do this by connecting the Poisson and Gamma distributions in displayed above to attributes of the corpus. It seems intuitively reasonable that the number of times word w_i and word w_j co-occur within the corpus could be modeled as having a Poisson distribution. It's also possible that the number of tokens between word w_i and word w_j at some co-occurence is appromximately InvGamma for some choice of rate and scale parameters. Note there's an approximation being made here where the number of tokens is an integer but the InvGamma isn't necessarily.

Now we need to find the weight W and the pseudo-response Z that the Tweedie distribution provides. This only amounts to plugging in the cumulant generating function b which we found in 7.1. When we do this, we find that

$$\begin{cases} W_{ij} = \mu_{ij}^{2-p}, \\ z_{ij} = \frac{y_{ij}-\mu_{ij}}{2\mu_{ij}} + \log \mu_{ij}. \end{cases}$$

When we initialize the algorithm with $\hat{\mu}^{(0)} = Y$, the pseudo response simplified to $z_{ij} = \log y_{ij}$. When you further choose $p = 1.25$, the weight simplifies to $y_{ij}^{3/4}$. In summary, we've shown that:

Result 1 *Inputting the harmonic co-occurence matrix used by GloVe, Tweedie distribution with power p = 1.25, number of iterations $T = 1$ into our algorithm results in GloVe (without the weights being truncated.)*

Given this connection, we can extend GloVe by either early stopping the algorithm some time after the first iteration or running the algorithm either until completion.

4.3 Second example: SVD, Skip-gram, and more

We will pick a new co-occurence matrix to summarize the corpus. In this case, we will pick the co-occurence matrix $Y \in \mathbb{R}^{w \times c}$ so that Y_{ij} is the number of times word w_i appears in context c_j.

4.3.1 The SVD

For step 2 of the algorithm, we could propose that the entries of Y follow a Gaussian distribution. For step 3, the algorithm will always converge in one iteration, so our method recovers the method of computing a truncated SVD by Eckart and Young [1936].

4.3.2 Skip-gram

For step 2 of the algorithm, we could propose that the entries of Y follow a multinomial distribution. Specifically, we could propose that the the row of Y corresponding to the i^{th} word has the distribution

$$\boldsymbol{y}_i \sim \text{Multinomial}\left(\sum_{j=1}^{c} y_{ij}, \boldsymbol{\pi}\right),$$

where $\boldsymbol{\pi} \in \mathbb{R}^c$ is vector of probabilities of appearing within each context and $\sum_{i=1}^{w} y_{ij}$ is the total number of times word w_i appears in the corpus. This was the approach taken by Cotterell et al. [2017].

It would be interesting to propose the weight and pseudo-response here so that we can see what the form of the one-step estimator would be. We suspect that it could recover the estimator of Arora et al. [2016], since they find their estimator via the second order Taylor expansion of the likelihood, completely analogous to Newton's method, which is the basis for IRLS. One iteration of algorithm would result in an estimator which is equivalent to doing a second-order Taylor expansion around the co-occurence matrix Y.

4.3.3 New estimator

In the previous section, there was a problem with the model description: the co-occurence matrix Y had entries appearing on both the left-hand and right-hand side! This problem appeared since it is unknown to us what the total number of times word w_i will appear in the corpus. This means that the Poisson distribution is a more appropriate model for the co-occurence matrix Agresti [2014, Section 7.2.1].

For step 2, we could instead propose a Poisson distribution to model the entries of the co-occurence matrix. It seems likely to the authors that this would lead to improvements.

5 Experiments

In Section 4 we introduced an IRLS algorithm to compute word embeddings such as those produced by GloVe. We now conduct quantitative evaluation experiments on an English word analogy task, a variety of word similarity tasks [Mikolov et al., 2013a] to demonstrate the performance of the algorithm. First, in Section 5.1 we introduce the analogy similarity task for evaluating word vectors. In Section 5.2 our parameter configurations and training procedure. In Sections 5.3-5.6 we present results of our algorithm in numerous scenarios showcasing improvement through multiple steps and robustness to other model parameters.

5.1 Word Analogies

We introduce the word analogy task following the presentation of [Pennington et al., 2014]. The word analogy task is a dataset of $19,544$ statements of the basic form "a is to b as c is to __", which are divided into a semantic and syntactic subsets. The semantic statements are typically analogies relating to people, places, or nouns such as "Athens is to Greece as Berlin is to __", while the syntactic questions relate to verb or adjective forms such as "dance is to dancing as fly is to __". The basic analogy statement is answered by finding the closest vector $\boldsymbol{u}_d$ to $\boldsymbol{u}_b - \boldsymbol{u}_a + \boldsymbol{u}_c$[1] in the embedding space via cosine similarity[2]. The task has been shown under specific assumptions to be provably solvable by methods such as GloVe and Skip-gram [Ethayarajh et al., 2018, Gittens et al., 2017] and as such is closely related to solving the objectives introduced in Sections 1 and 4.

5.2 Training Details

We train our models on the English text8 corpus[3] with approximately 17 million tokens. We filter out word types that occur fewer than 50 times to obtain a vocabulary size of approximately $11,000$; a ratio consistent with other embedding literature experiments[4].

The adjustable model configurations in our algorithm are the choice of power parameter p, penalty tuning parameter λ, and co-occurrence processing step. We experiment with different choices of $p \in \{1.1, 1.25, 1.5, 1.75, 1.9\}$, different choices of processing including no processing, clamping the weights (as in GloVe) and truncating the outliers in the co-occurrence matrix (elaborated on in Section 5.5, and set the penalty tuning parameter $\lambda = 0.002$. The estimated word vectors are the rows of $\frac{1}{2}\hat{U} + \frac{1}{2}\hat{V}$.

[1] When evaluating analogies, the search space for d excludes any of a, b, or c.

[2] Many have considered other forms of distance such as Euclidean distance, or other forms of evaluation such as multiplicative evaluation

[3] `http://mattmahoney.net/dc/text8.zi`

[4] By truncating the vocabulary size to $11,000$ we note that we are unable to solve all $19,544$ analogies. We are able to solve roughly one-third of the analogies, and present results on this subset.

Step	Semantic	Syntactic	Total
1	70.6	45.51	50.28
2	71.94	45.97	50.9
3	72.38	46.47	51.39
4	72.29	46.51	51.41
5	**72.49**	**46.57**	**51.47**

Table 1: Results of multiple iterations of the IRLS algorithm with weight trunctation of $y_{max} = 10$, $p = 1.25$, and initialization of $\mu = y$. The best performing word vectors on the analogy similarity task are those from the 5th iteration.

For all of our experiments, we set the dimension of the word vectors to $d = 150$, and the objective function at each iteration is optimized using Adagrad [Duchi et al., 2011] with a fixed learning rate of 0.1[5]. Models are trained for up to 50 epochs (50 passes through the co-occurrence matrix) with batches of size 512.

5.3 Experiment 1: Multiple Iteration Effects

We present results of multiple steps of our IRLS algorithm compared to the GloVe method. In particular, to match the GloVe objective function and processing, we set the weight function in our model from $f(y) = y^{2-p}$ to $f(y) = \left(\frac{y}{y_{max}}\right)^{2-p}$ with $y_{max} = 10$ and $p = 1.25$, and note when initializing $\mu = y$, the solution to 2 is exactly the solution of the IRLS algorithm at the first step. Results of the IRLS algorithm with these configurations is shown in Table 1.

We remark on a few observations based on the results in Table 1. First, as the number of steps increases, the accuracy on the analogy task increases as well with the models at steps 3-5 doing relatively similarly and over 1% higher than the GloVe model. Second, relatively few steps are needed with the accuracy being relatively similar for steps 3-5 in comparison to steps 1 and 2. Finally, we note that in this case running the algorithm for only 2 iteration is sufficient to see an improvement over the one-step estimates.

5.4 Experiment 2: Effects of Varying p

Here we examine the effect of the choice of the power p in the tuning parameter when you run a Tweedie generalized low rank model.

p	Iterations	Semantic	Syntactic	Total
1.25	1	69.89	44.32	49.18
1.25	2	**72.91**	45.41	50.63
1.5	1	72.2	45.7	50.73
1.5	2	72.2	**45.72**	**50.75**
1.75	1	60.3	43.1	46.36
1.75	1	64.83	42.49	46.74
1.9	1	50.8	40.18	42.4
1.9	2	60.39	39.43	43.41

Table 2: Results for multiple choices of p for one and two iterations.

The Results in Table 2 show that values of p which are high perform poorly, while values of p around 1.25 and 1.5 perform similarly. We find that $p = 1.5$ performs the best, and view this value of p as a good choice as it accounts for zero-inflation present in Y.

5.5 Experiment 3: Effects of Co-Occurrence Matrix Transformations

With power $p = 1.25$, we explore the effect of different strategies to handle large entries in the co-occurence matrix Y. One strategy is to simply input Y into step 3 of our algorithm. A second

[5]This training procedure is slightly different from the asynchronous stochastic gradient descent training procedure used in [Pennington et al., 2014].

strategy is to truncate the weight W that results from step 3 of our algorithm, in a strategy similar to GloVe. A third strategy is to input $\min\{Y, Y_{\max}\}$ into step 3 of our algorithm, where for us $Y_{\max} = \frac{100}{10} = 10$, where 100 is the threshold chosen by GloVe Pennington et al. [2014] and our corpus is $\frac{1}{10}^{\text{th}}$ of the size of their corpus.

Strategy	Iterations	Semantic	Syntactic	Total
1	1	69.89	44.32	49.18
1	2	72.91	45.41	50.63
2	1	73.18	47.87	52.67
2	2	75.22	47.97	53.15
3	1	55.95	44.47	46.65
3	1	55.95	45.66	47.61

Table 3: Results for multiple choices of regularizing the large values of the co-occurence matrix

We see that the clear best strategy is to truncate the weights at each step in the iterative algorithm, in a manner similar to GloVe.

5.6 Experiments: Regularization Effects

Strategy	Iterations	Semantic	Syntactic	Total
Penalty	1	69.89	44.32	49.18
Penalty	2	72.91	**45.41**	**50.63**
No Penalty	1	72.91	42.26	48.09
No Penalty	1	**P73.62**	43.26	49.03

Table 4: Results for including the penalty term in Equation (3) and not including the diagonal terms.

Finally, we experiment with whether the penalty introduced in Equation (3) improves results and accurately reduces noise in the estimate. We also consider not including the diagonal elements of Y as a form of regularization and experiment here as well, as these terms are often large (can be considered as outliers) and do not contain a great deal of information. Table 4 demonstrates the included regularization improves results.

6 Conclusion

We presented a general methodology for computing word vectors from a corpus, and demonstrate that many commonly used methods are special cases this methodology. Our methodology allows us to analyze the distributional properties of the co-occurrence matrices and yields better objective functions for estimating word functions more particular to the characteristics of the corpora. Experimental results on a small corpus demonstrates our method improves results on a word analogy similarity task. It is our hope that this methodology can lead to the development of better, statistically sound word embeddings, and improve results on many other downstream tasks.

References

Alan Agresti. *Foundations of Linear and Generalized Linear Models*. Wiley, 2014.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A Latent Variable Model Approach to PMI-based Word Embeddings. *Tacl*, 4:385–399, 2016. ISSN 2307-387X. URL `http://www.aclweb.org/anthology/Q/Q16/Q16-1028.pdf`.

Ryan Cotterell, Adam Poliak, B. Van Durme, and Jason Eisner. Explaining and generalizing skip-gram through exponential family principal component analysis. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, volume 2, 2017. ISBN 9781510838604. URL `https://github.`

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*, 2018.

JR Firth. A synopsis of linguistic theory. In *Studies in Linguistic Analysis*, pages 1–32. 1962. ISBN 9004102655.

RA Fisher. On the Mathematical Foundations of Theoretical Statistics. *Phil. Trans. R. Soc. Lond. A*, 222:309–368, 1922.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. Skip-gram-zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 69–76, 2017.

Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

Bent Jorgensen. Exponential Dispersion Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 49(2):127–162, 1987. ISSN 1369-7412. doi: 10.2307/2345415.

Bent Jorgensen. *The Theory of Dispersion models*. Chapman and Hall, 1997.

Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998. ISSN 0163-853X. doi: 10.1080/01638539809545028.

Andrew J Landgraf and Jeremy Bellay. word2vec skip-gram with negative sampling is a weighted logistic pca. *arXiv preprint arXiv:1705.09755*, 2017.

Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. *Advances in Neural Information Processing Systems (NIPS)*, pages 2177–2185, 2014. ISSN 10495258. doi: 10.1162/153244303322533223. URL `https://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf`.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *IJCAI*, pages 3650–3656, 2015.

Peter McCullagh and John A Nelder. *Generalized Linear Models*, volume 37. CRC press, 1989.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. The Role of Context Types and Dimensionality in Learning Word Embeddings. 2016.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12, 2013a. ISSN 15324435. doi: 10.1162/153244303322533223. URL http://arxiv.org/pdf/1301.3781v3.pdf.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. 2013b. ISSN 10495258. doi: 10.1162/jmlr.2003.3.4-5.951.

Cun Mu, Guang Yang, and Zheng Yan. Revisiting Skip-Gram Negative Sampling Model with Regularization. *arXiv preprint arXiv:1804.00306*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. ISBN 9781937284961. doi: 10.3115/v1/D14-1162. URL http://aclweb.org/anthology/D14-1162.

Gordon K Smyth. Regression modelling of quantity data with exact zeroes. In *Proceedings of the Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management. Technology Management Centre, University of Queensland*, pages 572–580. 1996. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.1646&rep=rep1&type=pdf.

Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

Madeleine Udell, Corinne Horn, Reza Zadeh, and Stephen Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2014. ISSN 1935-8237. doi: 10.1561/2200000055.

7 Appendix

In this section, we develop some background in exponential dispersion families [Jorgensen, 1997], generalized linear models [Agresti, 2014], and generalized low rank models [Udell et al., 2014].

7.1 Exponential dispersion families and the Tweedie distribution

We begin by discussing exponential dispersion families, the distribution of the response in generalized linear models.

Definition 2 *Let $y \in \mathbb{R}$ be a random variable. If the density function $f(y; \theta, \phi)$ of y satisfies*

$$\log f(y; \theta, \varphi) = \frac{y\,\theta - b(\theta)}{a(\varphi)} + c(y; \phi)$$

over its support, then the distribution of y is in the exponential dispersion family*. The parameter θ is the* natural parameter*, φ is the* dispersion parameter*, and the function b is the* cumulant generating function*.*

In many cases, the function $a(\varphi)$ is very simple, meaning that, for instance, $a(\varphi) = 1$ or $a(\varphi) = \varphi$. The function $c(y; \varphi)$ can be viewed as the normalizing constant ensuring that the density integrates to one. When y follows a distribution in the exponential dispersion family with natural parameter θ, its mean $\mu = b'(\theta)$, so we can equivalently specify the mean μ or the natural parameter θ.

Many classical distributions such as the Poisson, Normal, Binomial, and Gamma distribution are exponential dispersion families. For example, when $y \sim \mathrm{Normal}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, its log density satisfies

$$\log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{-1}{2} \left(\frac{y-\mu}{\sigma} \right)^2 \right] \right\} = \frac{y\mu - \frac{1}{2}\mu^2}{\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2) - \frac{y^2}{2\sigma^2}, \tag{4}$$

showing that here the natural parameter $\theta = \mu$, the dispersion parameter $\varphi = \sigma^2$, the functions $b(\theta) = \frac{1}{2}\theta^2$, $a(\varphi) = \varphi$, and $c(y; \varphi) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{y^2}{2\sigma^2}$.

The Tweedie distribution [Jorgensen, 1997], of particular importantance to us, also lies within the exponential dispersion family. Instead of defining the Tweedie distribution through the form of its density, we will define it through the relationship between its mean and variance. This relies on a result from [Jorgensen, 1987, Theorem 1] that distributions within the exponential dispersion family are defined by the relationship between their mean and variance.

Definition 3 *A random variable y has a Tweedie distribution with power parameter $p \in \{0\} \cup [1, \infty)$ when*

$$\mathrm{var}(y) = \varphi\,(\mathbb{E}[y])^p$$

and the distribution of y is an exponential dispersion family. In this case, we write $y \sim \mathrm{Tweedie}_p(\mu, \varphi)$, where $\mu = \mathbb{E}(y)$ is the mean.

The Normal distribution discussed above has a variance that does not depend on the mean. In our new notation, this means that the Normal distribution is a Tweedie distribution with power parameter $p = 0$. The Poisson distribution has variance equal to the mean and is in the exponential dispersion family, so is a Tweedie distribution with power parameter $p = 1$ and dispersion parameter $\varphi = 1$. A Gamma distribution with shape parameter α and rate parameter β is a Tweedie distribution with power $p = 2$, mean $\mu = \frac{\alpha}{\beta}$, and dispersion parameter $\varphi = \alpha^{-1}$.

We will only consider Tweedie distributions with power parameter $p \in (1, 2)$. These distributions are also known as compound Poisson-Gamma distributions due to the representation

$$\mathrm{Tweedie}_p(\mu, \varphi) = \sum_{i=1}^{n} g_i, \tag{5}$$

where $n \sim \mathrm{Poisson}(\lambda)$ and $g_i \overset{\text{i.i.d.}}{\sim} \mathrm{Gamma}(\alpha, \beta)$, and $\lambda = \frac{\mu^{2-p}}{(2-p)\varphi}$, $\alpha = \frac{2-p}{p-1}$, and $\beta = \frac{\mu^{1-p}}{(p-1)\varphi}$ [Smyth, 1996]. It is important to note that the Tweedie distribution has positive mass at zero, an

important characteristic for capturing the zero-inflation prominent in some co-occurence matrices. Specifically,

$$\mathbb{P}[y = 0] = \exp\left(\frac{-\mu^{2-p}}{\varphi(2-p)}\right) > 0.$$

Using other arguments related to representations of the mean and variance in terms of the cumulant generating function b, Jorgensen [1997] show that the Tweedie distribution has $b(\theta) = \frac{1}{2-p}\left((1-p)\theta\right)^{\frac{2-p}{1-p}}$.

7.2 Generalized linear models, the MLE, and iteratively re-weighted least squares

We start by introducing the linear model. Given a response $\boldsymbol{y} \in \mathbb{R}^n$ comprising n observations, the model for $\boldsymbol{y}$ is a linear model with covariates $\boldsymbol{x}_i \in \mathbb{R}^p$ when $y_i \overset{\text{ind.}}{\sim} \text{Normal}(\boldsymbol{x}_i^T\boldsymbol{\beta}, \sigma^2)$ for all $i \in \{1, \ldots, n\}$. In vector notation, this reads that $\boldsymbol{y} \sim \text{Normal}(X\boldsymbol{\beta}, \sigma^2 I)$, where $X \in \mathbb{R}^{n\times p}$ is a matrix with i^{th} row $\boldsymbol{x}_i$.

Generalized linear models remove the the assumptions of normality and that the mean is a linear function of the coefficients $\boldsymbol{\beta}$.

Definition 4 *For some exponential dispersion family* $\text{ED}(\mu, \varphi)$ *with mean parameter* μ *and dispersion parameter* φ*, the model for* $\boldsymbol{y} \in \mathbb{R}^n$ *is a generalized linear model with* link function g *when*

$$\begin{cases} y_i \overset{\text{ind.}}{\sim} \text{ED}(\mu_i, \varphi) \\ g(\mu_i) = \eta_i \\ \eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}. \end{cases}$$

In the first line of the displayed relationships, the distribution of the response $\boldsymbol{y}$ is described. In the third line, the *systematic component* η_i expresses the effect of the covariates $\boldsymbol{x}_i$. The second line connects the distribution to the covariates through the link function. That is, the covariates effect a link-transformed mean.

Generalized linear models are used as the default modeling framework in many fields of applied science for non-normal distributions [McCullagh and Nelder, 1989]. When $g(\mu) = \mu$ is the identity map and ED is the Normal distribution, the generalized linear model is simply the linear model. When $g(\mu) = \text{logit}(\mu) = \log\frac{1-\mu}{\mu}$ and ED is the Binomial distribution, the generalized linear model is logistic regression. Further, a generalized linear model can be viewed as a no-hidden-layer neural network with activation function g.

The coefficient $\boldsymbol{\beta}$ is unknown and a target of estimation in the generalized linear model. The standard approach to compute $\boldsymbol{\beta}$ in a generalized linear model is *maximum likelihood estimation* [Fisher, 1922, Section 7] to produce the *maximum likelihood estimator*, or MLE $\hat{\boldsymbol{\beta}}$, with $\hat{\boldsymbol{\beta}}$ defined to be the maximizer of the likelihood of $\boldsymbol{y}$ given X. Define $\ell(\boldsymbol{\beta})$ to be the negative log likelihood. A natural way to estimate the MLE is through Fisher scoring, a version of Newton's method on the log likelihood which uses the expectation of the Hessian in place of the Hessian [Agresti, 2014, Section 4.5]. In notation, Fisher scoring produces a sequence of estimates $\{\hat{\boldsymbol{\beta}}^{(t)}\}_{t=1}^{\infty}$ starting with some initialization $\hat{\boldsymbol{\beta}}^{(0)}$ so that

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} + \left(\mathbb{E}\left[D^2\ell(\boldsymbol{\beta})\right]\Big|_{\hat{\boldsymbol{\beta}}^{(t)}}\right)^{-1}\nabla\ell(\hat{\boldsymbol{\beta}}^{(t)}), \tag{6}$$

where $\nabla\ell$ is the gradient and $D^2\ell$ is the Hessian of the likelihood ℓ. Upon plugging in the gradient and expected hessian of ℓ for an exponential family, a surprising identity emerges: each iteration of Fisher scoring is equivalent to minimizing a weighted least squares objective:

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\|\left(W^{(t)}\right)^{1/2}\left(X\boldsymbol{\beta} - \boldsymbol{z}^{(t)}\right)\right\|_2^2, \tag{7}$$

where the weight $W^{(t)}$ is diagonal with

$$W_{ii}^{(t)} = \left[\left(g' \left(\mu_i^{(t)} \right) \right)^2 b'' \left((b')^{-1} \left(\mu_i^{(t)} \right) \right) \right]^{-1} \tag{8}$$

and the pseudo-response for observation i is

$$z_i^{(t)} = \eta_i^{(t)} + g' \left(\mu_i^{(t)} \right) \left(y_i - \mu_i^{(t)} \right), \tag{9}$$

where $\boldsymbol{\eta}^{(t)} = X\hat{\boldsymbol{\beta}}^{(t)}$ and $\mu_i^{(t)} = g^{-1}(\eta_i^{(t)})$.

7.3 Generalized low rank models