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 \( D \), process it into a word co-occurrence matrix \( Y \), then output word vectors with some dimension \( d \).

Skip-gram processes a corpus with \( w \) words and \( c \) contexts into a co-occurrence 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

\[
(U, V) = \arg \min_{U \in \mathbb{R}^{w \times d}, V \in \mathbb{R}^{c \times d}} \sum_{i=1}^{w} \sum_{j=1}^{c} y_{ij} \log \frac{\exp(u_i^T v_j)}{\sum_{k=1}^{d} \exp(u_i^T v_k)}
\]

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-occurrence 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,

\[
y_{ij} = \sum_{p_1 < p_2, \|p_1 - p_2\| \leq l, D(p_1) = w_i, D(p_2) = w_j} \frac{1}{|p_1 - p_2|},
\]
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

$$
(\hat{U}, \hat{V}, \psi, \omega) = \arg \min_{U, V \in \mathbb{R}^{d \times d}, \psi, \omega \in \mathbb{R}^{w}} \sum_{i,j=1}^{w} \left( \min\{y_{ij}, y_{\max}\} \right) \frac{75}{2} \left( u_{i}^{T} v_{j} + \psi_{i} + \omega_{j} - \log y_{ij} \right)^{2},
$$

(2)

where $u_{i}$ is the $i^{th}$ row of the matrix $U$, $v_{j}$ is the $j^{th}$ row of the matrix $V$, $\psi_{i}$ and $\omega_{j}$ 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-occurrences $Y$ is introduced by processing the corpus, and an objective function is introduced to find a low rank factorization related to the co-occurrences $Y$. 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-occurrence 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-occurrence 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-occurrence 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

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

where the Frobenius norm is the choice of loss function.

The co-occurrence 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 Multinomial 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-occurrence matrix which is a shifted PMI. Define the total number of times word $w_{i}$ appears $t_{i} = \sum_{j=1}^{c} y_{ij}$, the total number of times context $c_{j}$ appears $t_{j} = \sum_{i=1}^{w} y_{ij}$, and the total number of words $t = \sum_{i,j=1}^{w,c} y_{ij}$. The shifted PMI matrix has has entries $\log \frac{y_{ij}^{t_{ij}}}{t_{i} t_{j}} - \log k$. 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-occurrence matrix. They find a different co-occurrence 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

$$
\sum_{i,j=1}^{w,c} y_{ij} (u_{i}^{T} v_{j}) - (y_{ij} + k \frac{t_{i} t_{j}}{t}) \log \left( 1 + \exp(u_{i}^{T} v_{j}) \right).
$$
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-occurrence 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 \( \eta \).

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 \( y_{ij} \sim \text{Normal}(u_i^T v_j, \sigma^2) \) where each \( y_{ij} \) is independent, and \( u_i, v_j \in \mathbb{R}^d \) for some dimension \( d \ll n, m \). The maximum likelihood estimator for \( u_i \) 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 \( ED(\mu, \phi) \) with mean parameter \( \mu \) and dispersion parameter \( \phi \), the model for \( Y \in \mathbb{R}^{n \times m} \) is a generalized low rank model with link function \( g \) when

\[
\begin{align*}
  y_{ij} &\sim ED(\mu_{ij}, \phi) \\
g(\mu_{ij}) &= \eta_{ij} \\
\eta_{ij} &= u_i^T v_j + \psi_i + \omega_j,
\end{align*}
\]

where \( u_i, v_j \in \mathbb{R}^d \) are the rows of matrices \( U \in \mathbb{R}^{n \times d} \) and \( V \in \mathbb{R}^{m \times d} \), respectively, and \( \psi, \omega \in \mathbb{R}^d \) 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 \( \hat{\beta} \), 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-occurrence 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-occurrence 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.

\[ \text{Data: Co-occurrence Matrix } Y \in \mathbb{R}^{w \times c} \]

\[ \text{Require: Distribution } ED(x; \mu, \varphi), \text{ number of steps } T, \text{ Dimension } d \]

\[ \text{Result: } \hat{U} \in \mathbb{R}^{w \times d}, \hat{V} \in \mathbb{R}^{c \times d}, \psi \in \mathbb{R}^w, \omega \in \mathbb{R}^c \]

Initialize \( \mu^{(0)} = Y \);

Compute \( W^{(0)} \) and \( Z^{(0)} \) according to Equations (8) and (9):

\[
\begin{align*}
\text{for } t = 0 : T \text{ do} & \\
\text{Evaluate the least squares problem} & \\
\arg\min_{U \in \mathbb{R}^{w \times d}, V \in \mathbb{R}^{c \times d}, \psi \in \mathbb{R}^w, \omega \in \mathbb{R}^c} \sum_{i,j=1}^{w,c} W_{ij}^{(t)} \left( u_i^T \psi_j + \omega_i + \psi_j - z_{ij}^{(t)} \right)^2; & \\
\text{Update } \mu^{(t+1)} \text{ according to } & \\
\mu_{ij}^{(t+1)} = u_i^T \psi_j + \omega_i + \psi_j; & \\
\text{Update } W^{(t+1)} \text{ according to Equation (8);} & \\
\text{Update } Z^{(t+1)} \text{ according to Equation (9);} & \\
\text{end} & \\
\text{return } \hat{U}^{(T)}, \hat{V}^{(T)}, \hat{\psi}^{(T)}, \hat{\omega}^{(T)} & \\
\end{align*}
\]

\textbf{Algorithm 1: IRLS Algorithm for Generalized Low Rank Models}

We regularize the word vectors by including the penalty

\[
\lambda \frac{2}{\| U \|^2_F + \| V \|^2_F} \quad (3)
\]

with \( \lambda = .002 \) for two reasons. One is to reduce some noise in the estimation of the word vectors. 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 \( \hat{U} \hat{V}^T = U \Sigma \hat{V}^T \), for \( U \) and \( \hat{V} \) orthogonal and \( \Sigma \) diagonal.

\[ \text{Mu et al. [2018] Theorem 1] shows that using penalty (3) results in having } \hat{U} = U \Sigma^{1/2} Q \text{ and } \hat{V} = \hat{V} \Sigma^{1/2} Q \text{ for some orthogonal matrix } Q. \]

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 Example 1: GloVe

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

Now we must determine a plausible distribution for the co-occurrence 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

\[
\text{Tweedie} = \sum_{i=1}^{\text{Poisson}} \text{Gamma}_i
\]

\[
= \sum_{i=1}^{\text{Poisson}} \frac{1}{\text{InvGamma}_i}
\]

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

We argue that the Tweedie distribution is a reasonable distribution for the entries of the co-occurrence matrix. We do this by connecting the Poisson and Gamma distributions in displayed above to attributes of the corpus. Intuitively it is reasonable that the number of times word \(w_i\) and word \(w_j\) co-occur within the corpus can be modeled as having a Poisson distribution. Another intuitive distribution is that of an Inverse Gamma distribution for the number of tokens between word \(w_i\) and word \(w_j\) at some co-occurrence for some choice of rate and scale parameters, although it is an approximation as the number of tokens must be integer but the Inverse Gamma has support over the positive reals.

Next, we find the weight \(W\) and the pseudo-response \(Z\) that the Tweedie distribution provides. This amounts to plugging in the cumulant generating function \(b\) from \text{[7.1]} This gives

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

When the algorithm is initialized with \(\hat{\mu}^{(0)} = Y\), the pseudo response simplifies to \(z_{ij} = \log y_{ij}\).

Further taking \(p = 1.25\), the weight simplifies to \(y_{ij}^{3/4}\). In summary, we’ve shown that:

\textbf{Result 1} Inputting the harmonic co-occurrence matrix used by GloVe, Tweedie distribution with power \(p = 1.25\), number of iterations \(T = 1\) into our algorithm results in GloVe (without any weight truncation.)

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

\section{4.3 Example 2: SVD, Skip-gram, and More}

We consider another co-occurrence matrix to summarize the corpus. In this case, we will pick the co-occurrence matrix \(Y \in \mathbb{R}^{w \times c}\) such that \(Y_{ij}\) is the number of times word \(w_i\) appears in context \(c_j\).

\subsection{4.3.1 The SVD}

For step 2 of our method, one proposed distribution for the entries of \(Y\) is a Gaussian distribution. For step 3, the IRLS algorithm will always converge in one iteration, so our method recovers the method of computing a truncated SVD by \text{[Eckart and Young] [1936]}.

\subsection{4.3.2 Skip-gram}

For step 2 of our method, another proposed distribution for the entries of \(Y\) is a multinomial distribution. Specifically, we could propose that the the rows of \(Y\) corresponding to the \(i^{th}\) word have the distribution

\[
y_i \sim \text{Multinomial} \left( \sum_{j=1}^{c} y_{ij}, \pi \right),
\]

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

We suspect that some weight and pseudo-reponse with this distribution will yeild a one-step estimator which is the estimator of \text{[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-occurrence matrix \( Y \).

### 4.3.3 Poisson Estimator

In Sections 4.3.2 and 4.3.1 we note some model mis-specification. The co-occurrence matrix \( Y \) had entries appearing on both the left-hand and right-hand side, since it is unknown what the total number of times word \( w \) will appear in the corpus. This means that the Poisson distribution is a more appropriate model for the co-occurrence matrix \[ \text{Agresti} [2014, \text{Section 7.2.1}] \]. For step 2, we could instead propose a Poisson distribution to model the entries of the co-occurrence matrix, which should yield improved results over the multinomial model.

### 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 \[ \text{Mikolov et al.} [2013a] \] 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 \[ \text{Pennington et al.} [2014] \]. 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 \( u_d \) to \( u_b - u_a + u_c \) 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 \[ \text{Ethayarajh et al.} [2018], \text{Gittens et al.} [2017] \] 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 \( \lambda \), 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 \( \frac{1}{2} U + \frac{1}{2} V \).

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 \[ \text{Duchi et al.} [2011] \] with a fixed learning rate.

---

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 \[ \text{http://mattmahoney.net/dc/text8.zi} \].

3. 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.
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-occurrence matrix $Y$. One strategy is to simply input $Y$ into step 3 of our algorithm. A second strategy is to truncate the weight $W$ that results from step 3 of our algorithm, in a strategy similar to the one used in [Pennington et al., 2014].

---

Table 1: Results of multiple iterations of the IRLS algorithm with weight truncation 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>70.6</td>
<td>45.51</td>
<td>50.28</td>
</tr>
<tr>
<td>2</td>
<td>71.94</td>
<td>45.97</td>
<td>50.9</td>
</tr>
<tr>
<td>3</td>
<td>72.38</td>
<td>46.47</td>
<td>51.39</td>
</tr>
<tr>
<td>4</td>
<td>72.29</td>
<td>46.51</td>
<td>51.41</td>
</tr>
<tr>
<td>5</td>
<td>72.49</td>
<td>46.57</td>
<td>51.47</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$p$</th>
<th>Iterations</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.25</td>
<td>1</td>
<td>69.89</td>
<td>44.32</td>
<td>49.18</td>
</tr>
<tr>
<td>1.25</td>
<td>2</td>
<td>72.91</td>
<td>45.41</td>
<td>50.63</td>
</tr>
<tr>
<td>1.5</td>
<td>1</td>
<td>72.2</td>
<td>45.7</td>
<td>50.73</td>
</tr>
<tr>
<td>1.5</td>
<td>2</td>
<td>72.2</td>
<td>45.72</td>
<td>50.75</td>
</tr>
<tr>
<td>1.75</td>
<td>1</td>
<td>60.3</td>
<td>43.1</td>
<td>46.36</td>
</tr>
<tr>
<td>1.75</td>
<td>1</td>
<td>64.83</td>
<td>42.49</td>
<td>46.74</td>
</tr>
<tr>
<td>1.9</td>
<td>1</td>
<td>50.8</td>
<td>40.18</td>
<td>42.4</td>
</tr>
<tr>
<td>1.9</td>
<td>2</td>
<td>60.39</td>
<td>39.43</td>
<td>43.41</td>
</tr>
</tbody>
</table>

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

---

5.3 Experiment 1: Multiple Iteration Effects

We present results of multiple iterations 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.

---

Models are trained for up to 50 epochs (50 passes through the co-occurrence matrix) with batches of size 512.

---

This training procedure is slightly different from the asynchronous stochastic gradient descent training procedure used in [Pennington et al., 2014].
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 \( \frac{1}{10} \) th of the size of their corpus.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Iterations</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>69.89</td>
<td>44.32</td>
<td>49.18</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>72.91</td>
<td>45.41</td>
<td>50.63</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>73.18</td>
<td>47.87</td>
<td>52.67</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>75.22</td>
<td>47.97</td>
<td>53.15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>55.95</td>
<td>44.47</td>
<td>46.65</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>55.95</td>
<td>45.66</td>
<td>47.61</td>
</tr>
</tbody>
</table>

Table 3: Results for multiple choices of regularizing the large values of the co-occurrence matrix. Our
strategies are (1) count matrix, (2) truncation of the weight only, (3) truncation of the co-occurrence
matrix to \( y_{\max} = 10 \).

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 Experiment 4: Regularization Effects

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Iterations</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalty</td>
<td>1</td>
<td>69.89</td>
<td>44.32</td>
<td>49.18</td>
</tr>
<tr>
<td>Penalty</td>
<td>2</td>
<td>72.91</td>
<td><strong>45.41</strong></td>
<td><strong>50.63</strong></td>
</tr>
<tr>
<td>No Penalty</td>
<td>1</td>
<td>72.91</td>
<td>42.26</td>
<td>48.09</td>
</tr>
<tr>
<td>No Penalty</td>
<td>1</td>
<td><strong>73.62</strong></td>
<td>43.26</td>
<td>49.03</td>
</tr>
</tbody>
</table>

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.

Acknowledgements

We thank YA, BA, DM, and WZ for their helpful contributions. The work in this paper is supported
by an AWS grant.
References


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 \( f(y; \theta, \varphi) \) of \( y \) satisfies

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

over its support, then the distribution of \( y \) is in the exponential dispersion family. The parameter \( \theta \) is
the natural parameter, \( \varphi \) 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 \( \theta \), its
mean \( \mu = b'(\theta) \), so we can equivalently specify the mean \( \mu \) or the natural parameter \( \theta \).

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

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

showing that here the natural parameter \( \theta = \mu \), the dispersion parameter \( \varphi = \sigma^2 \), the functions
\( b(\theta) = \frac{1}{2}\theta^2 \), \( a(\varphi) = \sigma^2 \), 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 importance 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

\[
\text{var}(y) = \varphi \left( \mathbb{E}[y] \right)^p
\]

and the distribution of \( y \) is an exponential dispersion family. In this case, we write \( y \sim \text{Tweedie}_p(\mu, \varphi) \), 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 \( \alpha \) and rate parameter \( \beta \) 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

\[
\text{Tweedie}_p(\mu, \varphi) = \sum_{i=1}^{n} g_i,
\]

where \( n \sim \text{Poisson}(\lambda) \) and \( g_i \sim \text{Gamma}(\alpha, \beta) \), and \( \lambda = \frac{\mu^{2-p}}{(2-p)\beta^2}, \alpha = \frac{2-p}{p-1}, \text{ and } \beta = \frac{\mu^{1-p}}{(p-1)^2p} \)
[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-occurrence matrices.

Specifically,

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

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 $b(\theta) = \frac{1}{2-p} ((1-p)\theta) \frac{1}{\phi(1-p)}$.

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

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

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

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

$$
\begin{align*}
&y_i \mid \text{ind.} \sim ED(\mu_i, \phi), \\
g(\mu_i) = \eta_i, \\
\eta_i = x_i^T \beta. 
\end{align*}
$$

In the first line of the displayed relationships, the distribution of the response $y$ is described. In the third line, the systematic component $\eta_i$ expresses the effect of the covariates $x_i$. 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 $g(\mu) = \logit(\mu) = \log \frac{\mu}{1-\mu}$ 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 $\beta$ is unknown and a target of estimation in the generalized linear model. The standard approach to compute $\beta$ in a generalized linear model is maximum likelihood estimation [Fisher, 1922, Section 7] to produce the maximum likelihood estimator, or MLE $\hat{\beta}$, with $\beta$ defined to be the maximizer of the likelihood of $y$ given $X$. Define $\ell(\beta)$ 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 $\{\beta^{(t)}\}_{t=1}^{\infty}$ starting with some initialization $\beta^{(0)}$ so that

$$
\beta^{(t+1)} = \beta^{(t)} + \left( \mathbb{E} \left[ D^2 \ell(\beta) \right] \right)^{-1} \nabla \ell(\beta^{(t)}),
$$

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

$$
\hat{\beta}^{(t+1)} = \arg \min_{\beta \in \mathbb{R}^p} \left\| (W^{(t)})^{1/2} (X\beta - z^{(t)}) \right\|_2^2,
$$
where the weight $W^{(t)}$ is diagonal with
\[ W_{ii}^{(t)} = \left[ \left( g' (\mu_i^{(t)}) \right)^2 b'' \left( b' - 1 \left( \mu_i^{(t)} \right) \right) \right]^{-1} \] (8)
and the pseudo-response for observation $i$ is
\[ z_i^{(t)} = \eta_i^{(t)} + g' (\mu_i^{(t)}) (y_i - \mu_i^{(t)}) , \] (9)
where $\eta_i^{(t)} = X \hat{\beta}^{(t)}$ and $\mu_i^{(t)} = g^{-1}(\eta_i^{(t)})$. 

where $W^{(t)}$ is diagonal with
\[ W_{ii}^{(t)} = \left[ \left( g' (\mu_i^{(t)}) \right)^2 b'' \left( b' - 1 \left( \mu_i^{(t)} \right) \right) \right]^{-1} \] (8)
and the pseudo-response for observation $i$ is
\[ z_i^{(t)} = \eta_i^{(t)} + g' (\mu_i^{(t)}) (y_i - \mu_i^{(t)}) , \] (9)
where $\eta_i^{(t)} = X \hat{\beta}^{(t)}$ and $\mu_i^{(t)} = g^{-1}(\eta_i^{(t)})$. 

13