Exponential Family Word Embeddings: An Iterative Approach for Learning Word Vectors

Benjamin R. Baer[*]
Cornell University
Ithaca, NY 14853
brb225@cornell.edu

Skyler Seto[*]
Cornell University
Ithaca, NY 14853
ss3349@cornell.edu

Martin T. Wells
Cornell University
Ithaca, NY 14853
mtw1@cornell.edu

Abstract

GloVe and Skip-gram word embedding methods learn word vectors by decomposing a denoised matrix of word co-occurrences into a product of low-rank matrices. In this work, we propose an iterative algorithm for computing word vectors based on modeling word co-occurrence matrices with 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 co-occurrence matrix, distribution of co-occurences, and the number of iterations in the iterative algorithm. For example, using a Tweedie distribution with one iteration results in GloVe and using a Multinomial distribution with full-convergence mode results in Skip-gram. Experimental results demonstrate that multiple iterations of our algorithm improves results over the GloVe method on the Google 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, 1957]. 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 X, then output word vectors with some dimension d.

Skip-gram processes a corpus with w words into a count co-occurence matrix $X \in \mathbb{R}^{w \times w}$, where x_{ij} is the number of times word w_i appears in the same context as the word w_j. Here, two words being in the same context means that they're within l_c tokens of each other. Define this co-occurence matrix to be the *count co-occurence matrix*. Next, Skip-gram [Pennington et al., 2014, Section 3.1] estimates

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

where $\boldsymbol{u}_i^T$ is the i^{th} row of U, then defines the word vectors to be the rows of $\hat{U}$.

GloVe processes a corpus with w words into a harmonic co-occurence matrix $X \in \mathbb{R}^{w \times w}$ where x_{ij} is the harmonic sum of the number of tokens between words w_i and w_j over each co-occurrence. That is, $x_{ij} = \sum_{p_1 < p_2, |p_1 - p_2| \leq l_c, \mathcal{D}(p_1) = w_i, \mathcal{D}(p_2) = w_j} \frac{1}{|p_1 - p_2|}$, where $\mathcal{D}(p_1)$ is the p_1^{th} word in the corpus and l_c is the length of the *context window*. Define this co-occurence matrix to be the *harmonic*

[*]Indicates co-first author; authors contributed equally and order was determined alphabetically.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

co-occurence matrix. Next, GloVe estimates

$$(\hat{U}, \hat{V}, \hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}) = \arg \min_{U,V \in \mathbb{R}^{w \times d}; \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^w} \sum_{i=1}^{w} \sum_{j=1}^{w} h(x_{ij}) \left(\boldsymbol{u}_i^T \boldsymbol{v}_j + a_i + b_j - \log x_{ij} \right)^2, \tag{2}$$

where a_i and b_j are bias terms, $h(x_{ij}) = (\min\{x_{ij}, x_{\max}\})^{.75}$ is the weight, and $x_{\max}$ 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 X is introduced by processing the corpus, and an objective function is introduced to find a low rank factorization related to the co-occurences X. In this paper, we derive the objective functions from a model-based perspective. We introduce an iterative algorithm, and show that problem (1) results from running the iterative algorithm on full-convergence mode for a Multinomial model and problem (2) is one step of the iterative algorithm for a Tweedie model. This algorithm additionally allows us to introduce methods to "fill in the gaps" between Skip-gram and GloVe and to introduce altogether new methods for finding word vectors.

2 Related Work

We saw that Skip-gram and GloVe compute a co-occurence matrix X which results from processing the corpus $\mathcal{D}$ and an objective function J to relate the matrix X to a product of low rank matrices U and V. Many existing approaches for explaining word embedding methods do so by identifying or deriving the co-occurence matrix X or the objective function J. In this section, we review relevant work in this area, which helps frame our approach discussed in Section 4.1.

Much of the related work involves using the co-occurence matrix from Skip-gram. For the remainder of this section, let X be the count co-occurence 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 an objective function J which is invariant to orthogonal transformation. For simplicity, we specialize to the Frobenius norm and say these early approaches find

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

That is, here $J(M, X) = \|M - X\|_F^2$ is the objective function and X is the co-occurence matrix.

The co-occurence matrix and the loss function for Skip-gram can be read off from problem (1): the co-occurence matrix is X and the objective function is written in problem (1) with $\boldsymbol{u}_i^T \boldsymbol{v}_j$ replaced by m_{ij}. Cotterell et al. [2017] find a probabilistic interpretation of this loss function related to a Multinomial distribution, but do not take advantage of it and only replace the inner product with a (higher dimensional) variant, somewhat similar to the approach in Tifrea et al. [2018].

Mikolov et al. [2013a] 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 appear within a context window of another word, 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 that is a shifted PMI matrix.[2] However, they did not determine the objective function. Later, Li et al. [2015] and Landgraf and Bellay [2017] explicitly identified both the co-occurence matrix and the objective function. 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 establish that SGNS is finding a low-rank matrix related to X, the same matrix that Skip-gram uses. The loss function is

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

[2] Define the total number of times word w_i appears to be $x_{i\cdot} = \sum_{j=1}^{w} x_{ij}$, the total number of times context w_j appears to be $x_{\cdot j} = \sum_{i=1}^{w} x_{ij}$, and the total number of words to be $x_{\cdot\cdot} = \sum_{i,j=1}^{w,w} x_{ij}$. The shifted PMI matrix has entries $\log \frac{x_{ij} x_{\cdot\cdot}}{x_{i\cdot} x_{\cdot j}} - \log k$.

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 prediction from within their model.

The approach in this paper will be to view the entries of the co-occurence matrix as random variables and introduce an objective function via the likelihood of that random variable. Our approach is most similar to Landgraf and Bellay [2017] and, to a lesser extent, Cotterell et al. [2017]. In order proceed, some background in probabilistic modeling and estimation needs to be developed.

3 Background

In this section, we review iteratively reweighted least squares (IRLS) for generalized linear models and review generalized low rank models [Udell et al., 2016]. Further background (and notation) in exponential dispersion families and generalized linear models is developed in Section A.

3.1 Iteratively Reweighted Least Squares

Generalized linear models (GLMs) are a flexible generalization of linear regression where the mean is a not necessarily linear function of a coefficient $\boldsymbol{\beta}$ and the response has an error distribution which is an exponential dispersion family. The coefficient $\boldsymbol{\beta}$ is unknown and a target of estimation. The standard approach to estimate $\boldsymbol{\beta}$ is *maximum likelihood estimation* [Fisher, 1922, Section 7] to produce the *maximum likelihood estimator*, or MLE, $\hat{\boldsymbol{\beta}}$.

A computational approach to find the MLE is through Fisher scoring, a variant of Newton's method on the log likelihood which uses the expectation of the Hessian in place of the Hessian [Agresti, 2015, Section 4.5]. Define $\ell(\boldsymbol{\beta})$ to be the log likelihood. Specifically, Fisher scoring produces a sequence of estimates $\{\hat{\boldsymbol{\beta}}^{(t)}\}_{t=1}^{\infty}$ starting with some initialization $\hat{\boldsymbol{\beta}}^{(0)}$ so that $\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} + \left(\mathbb{E}\left[D^2\ell(\boldsymbol{\beta})\right]\Big|_{\hat{\boldsymbol{\beta}}^{(t)}}\right)^{-1} \nabla\ell(\hat{\boldsymbol{\beta}}^{(t)})$, where $\nabla\ell$ is the gradient and $D^2\ell$ is the Hessian. Upon plugging in the gradient and expected Hessian for an exponential dispersion family, a surprising identity emerges: each iteration of Fisher scoring is equivalent to minimizing a weighted least squares objective:

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\| \left(H^{(t)}\right)^{1/2} \left(X\boldsymbol{\beta} - \boldsymbol{z}^{(t)}\right) \right\|_2^2, \tag{3}$$

where the weight $H^{(t)}$ and pseudo-response $z^{(t)}$ at iteration t have

$$h_{ii}^{(t)} = \left[\left(g'\left(\mu_i^{(t)}\right)\right)^2 b''\left((b')^{-1}\left(\mu_i^{(t)}\right)\right)\right]^{-1}, \; z_i^{(t)} = \eta_i^{(t)} + g'\left(\mu_i^{(t)}\right)\left(y_i - \mu_i^{(t)}\right), \tag{4}$$

$h_{ij} = 0$ for $i \neq j$, $\boldsymbol{\eta}^{(t)} = X\hat{\boldsymbol{\beta}}^{(t)}$, and $\mu_i^{(t)} = g^{-1}(\eta_i^{(t)})$.

3.2 Generalized Low Rank Models

Principal components analysis [Jolliffe, 2011] is one well-known method for finding a low rank matrix related to $X \in \mathbb{R}^{w\times c}$. In principal components analysis, we model $x_{ij} \overset{\text{ind.}}{\sim} \text{Normal}(\boldsymbol{u}_i^T\boldsymbol{v}_j, \sigma^2)$ where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^d$ for some dimension $d \ll w, c$. A maximum likelihood estimator for $\boldsymbol{u}_i$ is taken to be a low-dimensional embedding of the i^{th} row of X. The low-dimensional embedding enables interpretability and reduces noise. However, data cannot always be viewed as being drawn from a normal distribution, so it's necessary to extend the method of principal components to non-normal data. The extension can be made in a manner similar to the extension from linear models to generalized linear models: the new model, called a *generalized low rank model* [Udell et al., 2016] allows 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* $X \in \mathbb{R}^{w\times c}$ *is a* generalized low rank model *with* link function g *when*

$$\begin{cases} x_{ij} \overset{\text{ind}}{\sim} \text{ED}(\mu_{ij}, \varphi) & (5) \\ g(\mu_{ij}) = \eta_{ij} = \boldsymbol{u}_i^T\boldsymbol{v}_j + a_i + b_j, , & (6) \end{cases}$$

where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^d$ are the rows of matrices $U \in \mathbb{R}^{w \times d}$ and $V \in \mathbb{R}^{c \times d}$, respectively, and $\boldsymbol{a} \in \mathbb{R}^w$ and $\boldsymbol{b} \in \mathbb{R}^c$ are bias (or offset) terms.

The difference between the generalized low rank model and the generalized linear model is in the systematic component in equation (6). Here, the data is modeled as having its link-transformed mean be a matrix with rank at most d. This formalizes the way in which we relate the co-occurence matrix X to a low rank factorization.

When the link function g is taken to be canonical, the generalized low rank model is identical to ePCA [Collins et al., 2002]. The generalization is worthwhile since the canonical link can be inappropriate, as we will see, for instance, in Section 5.1.

4 Methodology

We now present a method to find word vectors. A key innovation in the method is an iterative algorithm inspired by IRLS to find a maximum likelihood estimator in a generalized low rank model.

4.1 Our Proposed Method

Our method has three steps:

Step 1 Choose a co-occurence matrix $X \in \mathbb{R}^{w \times c}$ to summarize the document. (Note, in many cases $c = w$ so that the "contexts" are just the words.)

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

Step 3 Choose a number of iterations r to run IWLRLS (Algorithm (1)) with the input specified above to output word vectors.

Data: Co-occurrence matrix $X \in \mathbb{R}^{w \times c}$
Require: Distribution $ED(\mu, \varphi)$, link function g, number of iterations r, dimension d
Result: $\hat{U} \in \mathbb{R}^{w \times d}, \hat{V} \in \mathbb{R}^{c \times d}, \hat{\boldsymbol{a}} \in \mathbb{R}^w, \hat{\boldsymbol{b}} \in \mathbb{R}^c$
Initialize $\mu^{(0)} = X$;
for $t = 0 : r$ **do**

Update $H^{(t+1)}$ according to $h_{ij}^{(t)} = \left[\left(g' \left(\mu_{ij}^{(t)} \right) \right)^2 b'' \left((b')^{-1} \left(\mu_{ij}^{(t)} \right) \right) \right]^{-1}$;

Update $Z^{(t+1)}$ according to $z_{ij}^{(t)} = g(\mu_{ij}^{(t)}) + g' \left(\mu_{ij}^{(t)} \right) \left(x_{ij} - \mu_{ij}^{(t)} \right)$;

Evaluate the least squares problem

$$\arg \min_{U \in \mathbb{R}^{w \times d}, V \in \mathbb{R}^{c \times d}, \boldsymbol{a} \in \mathbb{R}^w, \boldsymbol{b} \in \mathbb{R}^c} \sum_{i,j=1}^{w,c} h_{ij}^{(t)} \left(\boldsymbol{u}_i^T \boldsymbol{v}_j + a_i + b_j - z_{ij}^{(t)} \right)^2;$$

Update $\mu^{(t+1)}$ according to $g(\mu_{ij}) = \boldsymbol{u}_i^T \boldsymbol{v}_j + a_i + b_j$;

end
return $\hat{U}^{(r)}, \hat{V}^{(r)}, \hat{\boldsymbol{a}}^{(r)}, \hat{\boldsymbol{b}}^{(r)}$

Algorithm 1: Iteratively weighted low rank least squares (IWLRLS) algorithm for GLRMs

The first step of our method processes the corpus in order to extract the linguistic information. Some co-occurence statistics use more information than others: for instance, the harmonic co-occurence matrix makes use of the number of tokens between words while the count co-occurence matrix does not. A typical tuning parameters here is the length l_c of the context window. We view this step as involving a "linguistic" choice.

The second step specifies a distribution for the co-occurence matrix. A distribution can be considered as plausibly corresponding to reality if it can be derived by a connection to the corpus. In our

Co-occurence:	Harmonic	Count			
Distribution:	Tweedie	Gaussian	Multinomial	Poisson	Binomial
One iteration	GloVe	SVD	·	Arora et al. [2016]	·
Early stopping	·	SVD	·	·	·
Full likelihood	·	SVD	Skip-gram	·	SGNS

Table 1: The rows refers to the number of steps of IWLRLS. A "·" represents no existing work. All filled-in positions in the lowest row were established in previous work.

framework, the model is explicit: this is helpful since knowing a model provides interpretation for its output [Gilpin et al., 2018, Section II.A.]. The choice of distribution will often determine, through convention, the link function, so the link function often does not need to be separately chosen. We view this step as involving a "statistical" choice.

The third step runs IWLRLS, a generalized version of IRLS. Recall that IRLS is derived by iteratively maximizing a second order Taylor expansion of the likelihood as a function β. The Taylor expansion is centered at the previous iterate. IWLRLS can be derived by iteratively maximizing a second order Taylor expansion of the likelhood as a function of η subject to the constraint 6. We view this as a "computational" choice that we fix in advance.

5 Examples

In the following subsections, we run through many examples of our method as it would be used in practice. There are two distinct choices of co-occurence matrices that are made. Various choices of distributions recover common methods for finding word vectors. An altogether new estimator is proposed via an improvement of the assumed distribution in Skip-gram. Casting these estimators in this general framework provides an interpretation and understanding of them: we make explicit their assumptions and therefore know the driver of their behavior.

5.1 Example 1: GloVe

We will apply our proposed method under the choice of the harmonic co-occurence matrix and the Tweedie distribution: one iteration of IWLRLS will recover GloVe.

Step 1 The first step of our method is to pick a co-occurence matrix that summarizes the corpus. We choose the harmonic co-occurence matrix $X \in \mathbb{R}^{w \times w}$.

Step 2 Now we must determine a plausible distribution for the co-occurence matrix that is an exponential dispersion family. Recall that the $\mathrm{Tweedie}$ distribution has the property mentioned in equation (12) that it is a sum of Poisson many independent Gamma distributions. An informal way to write this is that

$$\mathrm{Tweedie} \overset{\mathrm{d}}{=} \sum_{i=1}^{\mathrm{Poisson}} \mathrm{Gamma}_i \overset{\mathrm{d}}{=} \sum_{i=1}^{\mathrm{Poisson}} \frac{1}{\mathrm{InvGamma}_i}.$$

We argue that the Tweedie distribution is reasonable by connecting the Poisson and Inverse Gamma distributions 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 choice of distribution is that of an Inverse Gamma distribution for the number of tokens between word w_i and word w_j at some co-occurence, although it is an approximation as the number of tokens is an integer while the Inverse Gamma is supported on non-integers.

Instead of using the canonical link function, we will take $g(\mu) = \log \mu$, which is standard [Smyth, 1996]. A problem with the canonical link function preventing its use is that its range is nonpositive.

Step 3 Next, we find the form of the weight H and the pseudo-response Z that the Tweedie distribution provides. This amounts to plugging in the cumulant generating function ψ that is given in Section A.1. This results in

$$h_{ij} = \mu_{ij}^{2-p}, \; z_{ij} = \frac{x_{ij} - \mu_{ij}}{\mu_{ij}} + \log \mu_{ij}. \tag{7}$$

When the algorithm is initialized with $\hat{\mu}^{(0)} = X$, the pseudo-response simplifies to $z_{ij} = \log x_{ij}$. Taking the power $p = 1.25$, the weight simplifies to $x_{ij}^{3/4}$. In summary, we've shown that:

Result 1 *Inputting the harmonic co-occurence matrix, the Tweedie distribution with power $p = 1.25$, the $\log$ link, and the number of iterations $k = 1$ into IWLRLS results in GloVe (without the algorithmic regularization induced by truncating the weights.)*

Given this connection, we can extend GloVe for several iterations rather than one or even use the full likelihood. We experiment with this using real data examples in Section 6. This result shows that even though the first iteration does not depend on word pairs where $x_{ij} = 0$, later iterations do.

5.2 Example 2: SVD, Skip-gram, and More

We now consider an alternative first step: we choose another co-occurence matrix to summarize the corpus. Then, we make multiple possible choices for step 2 to illustrate connections to previous work that step 3 recovers. Various choices for step 2 will recover the SVD [Landauer et al., 1998], Skip-gram [Mikolov et al., 2013a], a new estimator which is a distributional improvement over those, and Skip-gram with negative sampling [Mikolov et al., 2013b].

Step 1 We choose the count co-occurence matrix.

5.2.1 The SVD

Step 2 A proposed distribution for the entries of X is the Gaussian distribution. This may not be the best choice, since the entries of X are non-negative integers. As is usual, we take the link function to be $g(\mu) = \mu$. We restrict the systematic component to not include the bias terms, so that $\eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j$.

Step 3 We showed in Section A.1 that the cumulant generation function from the normal distribution is $\psi(\theta) = \frac{1}{2}\theta^2$. This makes it so that

$$h_{ij} = 1, \; z_{ij} = x_{ij}. \tag{8}$$

In other words, the IWLRLS algorithm will always converge in one iteration, so our method recovers the method of computing a truncated SVD of X by Eckart and Young [1936].

Another choice that could have been made in step 2 is to have the link function $g(\mu) = \log \mu$. This still may not be the best choice since the normal distribution still has the same problems as before.

5.2.2 Skip-gram

Step 2 Another proposed distribution for the entries of X is a Multinomial distribution. Specifically, we could propose that the the row of X corresponding to word w_i has the distribution $\boldsymbol{x}_i \sim \text{Multinomial}\left(\sum_{j=1}^w x_{ij}, \boldsymbol{\pi}\right)$, where $\boldsymbol{\pi} \in \mathbb{R}^w$ is vector of probabilities of word w_i appearing within a context window with the other words and $\sum_{j=1}^w x_{ij}$ is the total number of times word w_i appears in the corpus. We take the link function to be the multi-logit.[3]

Cotterell et al. [2017] show that the objective function of Skip-gram coincides with the likelihood of this model when the bias terms are removed, so that the systematic component $\eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j$ instead of the usual representation in equation 6.

Step 3 The Poisson trick [Birch, 1963] can be used to reduce estimation in a Multinomial model to estimation in a particular Poisson model. Let $\hat{U}$, $\hat{V}$ be the maximum likelihood estimators in the Multinomial generalized low rank model described in step 2. Using this trick, it holds that $\hat{\boldsymbol{a}}$, (the same) $\hat{U}$, and (the same) $\hat{V}$ are maximum likelihood estimators in a Poisson generalized low rank model with independent responses x_{ij} and systematic component

$$\eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j + a_i. \tag{9}$$

Notice that there is only one bias term. The weight and pseudo-response are

$$h_{ij} = \mu_{ij}, \; z_{ij} = \frac{x_{ij} - \mu_{ij}}{\mu_{ij}} + \log \mu_{ij}. \tag{10}$$

[3]The Multinomial distribution is not in the exponential dispersion family, while it is in the multivariate exponential dispersion family. (See Section A.1.1.) In step 3, the problem is reduced to one in an exponential dispersion family.

5.2.3 Poisson Estimator

In the previous subsubsection, we saw that the choice of Multinomial model implicitly gives rise to a Poisson model with a systematic component given by equation (9). Since it could be most appropriate to have bias terms for both rows and columns due to the symmetry of the co-occurence matrix, we directly introduce a Poisson estimator with a non-restricted systematic component.

Step 2 Another proposed distribution is a Poisson. Due to the "law of rare events" [Durrett, 2010, Section 3.6.1], this is a plausible model. We use the canonical link function $g(\mu) = \log \mu$.

Step 3 The cumulant generating function is $\psi(\theta) = \exp(\theta)$ [Agresti, 2015], so that the weight and pseudo-response are given by equations (10).

Arora et al. [2016] propose an estimator which is a close variant of one iteration of IWLRLS. At one point in their derivation, they (using our notation) take $\eta_{ij} = \|\boldsymbol{u}_i - \boldsymbol{v}_j\|_2^2 + c$, where c is an arbitrary constant which does not depend on the word. This is inspired by their theorem 2.2. On the other hand, taking $\eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j + a_i + b_j$ (as in equation 6) in their derivation recovers one iteration of IWLRLS.

The Negative-Binomial distribution is commonly used as an alternative for the Poisson in the presence of over-dispersion, which is the case when the variance is higher than the mean. It produces the same weight and pseudo-response as the Poisson.

5.2.4 Skip-gram with Negative Sampling

Step 2 We model $x_{ij} \overset{\text{ind.}}{\sim} \text{binomial}\,(s_{ij}, \pi_{ij})$, where $s_{ij} = x_{ij} + k\frac{x_{i\cdot}x_{\cdot j}}{x_{\cdot\cdot}}$ is an inflated count, $\eta_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j$, and $k \geq 0$. Landgraf and Bellay [2017] showed that a maximum likelihood estimator from this model with canonical link $g(\pi) = \log \frac{\pi}{1-\pi}$ is identical to a SGNS estimator.

Step 3 The cumulant generating function for the binomial distribution is $\psi(\theta) = \log(1 + \exp\theta)$, so the weight and pseudo-response are:

$$h_{ij} = \pi_{ij}(1 - \pi_{ij}),\ z_{ij} = \log \frac{\pi_{ij}}{1 - \pi_{ij}} + \frac{1}{\pi_{ij}(1 - \pi_{ij})}\left(\frac{x_{ij}}{s_{ij}} - \pi_{ij}\right) \tag{11}$$

6 Experiments

In Section 4.1 we introduced the IWLRLS algorithm to compute word vectors such as those produced by GloVe or SGNS. 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 6.1 we introduce the analogy similarity task for evaluating word vectors. In Section 6.2 we present results of the algorithm with different distributions according to those presented in Section 5.1 and 5.2. In Section B.1 we provide parameter configurations and training procedures, and in Sections B.2-B.5 we present results of IWLRLS in numerous scenarios showcasing improvement through multiple iterations and robustness to other model parameters.

6.1 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 $\boldsymbol{u}_b - \boldsymbol{u}_a + \boldsymbol{u}_c$[4] in the embedding space via cosine similarity[5]. 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 4.1.

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

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

6.2 Experimental Results

In this section, results of the IWLRLS algorithm are performed for the Tweedie, Multinomial, and Poisson models. Based on the additional experiments in Sections B.2-B.6 we train the Tweedie model with $p = 1.25$ (Section B.3) and for all models include weight truncation to penalize large co-occurrences (Section B.4), regularization terms (outlined in Section B.5), and include only a single bias term within the systematic component of the Tweedie model (Section B.6).

Step	Semantic			Syntactic			Total		
	Tweed	Pois	Mult	Tweed	Pois	Mult	Tweed	Pois	Mult
One-step	71.4	61.37	**73.27**	**47.62**	43.18	46.63	**52.13**	46.63	50.65
Early-stop	73.98	64.83	**75.22**	45.64	43.43	**46.05**	51.02	47.49	**51.59**
Full-likelihood	74.51	66.87	**76.64**	**48.20**	43.03	47.43	**53.20**	47.56	52.98

Table 2: Accuracy of the IWLRLS algorithm for Multinomial, Tweedie, and Poisson distributions on the Google word analogy task.

To demonstrate the effectiveness of performing multiple iterations of the IWLRLS algorithm, we present results for the one-step estimator, an early-stopped estimator, and the full-likelihood estimator. Of particular interest in our results are the Tweedie one-step estimator (a variant of the GloVe method), and the full-likelihood estimator for the Multinomial (a variant of the Skip-gram method). For the results in Table 2, the full-likelihood result is taken to be the iteration which achieves the maximum total accuracy on the analogy task, and the early-stop algorithm is taken to be an iteration between the one-step and full-likelihood iterations which performs best in total accuracy on the analogy task. For both the Tweedie and Multinomials, the full-likelihood result is the result after 3 iterations and the early-stopped result is the result after 2 iterations. For the Poisson model, the full-likelihood result is the result after 9 iterations, and the early-stopped result is the result after 3 iterations.

We find a small difference in total accuracy on the analogy task with the one-step estimator (GloVe) and the full-likelihood differing by roughly 1%. We find a similar relationship in the Poisson estimator and further note that the early-stopped estimator for the Poisson has very similar accuracy to the full-likelihood algorithm. Finally, the Multinomial model yields a difference of 2% between the full-likelihood algorithm (Skip-gram) and the one-step algorithm. The early-stopped algorithm for the Multinomial also performs 1% higher than the one-step algorithm indicating a fair tradeoff between running an additional iteration and stopping after only one iteration.

7 Conclusion

We present a general model-based methodology for finding word vectors from a corpus. This methodology involves choosing the distribution of a chosen co-occurrence matrix to be an exponential dispersion family and choosing the number of iterations to run our algorithm.

In Table 1, we see that our methodology unifies the dominant word embedding methods available in the literature and provides new and improved methods. We introduce an extension of Skip-gram that is stopped before full-convergence analagously to GloVe and an extension to GloVe beyond one iteration. Experimental results on a small corpus demonstrate our method improves upon GloVe and Skip-gram on the Google word analogy similarity task. It is our hope that this methodology can lead to the development of better, more statistically sound, word embeddings and consequently improve results on many other downstream tasks.

Acknowledgements

We thank Robin Alexander, Ben Athiwaratkun, Daniel E. Gilbert, David Mimno, and Wenyu Zhang for their helpful contributions. The work in this paper is supported by an AWS Cloud Credits for Research grant.

References

Alan Agresti. *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

M. W. Birch. Maximum likelihood in three-way contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 220–233, 1963.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624, 2002.

Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. Explaining and generalizing skip-gram through exponential family principal component analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 175–181, 2017.

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.

R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010. ISBN 9781139491136.

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.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

RA Fisher. On the mathematical foundations of theoretical statistics. *Phil. Trans. R. Soc. Lond. A*, 222(594-604):309–368, 1922.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An approach to evaluating interpretability of machine learning. *arXiv preprint arXiv:1806.00069*, 2018.

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 Jørgensen. Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 127–162, 1987.

Bent Jørgensen. *The theory of dispersion models*. CRC Press, 1997.

Tatsuru Kobayashi and Kumiko Tanaka-Ishii. Taylor's law for human linguistic sequences. *arXiv preprint arXiv:1804.07893*, 2018.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

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. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

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. *arXiv preprint arXiv:1601.00893*, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

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.

Gordon K Smyth. Regression analysis of quantity data with exact zeros. In *Proceedings of the Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management, Gold Coast, Australia*, pages 17–19. Citeseer, 1996.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

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, Stephen Boyd, et al. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

A Extended Background

Further background in exponential dispersion families and generalized linear models is developed here.

A.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*

$$\log f(y; \theta, \varphi) = \frac{y\,\theta - \psi(\theta)}{\delta(\varphi)} + c(y; \varphi)$$

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 ψ is the* cumulant generating function*.*

In many cases, the function $\delta(\varphi)$ is very simple, meaning that, for instance, $\delta(\varphi) = 1$ or $\delta(\varphi) = \varphi$. 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 $\mu = \psi'(\theta)$, 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

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

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

The Tweedie distribution [Jørgensen, 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 [Jørgensen, 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

$$\mathrm{Tweedie}_p(\mu, \varphi) = \sum_{i=1}^{n} g_i, \tag{12}$$

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 $\beta = \frac{\mu^{1-p}}{(p-1)\varphi}$ [Jørgensen, 1997]. 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 due to some words never appearing within the same context. 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 ψ, Jørgensen [1997] show that the Tweedie distribution has $\psi(\theta) = \frac{1}{2-p}\left((1-p)\theta\right)^{\frac{2-p}{1-p}}$.

A.1.1 Multivariate Exponential Dispersion Families

Exponential dispersion families are defined over real numbers. Now, we generalize their definition to a multivariate setting.

Definition 4 *Let $\boldsymbol{y} \in \mathbb{R}^m$ be a random vector. If the density function $f(\boldsymbol{y}; \boldsymbol{\theta}, \varphi)$ of $\boldsymbol{y}$ satisfies*

$$\log f(\boldsymbol{y}; \boldsymbol{\theta}, \varphi) = \frac{\boldsymbol{y}^T \boldsymbol{\theta} - \psi(\boldsymbol{\theta})}{\delta(\varphi)} + c(\boldsymbol{y}; \varphi)$$

over its support, then the distribution of $\boldsymbol{y}$ is in the multivariate exponential dispersion family. *The parameter $\boldsymbol{\theta} \in \mathbb{R}^m$ is the* natural parameter*, $\varphi \in \mathbb{R}$ is the* dispersion parameter*, and the function $\psi : \mathbb{R}^m \to \mathbb{R}$ is the* cumulant generating function*.*

A collection of independent draws from the same exponential dispersion family is a multivariate exponential dispersion family. To see this, let y_i $(i = 1, \ldots, m)$ be i.i.d. from an exponential dispersion family. Then, the density of $\boldsymbol{y}$ satisfies $\log f(\boldsymbol{y}; \boldsymbol{\theta}, \varphi) = \sum_{j=1}^m \log f(y_i; \theta_i, \varphi) = \sum_{j=1}^m \frac{y_j\theta_j - \psi(\theta_j)}{\delta(\varphi)} + c(y_j; \varphi)$, which has cumulant generation function $\psi(\boldsymbol{\theta}) = \sum_{j=1}^m \psi(\theta_j)$.

Another useful example of a multivariate exponential dispersion family is the Multinomial. Let $\boldsymbol{x} \in \mathbb{R}^c$ have be distributed as $\boldsymbol{x} \sim \text{multinomial}(s, \boldsymbol{\pi})$, where $s \in \mathbb{N}$ is the total number of draws and $\boldsymbol{\pi} \in \mathbb{R}^c$ is the probability vector. Introduce a change of parameters where $\pi_j = \frac{\exp\theta_j}{\sum_{k=1}^c \exp\theta_k}$. Then the log density

$$\log \prod_{j=1}^c \pi_j^{x_j} = \sum_{j=1}^c x_j\theta_j - s\log\left(\sum_{k=1}^c \exp\theta_k\right),$$

showing that the multinomial distribution is in the multivariate exponential dispersion family with $\psi(\boldsymbol{\theta}) = s\log\left(\sum_{k=1}^c \exp\theta_k\right)$.

A.2 Generalized Linear Models

We start by reviewing 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 $\boldsymbol{x}_i^T$. This is one of the more primitive models in the statistical modeling toolbox and isn't always appropriate for the data.

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

Definition 5 *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*

$$\begin{cases} y_i \overset{\text{ind.}}{\sim} \text{ED}(\mu_i, \varphi) & (13) \\ g(\mu_i) = \eta_i & (14) \\ \eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}. & (15) \end{cases}$$

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.

The canonical link $(b')^{-1}$ is often chosen as the link function, due to its computational and mathematical properties [Agresti, 2015]. Other times, the canonical link is inappropriate and there are alternative default choices.

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.

B Extended Experiments

B.1 Training Details

We train our models on the English text8 corpus[6] 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[7].

The adjustable model configurations in IWLRLS 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 B.4, and set the penalty tuning parameter $\lambda = 0.002$. The estimated word vectors are the rows of $\frac{1}{2}\hat{U} + \frac{1}{2}\hat{V}$.

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[8]. Models are trained for up to 50 epochs (50 passes through the co-occurrence matrix) with batches of size 512. We evaluate the impact of multiple iterations of the IWLRLS algorithm on all models, but examine different additions to the model only when $p = 1.25$. We believe the impact of these changes will be present however for any value of p.

B.2 Experiment 1: Effects of Multiple Iterations

We present results of multiple iterations of our IWLRLS algorithm with different distributions. In particular, we perform multiple iterations of the IWLRLS algorithm with Tweedie distribution and weight truncation to match the GloVe objective function and processing by setting the weight function in our model from $h(x) = x^{2-p}$ to $h(x) = (\min\{x, x_{\max}\})^{.75}$ with $x_{\max} = 10$ and $p = 1.25$. We also presents results for an early-stopped version of skip-gram, and the new Poisson estimator.

The results are summarized in Figure 1. We remark on a few observations based on these results. First, as the number of steps increases, the accuracy on the analogy task increases for the first few iterations. Second, relatively few steps are needed with the accuracy of Tweedie model performing best at the first and second steps of the algorithm, and the Multinomial and model performing best in steps 3-5 but with very similar performance at earlier steps. The Poisson model performs best after 9 iteration, however performs nearly identically to the result of an early stopped algorithm at 3 iterations. In conclusion, we find that early-stopped and one-step versions of the algorithm can perform comparably to full-likelihood methods.

[6] `http://mattmahoney.net/dc/text8.zip`

[7] 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.

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

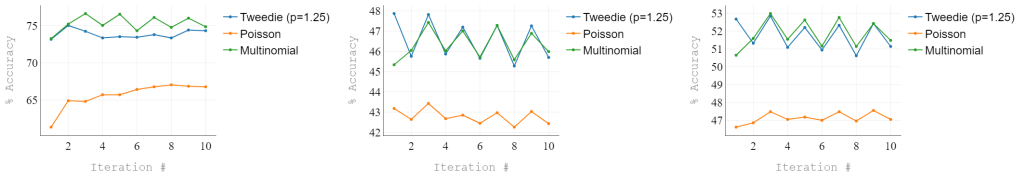

Figure 1: Accuracy **Left**: Semantic Accuracy, **Middle**: Syntactic Accuracy, **Right**: Total Accuracy on Google word analogy task with multiple iterations of the IWLRLS algorithm.

B.3 Experiment 2: Effects of Varying p

In this section, 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
1.1	1	66.7	36.8	42.48
1.1	2	72.74	43.39	48.96
1.25	1	72.9	42.26	48.09
1.25	2	**74.6**	**45.26**	**50.84**
1.5	1	73.0	44.7	50.08
1.5	2	70.78	44.68	49.64
1.75	1	63.95	43.5	47.38
1.75	2	65.81	40.01	44.91
1.9	1	53.2	39.91	42.43
1.9	2	53.02	33.6	37.29

Table 3: Results for multiple choices of p for one and two iterations.

The Results in Table 3 show that values of p which are high perform poorly, while values of p below 1.5 perform similarly. We find that $p = 1.25$ performs the best, and view this value of p as a good choice as it accounts for zero-inflation present in the co-occurence X. This also agrees with the results of [Pennington et al., 2014] and [Kobayashi and Tanaka-Ishii, 2018].

An even more interesting and perhaps more appropriate way to estimate the power p of the Tweedie distribution is in a data-driven and model-based way. This approach is taken in Kobayashi and Tanaka-Ishii [2018]. In future work, we plan to use an improved estimating equation relative to [Kobayashi and Tanaka-Ishii, 2018] to estimate p as part of the algorithm. This would be modeling the marginal distribution of the co-occurences as being Tweedie with the same power. Under a similar assumption, modified likelihood calculations are tractable and so are another possibility. We plan to explore this in future work.

B.4 Experiment 3: Effects of Co-occurrence Matrix and Weight Truncation

We set $p = 1.25$ in our algorithm with Tweedie distribution, and explore the effect of different strategies in handling large entries in the co-occurrence matrix X. One strategy is to simply input X into step 3 of our method. A second strategy is to clamp the weight $h(\cdot)$ that results from step 3 of our method by taking $h(x) = (\min\{x, x_{\max}\})^{.75}$ as in GloVe. A third strategy is to input $\min\{x, x_{\max}\}$ for each entry of the matrix X, where $x_{\max} = 10$, into step 3 of our method[9].

We find that no adjustment to the weights and GloVe's method of weight truncation both perform similarly with weight truncation slightly outperforming no adjustment. We suspect a more significant improvement will show with larger corpora such as a full Wikipedia corpus.

Alternative approaches to alleviating the problem of large co-occurences are to use a more robust distribution or link function. Indeed, the weight truncation in GloVe can be directly mimicked by

[9]The choice of $x_{\max} = 10$ is set according to the default hyperparameters provided in the GLoVe source code available at `https://github.com/stanfordnlp/GloVe` for training on the text8 corpus.

Strategy	Iterations	Semantic	Syntactic	Total
1	1	72.91	42.26	48.09
1	2	**74.6**	45.26	50.84
2	1	70.6	45.51	50.28
2	2	71.94	**45.97**	**50.9**
3	1	54.88	44.01	46.08
3	2	53.82	45.3	46.92

Table 4: Results for multiple choices of regularizing the large values of the co-occurrence matrix. Our strategies are (1) harmonic matrix, (2) truncation of the weight only, (3) truncation of the co-occurrence matrix to $x_{max} = 10$.

either altering the distribution or the link function. The desired form can be found via the weight and pseudo-response equations in algorithm 1. We leave this to future work.

B.5 Experiment 4: Regularization Effects

Strategy	Iterations	Semantic	Syntactic	Total
Penalty	1	69.89	44.32	49.18
Penalty	2	73.98	**46.2**	**51.48**
No Penalty	1	72.91	42.26	48.09
No Penalty	2	**74.6**	45.26	50.84

Table 5: Results for including the penalty term in Equation (16) and not including the diagonal terms.

We consider regularizing the word vectors by including the penalty

$$\frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right), \tag{16}$$

with $\lambda = .002$ for two reasons. One is to reduce noise in the estimation of the word vectors. Udell et al. [2016, Lemma 7.3] show that penalizing by (16) 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 (16) 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 empirically by Levy et al. [2015] that a symmetric distribution of singular values works optimally on semantic tasks.

Finally, we experiment with whether the penalty introduced in Equation (16) improves results and accurately reduces noise in the estimate. We also consider not including the diagonal elements of X 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 linguistic information. Table 5 demonstrates the included regularization within the IWLRLS algorithm with Tweedie distribution and $p = 1.25$ improves results.

B.6 Experiment 5: Bias Term Effects

In Experiment 1, we found that the Multinomial model outperforms the Poisson model, although the Poisson model has an additional bias term to model context word frequencies. This result was fairly counterintuitive, so we additionally experiment with having only a single bias term a_i in the Tweedie model as in the Multinomial model.

We find overall that the Tweedie model with a systematic component without the bias term b_j performs slightly better than the Tweedie model with systematic component containing both bias terms a_i and b_j. We hope to further study the impact of bias terms and other systematic components in future work.

Strategy	Iterations	Semantic	Syntactic	Total
Both Bias	1	73.18	47.87	52.67
Both Bias	2	75.04	45.76	51.32
Both Bias	3	74.25	47.82	52.84
Both Bias	4	73.36	45.87	51.09
Both Bias	5	73.53	47.20	52.20
Single Bias	1	71.4	47.62	52.13
Single Bias	2	73.98	45.64	51.02
Single Bias	3	74.51	**48.20**	**53.20**
Single Bias	4	74.51	46.24	51.61
Single Bias	4	**75.22**	47.87	53.06

Table 6: Results for including the bias term on the context word b_j in addition to a_i.