The strange geometry of skip-gram with negative sampling

David Mimno and **Laure Thompson**
Cornell University
mimno@cornell.edu, laurejt@cs.cornell.edu

Abstract

Despite their ubiquity, word embeddings trained with skip-gram negative sampling (SGNS) remain poorly understood. We find that vector positions are not simply determined by semantic similarity, but rather occupy a narrow cone, diametrically opposed to the context vectors. We show that this geometric concentration depends on the ratio of positive to negative examples, and that it is neither theoretically nor empirically inherent in related embedding algorithms.

1 Introduction

It is generally assumed that the geometry of word embeddings is determined by semantic relatedness. Vectors are assumed to be distributed throughout a K-dimensional space, with specific regions devoted to specific concepts. We find that vectors trained with the skip-gram with negative sampling (SGNS) algorithm (Mikolov et al., 2013) are not only influenced by semantics but are also strongly influenced by the negative sampling objective. In fact, far from spanning the possible space, they exist only in a narrow cone in $\mathbb{R}^K$. Nevertheless, SGNS vectors have become a foundational tool in NLP and perform as well or better than numerous methods with similar objectives (Turian et al., 2010; Dhillon et al., 2012; Pennington et al., 2014; Luo et al., 2015) with respect to evaluations of intrinsic and extrinsic quality (Schnabel et al., 2015).

SGNS works by training two sets of embeddings: the "official" word embeddings and a second set of *context* embeddings, with one K-dimensional vector in each set for each word in the vocabulary. The objective tries to make the word vector and context vector closer for a pair of words that actually occur together than for randomly sampled "negative" words. Following training, the word vectors are typically saved; the context vectors are deleted. Any difference between these two sets of vectors is puzzling, since the sliding window used in training is symmetrical: a word and its context word reverse roles almost immediately. Indeed, the superficially similar GloVe algorithm (Pennington et al., 2014) also defines word and context vectors and by default returns the mean of these two vectors.

Previous work has analyzed what the algorithm might be doing in theory, as an approximation to a matrix factorization (Levy and Goldberg, 2014). Other work has considered the empirical effects of some of the more arbitrary-seeming algorithmic choices (Levy et al., 2015). But we still have relatively little understanding of how the algorithm actually determines parameter values.

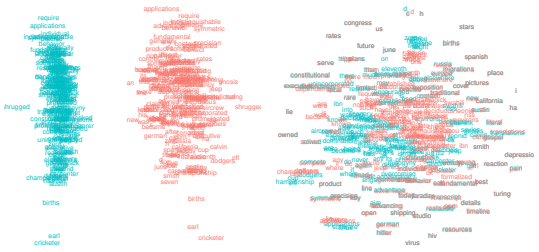

Figure 1: SGNS word vectors and their context vectors projected using PCA (left) and t-SNE (right). t-SNE provides a more readable layout, but masks the divergence between word and context vectors.

In this work we measure geometric properties of SGNS-trained word vectors and their context vectors. Although the word vectors appear to span K-dimensional space, we find that the SGNS objective results in vectors that are narrowly clustered in a single orthant, and can be made non-negative without significant loss. Figure 1 shows two visualizations of SGNS vectors and context vectors. The context vectors mirror the "official" word vectors, with the angle between vectors effectively controlled by the number of negative samples. We

show that this effect is due to negative sampling and not the general embedding objective. We note that this relationship between vectors is effectively hidden by the commonly-used t-SNE projection (van der Maaten and Hinton, 2008).

2 Word embeddings with SGNS

The SGNS algorithm defines two sets of parameters, K-dimensional word vectors $\boldsymbol{w}_i$ and context vectors $\boldsymbol{c}_i$ for each word i. We define a weight between a word i and a context word j as $\sigma_{ij} = \frac{\exp(w_i^T c_j)}{1+\exp(w_i^T c_j)}$. For each observed pair i, j we sample S "negative" context words from a modified unigram distribution $p(w)^{0.75}$. The stochastic gradient update for one parameter w_{ik} is then

$$\frac{d\ell}{dw_{ik}} = (1 - \sigma_{ij})c_{jk} + \sum_{s=1}^{S} -\sigma_{is}c_{sk}, \quad (1)$$

suppressing for clarity a learning rate parameter λ. A symmetrical update is performed for the context word parameters $\boldsymbol{c}_j$ and $\boldsymbol{c}_s$, substituting $\boldsymbol{w}_i$ for $\boldsymbol{c}$. This update has been shown to be equivalent to the gradient of a factorization of a pointwise mutual information matrix (Levy and Goldberg, 2014).

The impact of the update is to push the vector $\boldsymbol{w}_i$ closer to the context vector of the observed context word $\boldsymbol{c}_j$ and away from the context vectors of the negatively sampled words. The amount of change at any given update is dependent on the degree to which the current model predicts the "correct" source of the context word, whether from the real data distribution or the negative sampling distribution. If the model is infinitely certain that the real word is real ($\sigma_{ij} = 1.0$) and the fake words are fake ($\sigma_{is} = 0.0\ \forall s$), it will make no change to the current parameters.

3 Results

We first present a series of empirical observations based on vectors trained from a corpus of Wikipedia articles that is commonly distributed with word embedding implementations.[1] We then evaluate the sensitivity of these properties to different algorithmic parameters. We make no assertion that these are optimal (or even particularly good) vectors, only that they are representative of the properties of the algorithm.

[1] http://mattmahoney.net/dc/text8.zip

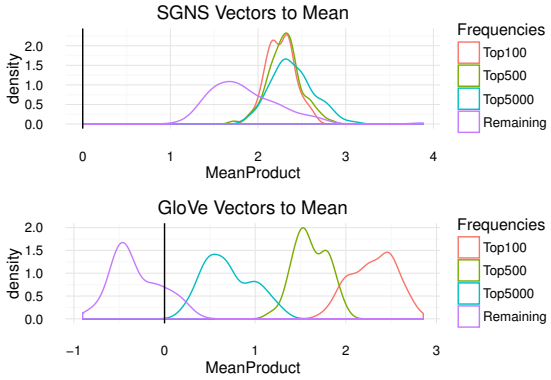

Figure 2: SGNS-trained vectors mostly point in the same direction, towards a mean vector $\hat{\boldsymbol{w}}$.

To determine whether observed properties are due to SGNS specifically or to embeddings in general, we compare SGNS-trained vectors to vectors trained by the GloVe algorithm (Pennington et al., 2014). The choice of GloVe as a comparison is due to its popularity and superficial similarity to SGNS.[2] We begin by examining one set of embeddings from each algorithm, both with $K = 50$ dimensions, a vocabulary of $\approx$ 70k words, and context window 5. We then evaluate sensitivity to negative samples, window size, and dimension.

Embeddings are sensitive to word frequency (Hellrich and Hahn, 2016). Following Zipf's law, words in natural language tend to sort into ranges of frequent words (the majority of tokens) and rare words (the majority of types), with a large class of intermediate-frequency terms in the middle. As a result, the large majority of interactions are between frequent terms or between frequent and infrequent terms. Interactions between infrequent terms are rare, no matter how large the corpus. We define four categories of words by ranked frequency: the top 100 words (ultra-high frequency), the 100–500th ranked words (high frequency), the 500–5000th ranked words (moderate frequency) and the remaining (low frequency) words.

SGNS vectors are arranged along a primary axis. Our first observation is that SGNS-trained vectors all point in roughly the same direction. We can define a mean vector $\bar{w}$ by averaging the vectors of the complete vocabulary $\boldsymbol{w}$. We sample a balanced set of 400 total words with 100 each from the four frequency categories. Figure 2 shows the

[2] We make no attempt in this work to compare the *quality* of SGNS and GloVe vectors, nor should the omission of other algorithms be attributed to anything but space constraints.

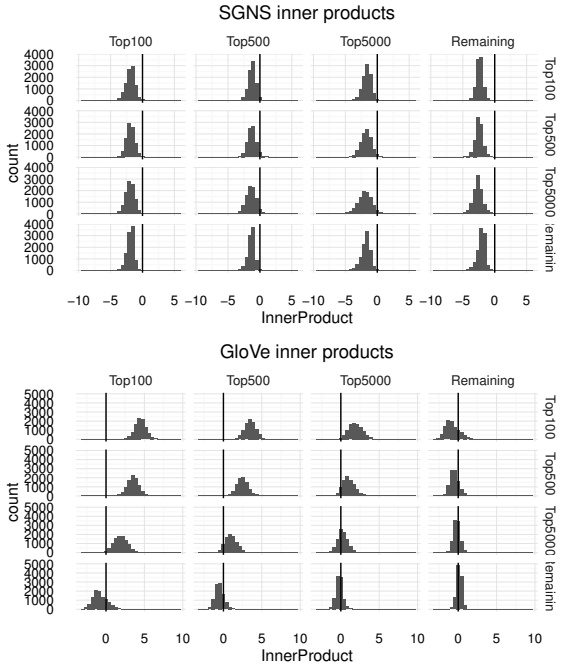

Figure 3: Almost all combinations of words have negative inner products for SGNS, unlike GloVe.

distribution of inner products between these 400 sampled words and their mean vector $\hat{\boldsymbol{w}}$. All vectors have a large, positive inner product with the mean, indicating that they are not evenly dispersed through the space. Furthermore, the frequency category of words has relatively little effect on the inner product, with the exception of the rare words, which have slightly less positive inner products. As a comparison, the vectors trained by GloVe show a clear relationship with word frequency, with low-frequency words opposing the frequency-balanced mean vector.

This result does not depend on a specific mean vector. Using the *global* mean vector rather than the frequency-balanced mean vector reverses the order of frequency categories within each plot, but does not change their overall shape. SGNS vector inner products are all positive, with low-frequency words the most positive. GloVe inner products become positive for low-frequency words and negative for high-frequency words.

The inner product between vectors is used by the algorithm during training, but in practice vectors are often normalized to have unit length before use. It is possible that the apparent pattern shown in Figure 2 may be an artifact of differing average lengths between words of different frequencies. After normalizing SGNS vectors to length 1.0, the lowest and highest frequency words are most similar to the mean vector, with the moderate-frequency words showing the greatest deviation. Normalization does not change the relative order for GloVe vectors.

SGNS vectors point away from context vectors. It is possible that vectors could have a positive inner product with the mean vector but be mutually orthogonal. Figure 3 shows the distribution of inner products $\boldsymbol{w}_i^T \boldsymbol{c}_j$ for pairs of words divided by frequency for SGNS and GloVe. Almost all interactions have similar, negative inner products for SGNS, while GloVe interactions are sensitive to frequency and vary more widely. We note that the high-frequency words in GloVe appear to form a cohesive cluster between themselves (positive inner products) that points away from the lower frequency words (negative inner products), while infrequent words are more dispersed and have no clear pattern relative to each other.

SGNS vectors are mostly non-negative. Not only do SGNS vectors occupy a narrow region of embedding space, it appears that the vectors can be rotated to fall mostly within the positive orthant. For each column of the matrix of vectors $\boldsymbol{w}$ we can compute the dimension-wise mean $\bar{w}_k$. Multiplying $\boldsymbol{w}$ by a diagonal matrix of the signs of the means $diag(sign(\bar{w}_k))$ flips each dimension so that its mean is positive. Figure 4 shows the resulting positive-mean histogram for 12 of the 50 dimensions trained by SGNS (the remaining dimensions are similar). Some dimensions have medians close to 0.0, but most skew positive.

Indeed, it is possible to simply drop all remaining negative values without radically changing the properties of the vectors. Embeddings are often evaluated based on word similarity prediction (Schnabel et al., 2015). Using only positive entries, Spearman rank correlation drops from 0.283 to .276 on the SIMLEX word similarity task and from 0.556 to 0.542 on the MEN task. Subtracting the global mean vector has similarly little impact, reducing SIMLEX correlation to 0.271 and increasing MEN correlation to 0.575. This property may help explain why sparse (Faruqui et al., 2015) and non-negative (Luo et al., 2015) embeddings do not lose significant performance.

SGNS context vectors point away from the word vectors. What then is the geometry of the *context* vectors $\boldsymbol{c}$? The two sets of vectors appear to present a noisy mirror image of each other. Figure 5 shows the distribution of inner products between

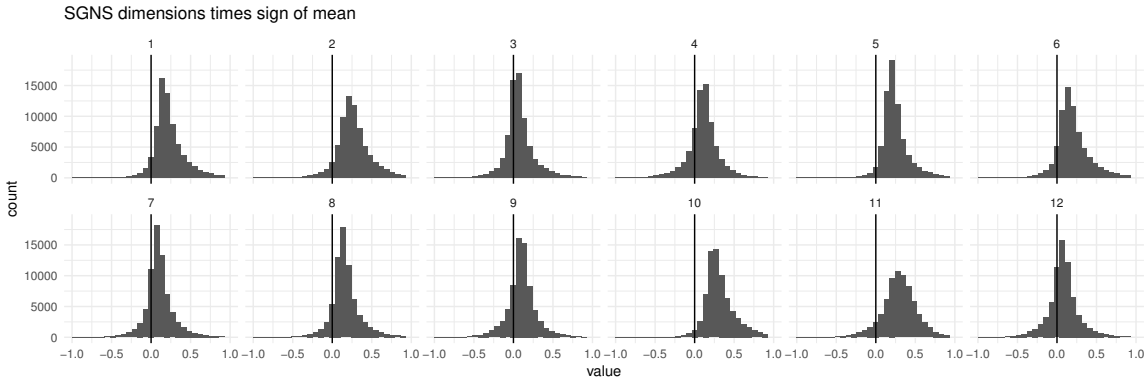

Figure 4: Most latent dimensions show significant skew. Each panel shows a histogram of values for one of the first 12 latent dimensions, after multiplication by the sign of the mean for that dimension.

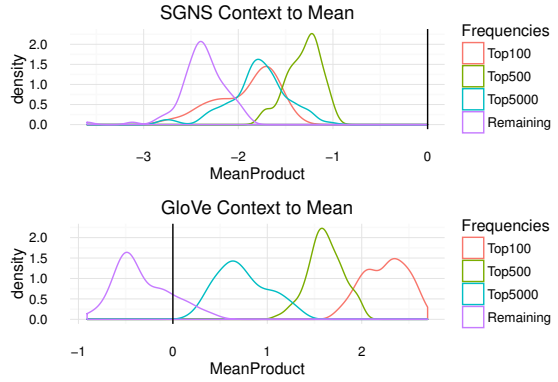

Figure 5: SGNS context vectors point *away* from the mean vector $\hat{\boldsymbol{w}}$, GloVe context vectors do not.

the context vectors and the same mean vector $\hat{\boldsymbol{w}}$ used in Figure 2. These inner products are negative, indicating that the context vectors point in the opposite direction from the word vectors. In contrast, the GloVe context vectors have essentially the same relationship to the mean of the word vectors as the word vectors themselves. This property explains why it is common to output the mean of $\boldsymbol{w}_i$ and $\boldsymbol{c}_i$ for each word for GloVe but not for SGNS: in Glove these two vectors are essentially noisy copies of one another, while in SGNS the two vectors are pointing almost in the opposite direction.

Positive and negative weights come to equilibrium. Eq. 1 balances two terms, a positive interaction term $1.0 - \sigma_{ij}$ between a word and a context word and negative interaction terms $0.0 - \sigma_{is}$ between a word and one of S randomly sampled words. These terms can be viewed as a "label" minus an expectation, as in the gradient for logistic regression. Since there is no $1/S$ term to balance the number of random samples, one might expect that the "power" of the sampled context terms might overwhelm the true interaction term. In practice, these samples appear to find an equilibrium that effectively balances out the number of random samples after a short burn-in phase. We recorded a moving average of positive and negative weights for an ultra-frequent word (*the*) and a moderately frequent word (*tuesday*). In both cases, the mean of the values for positive samples starts at 0.5 and for the negative samples at -0.5. The positive values converge toward $S = 5$ times the mean of the values for negative samples: 0.581 vs. -0.182 for *the* and 0.693 vs. -0.138 for *tuesday*.

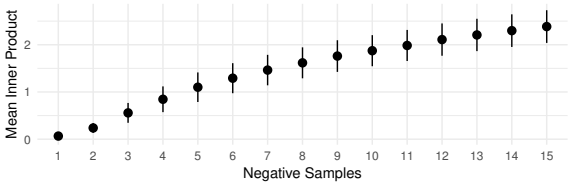

Figure 6: The number of negative samples affects the inner product between vectors and the mean vector. Results are indistinguishable across 10 initializations for each value.

The negative objective is optimized when each model vector points away from the context vectors. The positive objective, in contrast, is maximized when word and context vectors for related words are pointing in the same direction. The negative force acts to repel the vectors, the positive force acts to pull them together.

During the crucial early phases of the algorithm, negative samples have more weight than positive samples: when inner products are near zero, both types of samples will have values of σ_{ij} and σ_{is} close to 0.5, so negative samples will "count" S times more than positive. The early phases of the

algorithm will focus on pushing the two sets of vectors apart into separate regions of the latent space. Once vectors and context vectors separate, inner products will become negative, so σ_{ij} and σ_{is} will move closer to 0.0.

The balance between positive and negative samples consistently affects the geometry of the vectors, and is not sensitive to random initialization. We varied the number of negative samples from $S = 1$ to $S = 15$, and ran 10 trials for each value with different random initializations. As shown in Figure 6, as we increase S, the average inner product between vectors and the mean vector within each model increases.

SGNS vectors are concentrated and point away from their context vectors, and changing the number of negative samples appears to affect this property. We now consider whether other factors could also cause this behavior.

Effect of window size Both SGNS and GloVe operate over word co-occurrences within a sliding window centered around each token in the corpus. This window size parameter has an effect on the semantics of vectors, so it is important to consider whether it has an effect on the geometry of vectors. Simply setting an equal window size for SGNS and GloVe does not, however, guarantee that the two algorithms are seeing equivalent data, because each pair is weighted linearly by token distance in SGNS and by $1/distance$ in GloVe. Figure 7 shows average inner products for each frequency with the global mean vector for 10 trials each at window size 5, 10, 15, 20 with $K = 50$. Increasing window size leads to greater divergence between high- and low-frequency words for word and context vectors, but does not change their pattern. GloVe results are similarly unchanged.

Effect of vector size As with window size, the dimensionality K of the word vectors can affect their ability to represent semantic relationships. Figure 8 shows an increase in inner product with the global mean as we increase K (10 trials each, window size 15), but the effect is small relative to that of the number of negative samples S. GloVe inner products change by less than 0.05.

4 Conclusion

SGNS vectors encode semantic relatedness, but their arrangement is much more strongly influenced by the negative sampling objective than is usually assumed. We find that vectors lie on a narrow primary axis that is effectively non-negative. Users should not interpret relationships between vectors without recognizing this geometric context.

In this work we have deliberately restricted ourselves to describing the geometric properties of vectors. We see several areas for further work. First, there are likely to be theoretical reasons why the observed concentration of SGNS vectors in a narrow cone does not appear to affect performance relative to algorithms that do not have this property. Second, measuring the interplay between positive and negative objectives may provide insight into algorithmic choices that are now poorly understood, such as the effect of reducing the occurrence of frequent words in the corpus and the sampling distribution of negative examples. Finally, we suggest that in addition to theoretical analysis, more work should be done to understand the actual working of algorithms on real data.

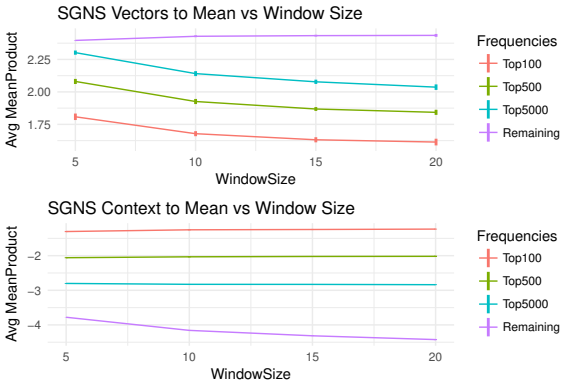

Figure 7: SGNS word and context vectors face in opposite directions regardless of window size.

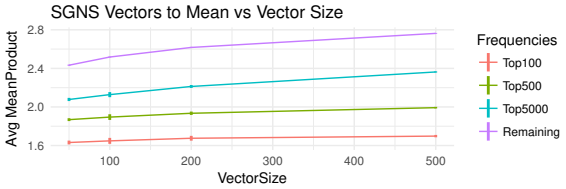

Figure 8: As vector size increases SGNS vectors shift toward the mean vector $\bar{\boldsymbol{w}}$. (GloVe vectors change by < 0.05.)

Acknowledgements

Anonymous reviewers helped make this paper much better. This work was supported by NSF #1526155, #1652536, and #DGE-1144153; and the Alfred P. Sloan Foundation.

References

Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster, and Lyle H. Ungar. 2012. Two step CCA: A new spectral method for estimating vector models of words. In *ICML*, pages 1551–1558.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *ACL*.

Johannes Hellrich and Udo Hahn. 2016. Bad company—neighborhoods in neural embedding spaces considered harmful. In *COLING*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

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

Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *EMNLP*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing high-dimensional data using t-SNE. *JMLR*, 9:2579–2605.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. *HLT-NAACL*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.

Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *EMNLP*, pages 298–307.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394.