Bayesian Paragraph Vectors

Geng Ji[1], **Robert Bamler**[2], **Erik B. Sudderth**[1], and **Stephan Mandt**[2]

[1]Department of Computer Science, UC Irvine, `{gji1, sudderth}@uci.edu`
[2]Disney Research, `firstname.lastname@disneyresearch.com`

Abstract

Word2vec (Mikolov et al., 2013b) has proven to be successful in natural language processing by capturing the semantic relationships between different words. Built on top of single-word embeddings, paragraph vectors (Le and Mikolov, 2014) find fixed-length representations for pieces of text with arbitrary lengths, such as documents, paragraphs, and sentences. In this work, we propose a novel interpretation for neural-network-based paragraph vectors by developing an unsupervised generative model whose maximum likelihood solution corresponds to traditional paragraph vectors. This probabilistic formulation allows us to go beyond point estimates of parameters and to perform Bayesian posterior inference. We find that the entropy of paragraph vectors decreases with the length of documents, and that information about posterior uncertainty improves performance in supervised learning tasks such as sentiment analysis and paraphrase detection.

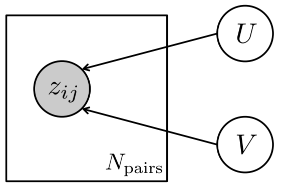
1 Introduction

Paragraph vectors (Le and Mikolov, 2014) are a recent method for embedding pieces of natural language text as fixed-length, real-valued vectors. Extending the word2vec framework (Mikolov et al., 2013b), paragraph vectors are typically presented as neural language models, and compute a single vector representation for each paragraph. Unlike word embeddings, paragraph vectors are not shared across the entire corpus, but are instead local to each paragraph. When interpreted as a latent variable, we expect them to have higher uncertainty when the paragraphs are short.

Recently, Barkan (2017) proposed a probabilistic view of word2vec that has motivated research on combining word2vec with other priors (Bamler and Mandt, 2017). Inspired by this progress, we extend paragraph vectors to a probabilistic model. Our model may be specified via modern inference tools like Edward (Tran et al., 2016), which makes it easy to experiment with different inference algorithms. The experiments in Sec. 4 confirm the intuition that paragraph vectors have higher posterior uncertainty when paragraphs are short, and we show that explicitly modeling this uncertainty improves performance in supervised prediction tasks.

2 Related work

Paragraph embeddings are built on top of word embeddings, a set of dimensionality reduction tools that map words from a large vocabulary to a dense vector representation. Most word embedding methods learn a point estimate for each embedding vector (Mikolov et al., 2013a,b; Mnih and Kavukcuoglu, 2013; Goldberg and Levy, 2014; Pennington et al., 2014). Barkan (2017) pointed out that the skip-gram model with negative sampling, also known as word2vec (Mikolov et al., 2013b), admits a Bayesian interpretation. The Bayesian skip-gram model allows uncertainty to be taken into account in a principled way, and lays the basis for our proposed Bayesian paragraph vector model.



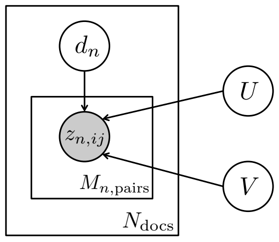


Figure 1: Left: Bayesian word embeddings by Barkan (2017). Right: Bayesian paragraph vectors.

Many tasks in natural language processing require fixed-length features for text passages of variable length, such as sentences, paragraphs, or documents (in this paper, we treat these three terms interchangeably). Generalizing embeddings of single words, several methods have been proposed to find dense vector representations of paragraphs (Le and Mikolov, 2014; Kiros et al., 2015; Wieting et al., 2015; Palangi et al., 2016; Pagliardini et al., 2017). Since paragraph embeddings are local to short pieces of text, we expect them to have high posterior uncertainty if the paragraphs are short. In this work, we incorporate the idea of paragraph vectors (Le and Mikolov, 2014) into the Bayesian skip-gram model in order to coherently infer the uncertainty associated with paragraph vectors.

3 Method

In Sec. 3.1, we summarize the Bayesian skip-gram model on which our model is based. We then present our Bayesian paragraph model in Sec. 3.2, and discuss two inference methods in Sec. 3.3.

3.1 Bayesian skip-gram model

The Bayesian skip-gram model (Barkan, 2017) is a probabilistic interpretation of word2vec (Mikolov et al., 2013b). The left part of Figure 1 shows the generative process. For each word i in the vocabulary, the model draws a latent word embedding vector $U_i \in \mathbb{R}^E$ and a latent context embedding vector $V_i \in \mathbb{R}^E$ from a Gaussian prior $\mathcal{N}(0, \lambda^2 I)$. Here, E is the embedding dimension and λ is a hyperparameter. The model then constructs N_{pairs} labeled pairs of words following a two-step process. First, a proposal pair of words (i, j) is drawn from a uniform distribution over the vocabulary. Then, the model assigns to the proposal pair a binary label $z_{ij} \sim \text{Bern}(\sigma(U_i^\top V_j))$, where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. The pairs with label $z_{ij} = 1$ form the so-called positive examples, and are assumed to correspond to occurrences of the word i in the context of word j somewhere in the corpus. The so-called negative examples with label $z_{ij} = 0$ do not correspond to any observation in the corpus. When training the model, we resort to the heuristics proposed in (Mikolov et al., 2013b) to create artificial evidence for the negative examples (see Section 3.2 below).

3.2 Bayesian paragraph vectors

Bayesian paragraph vectors (BPV) are a direct extension of the Bayesian skip-gram model. The right part of Figure 1 shows the generative process. In addition to global word and context embeddings U and V, the model draws a paragraph vector $d_n \sim \mathcal{N}(0, \phi^2 I)$ for each of the N_{docs} documents in the corpus. Following Le and Mikolov (2014), we add d_n to the context vector V_j when we classify a given pair of words (i, j) as a positive or a negative example. Thus, the likelihood of a word pair (i, j) in document n to have label $z_{n,ij} \in \{0, 1\}$ is

$$p(z_{n,ij} \mid U_i, V_j, d_n) = \sigma\big(U_i^\top (V_j + d_n)\big)^{z_{n,ij}} \, \sigma\big(-U_i^\top (V_j + d_n)\big)^{1-z_{n,ij}}. \tag{1}$$

We collect evidence for the positive examples $\mathcal{X}_n^+$ in each document n by forming pairs of words $(w_{n,t}, w_{n,t+\delta})$. Here, $w_{n,t}$ is the word class of the t^{th} token, t runs over all tokens in document n, δ runs from $-c$ to c where c is a small context window size, and we exclude $\delta = 0$. Negative examples are not observed in the corpus. Following Mikolov et al. (2013b), we construct artificial evidence $\mathcal{X}_n^-$ for negative pairs by sampling from the noise distribution $P(i, j) \propto f(i) f(j)^{\frac{3}{4}}$, where f is the empirical unigram frequency across the training corpus. The log-likelihood of the entire data is thus

$$\log p(\mathcal{X}^+, \mathcal{X}^- | U, V, d) = \sum_n \Bigg[\sum_{(i,j)\in\mathcal{X}_n^+} \log\sigma\big(U_i^\top (V_j + d_n)\big) + \sum_{(i,j)\in\mathcal{X}_n^-} \log\sigma\big(-U_i^\top (V_j + d_n)\big) \Bigg]. \quad (2)$$

In the limit $d_n \to 0$, Eq. (2) reduces to the negative loss function of word2vec. BPV can be easily specified in Edward, a Python library for probabilistic modeling and inference (Tran et al., 2016):

```
from edward.models import Bernoulli, Normal
U = Normal(loc=tf.zeros((W, E), dtype=tf.float32), scale=lam)
V = Normal(loc=tf.zeros((W, E), dtype=tf.float32), scale=lam)
d_n = Normal(loc=tf.zeros(E, dtype=tf.float32), scale=phi)
u_n = tf.nn.embedding_lookup(U, indices_n_I)
v_n = tf.nn.embedding_lookup(V, indices_n_J)
z_n = Bernoulli(logits=tf.reduce_sum(u_n * (v_n + d_n), axis=1))
```

3.3 MAP and black box variational inference

The BPV model has global and local latent variables. We expect the posterior of the global variables to be peaked, and therefore approximate the global word embedding matrices U and V via point estimates. We expect a broader posterior distribution for the local paragraph vectors d_n. Thus we use variational inference (VI) (Blei et al., 2016) to fit the posterior over d_n with a fully factorized Gaussian distribution. We split inference into two stages. In the first stage, we point estimate all parameters. In the second stage, we fix U and V and only perform VI for the paragraph vectors.

In the first stage, our goal is to train the global variables via stochastic gradient descent, where every minibatch contains a single document n and a fixed set of negative examples $\mathcal{X}_n^-$. We first maximize the joint probability $p(\mathcal{X}_n^+, \mathcal{X}_n^-, U, V, d_n)$ w.r.t the paragraph vector d_n. As this local optimization is noise free, it converges quickly under a constant learning rate. Then, we perform a single gradient step for the global variables U and V. This gradient is noisy due to the minibatch sampling and the stochastic generation of negative examples. For this reason, a decreasing learning rate is used. Finally, we reinitialize d_n and proceed to the next document. Optimizing d_n in a nested loop before each update step saves memory since we only need to keep track of the document vectors one at a time.

In the second stage, we fit a variational distribution for the paragraph vectors while holding U and V fixed. We use black box VI (Ranganath et al., 2014) with reparameterization gradients (Kingma and Welling, 2014; Rezende et al., 2014), which is provided by the Edward library. This time, we generate new negative examples in each update step to avoid overfitting. The stochastic optimization is again performed with a decreasing learning rate. We also perform a separate maximum a posteriori (MAP) estimate of the paragraph vectors to serve as the baseline for downstream classification tasks.

4 Experiments

Paragraph vectors are often used as input features for supervised learning in natural language processing (Le and Mikolov, 2014; Kiros et al., 2015; Palangi et al., 2016). In this section, we apply BPV to two binary classification tasks: sentiment analysis and paraphrase detection. We find that the posterior uncertainty of BPV decreases as the length of paragraphs grows. We also find that by concatenating the variational mean and standard deviation features inferred by VI, we improve classification accuracy compared to MAP point estimates of paragraph embeddings.

4.1 Sentiment analysis

We use the IMDB dataset (Maas et al., 2011) for sentiment analysis. It contains 100k movie reviews, split into 75k training points (25k labeled, 50k unlabeled) and 25k labeled test points. Positive and negative labels are balanced in both labeled subsets, and typical reviews consist of several sentences.

As our algorithm is unsupervised, we run the inference algorithms described in Sec. 3.3 using all the training data, and then train a logistic regression classifier using the paragraph vectors of the labeled training data only. We use the most frequent 10k words as the vocabulary, and set the context window size $c = 4$, the embedding dimension $E = 100$, the hyperparameters for the prior $\lambda = \phi = 1$, and the

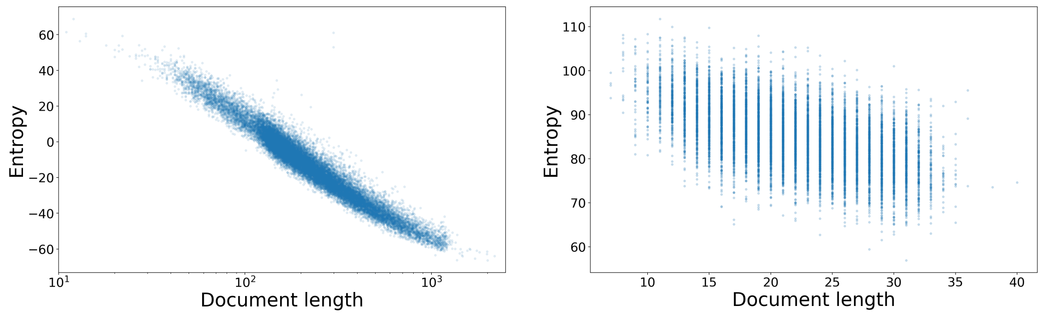


Figure 2: Entropy of paragraph vectors as a function of the number of words in each document. Left: movie reviews in the IMDB dataset. Right: news clips in the MSR dataset. In the left plot, a log scale is used for the horizontal axis to account for the wide range of document lengths.

number of negative examples per document equal to the average number of positive pairs of all the documents. The feature vectors x_n for the classifier are the point estimates of the paragraph vectors d_n for MAP, and the concatenation of the variational mean and standard deviation of d_n for VI.

Table 1 shows the test accuracy of the two inference methods. VI outperforms MAP since it takes into account posterior uncertainty in paragraph embeddings. Fig. 2 (left) shows the entropy of the paragraph vectors, computed using the posterior variance obtained from VI. As the document length grows, the entropy decreases, which makes intuitive sense since longer reviews can be more specific.

Table 1: Classification accuracy of MAP and variational inference

Task (dataset)	MAP	VI
Sentiment analysis (IMDB)	86.88	**86.96**
Paraphrase detection (MSR)	69.97	**70.96**

4.2 Paraphrase detection

We also test the discriminative power of BPV on the Microsoft Research Paraphrase Corpus (Dolan et al., 2004). Each data point contains of two sentences extracted from news sources on the web, and the goal is to predict whether they are paraphrases of each other. The training set contains 4076 sentence pairs in which 2753 are paraphrases, and the test set contains 1725 pairs among which 1147 are paraphrases. We use the same hyperparameters as in the sentiment analysis task, except that we take all the words appearing more than once into the vocabulary because this dataset is much smaller.

After finding the paragraph vectors, we train the classifier by following Kiros et al. (2015), where features are constructed by concatenating the component-wise product $x_n \cdot x_{n'}$ and the absolute difference $|x_n - x_{n'}|$ between each pair of features x_n and $x_{n'}$. The classification results in Table 1 show that VI again outperforms MAP. The relationship between entropy and document length shown in Fig. 2 (right) is also similar to that of the IMDB dataset.

5 Discussion

We proposed Bayesian paragraph vectors, a generative model of paragraph embeddings. We treated the local latent variables of paragraph vectors in a Bayesian way because we expected high uncertainty, especially for short documents. Our experiments confirmed this intuition, and showed that knowledge of the posterior uncertainty improves the performance of downstream supervised tasks.

In addition to MAP and VI, we experimented with Hamiltonian Monte Carlo (HMC) inference, but our preliminary results showed worse performance; we plan to investigate further. A possible reason might be that we had to use a fixed set of negative examples for each document when generating HMC samples, which may result in overfitting to the noise. Finally, we believe that more sophisticated models of document embeddings would also benefit from a Bayesian treatment of the local variables.

References

Bamler, R. and Mandt, S. (2017). Dynamic word embeddings. In *ICML*.

Barkan, O. (2017). Bayesian neural word embedding. In *AAAI*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2016). Variational inference: A review for statisticians. *arXiv:1601.00670*.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *ACL*.

Goldberg, Y. and Levy, O. (2014). Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv:1402.3722*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *ICLR*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *NIPS*.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *ACL*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *NIPS*.

Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*.

Pagliardini, M., Gupta, P., and Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv:1703.02507*.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *TASLP*, 24(4):694–707.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*.

Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *AISTATS*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv:1610.09787*.

Wieting, J., Bansal, M., Gimpel, K., Livescu, K., and Roth, D. (2015). From paraphrase database to compositional paraphrase model and back. *TACL*, 3:345–358.