

# Learning Topic Models by Neighborhood Aggregation

Ryohei Hisano

Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan  
em072010@yahoo.co.jp

## Abstract

Topic models are frequently used in machine learning owing to their high interpretability and modular structure. However, extending a topic model to include a supervisory signal, to incorporate pre-trained word embedding vectors and to include a nonlinear output function is not an easy task because one has to resort to a highly intricate approximate inference procedure. The present paper shows that topic modeling with pre-trained word embedding vectors can be viewed as implementing a neighborhood aggregation algorithm where messages are passed through a network defined over words. From the network view of topic models, nodes correspond to words in a document and edges correspond to either a relationship describing co-occurring words in a document or a relationship describing the same word in the corpus. The network view allows us to extend the model to include supervisory signals, incorporate pre-trained word embedding vectors and include a nonlinear output function in a simple manner. In experiments, we show that our approach outperforms the state-of-the-art supervised Latent Dirichlet Allocation implementation in terms of held-out document classification tasks.

## 1 Introduction

Topic models are widely used in both academia and industry owing to their high interpretability and modular structure [Blei *et al.*, 2003]. The highly interpretable nature of topic models allows one to gain important insights from a large collection of documents while the modular structure allows researchers to bias topic models to reflect additional information, such as supervisory signals [Mcauliffe and Blei, 2008], covariate information [Ramage *et al.*, 2009], time-series information [Park *et al.*, 2015] and pre-trained word embedding vectors [Nguyen *et al.*, 2015].

However, inference in a highly structured graphical model is not an easy task. This hinders practitioners from extending the model to incorporate various information besides text of their own choice. Furthermore, adding a nonlinear output function makes the model even more difficult to train.

The present paper shows that topic modeling with pre-trained word embedding vectors can be viewed as implementing a neighborhood aggregation algorithm [Hamilton *et al.*, 2017] where the messages are passed through a network defined over words. From the network view of topic models, Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] can be thought of as creating a network where nodes correspond to words in a document and edges corresponds to either a relationship describing co-occurring words in a document or a relationship describing the same word in the corpus. The network view makes it clear how a topic label configuration of a word in a document is affected by neighboring words defined over the network and adding supervisory signals amounts to adding new edges to the network. Furthermore, by replacing the message passing operation with a differentiable neural network, as is done for neighborhood aggregation algorithms [Hamilton *et al.*, 2017], we can learn the influence of the pre-trained word embedding vectors to the topic label configurations as well as the effect of the same label relationship, from the supervisory signal.

Our contribution is summarized as follows.

- We show that topic modeling with pre-trained word embedding vectors can be viewed as implementing a neighborhood aggregation algorithm where the messages are passed through a network defined over words.
- By exploiting the network view of topic models, we propose a supervised topic model that has an adaptive message passing where the parameters governing the message passing and the parameters governing the influence of the pre-trained word embedding vectors to the topic label configuration is learned from the supervisory signal.
- Our model includes a nonlinear output function connecting document to their corresponding supervisory signal.
- Our approach outperforms state-of-the-art supervised LDA implementation [Katharopoulos *et al.*, 2016; Zhu *et al.*, 2009] on a wide variety of datasets regarding predictive performance and gives a comparative performance concerning topic coherence.

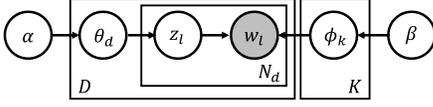


Figure 1: Bayesian network plate diagram representation of LDA

## 2 Notations

We briefly summarize the basic notations used throughout the paper. Let  $1 \leq d \leq D$ ,  $1 \leq w \leq W$ ,  $1 \leq k \leq K$  and  $1 \leq s \leq S$  respectively denote the document index, word index, topic number index and label index. We denote by  $x_{d,w}$  the number of word counts for a particular document–word pair  $(d, w)$ . The task of topic modeling is to assign the average topic label configuration  $z = z_{d,w}^k$  from the observed document word count matrix  $X = \{x_{d,w}\}$  where the average topic label configuration is defined as  $z_{d,w}^k := \frac{\sum_{i=1}^{x_{d,w}} z_{d,w,i}^k}{x_{d,w}}$  for all document–word pairs  $(d, w)$  in the corpus. The average topic label configuration sums to 1 over topic index  $1 \leq k \leq K$  (i.e.,  $\sum_{k=1}^K z_{d,w}^k = 1$ ) and one of the tasks in this paper is to calculate the messages (which we denote  $\mu_{d,w}^k$ ) that can approximate all  $z_{d,w}^k$  in the corpus. We denote by  $\mu_{d,w}^k(t)$  the estimated message at iteration  $t$ . Omission of the subscript  $k$  (i.e.,  $\mu_{d,w}(t)$ ) simply implies the vectorized form of  $\mu_{d,w}^k(t)$  (i.e.,  $[\mu_{d,w}^1(t) \cdots \mu_{d,w}^K(t)]$ ).

We use  $\theta_d$  to denote the topic proportion distribution of document  $d$  and  $\phi_w$  to denote the topic distribution. We also use  $v_i$  to denote node attribute information attached to node  $i$ .

## 3 Background

### 3.1 Factor Graph Approach to LDA

From the perspective of topic modeling, our approach is related to the work of [Zeng *et al.*, 2013], who reframed LDA as a factor graph and used the message passing algorithm for inference and parameter estimation.

The classical Bayesian network plate diagram representation of LDA is presented in Figure 1. The joint distribution of the model can be summarized as

$$p(x, z | \alpha, \beta) = \prod_{d=1}^D \prod_{k=1}^K \frac{\Gamma(\sum_{w=1}^W x_{d,w} z_{d,w}^k + \alpha)}{\Gamma(\sum_{k=1}^K (\sum_{w=1}^W x_{d,w} z_{d,w}^k + \alpha))} \times \prod_{w=1}^W \prod_{k=1}^K \frac{\Gamma(\sum_{d=1}^D x_{d,w} z_{d,w}^k + \beta)}{\Gamma(\sum_{w=1}^W (\sum_{d=1}^D x_{d,w} z_{d,w}^k + \beta))}, \quad (1)$$

where  $\alpha$  and  $\beta$  denote hyperparameters of the Dirichlet prior distribution. Meanwhile, by designing the factor function as

$$f_{\theta_d}(x, z, \alpha) = \prod_{k=1}^K \frac{\Gamma(\sum_{w=1}^W x_{d,w} z_{d,w}^k + \alpha)}{\Gamma(\sum_{k=1}^K (\sum_{w=1}^W x_{d,w} z_{d,w}^k + \alpha))} \quad (2)$$

and

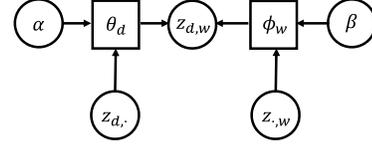


Figure 2: Factor graph representation of LDA

$$f_{\phi_w}(x, z, \beta) = \prod_{k=1}^K \frac{\Gamma(\sum_{d=1}^D x_{d,w} z_{d,w}^k + \beta)}{\Gamma(\sum_{w=1}^W (\sum_{d=1}^D x_{d,w} z_{d,w}^k + \beta))}, \quad (3)$$

a factor graph representation of LDA (i.e., Figure 2) can be summarized as

$$p(x, z | \alpha, \beta) = \prod_{d=1}^D f_{\theta_d}(x, z, \alpha) \prod_{w=1}^W f_{\phi_w}(x, z, \beta). \quad (4)$$

Equation (4) shows that the factor graph representation encodes exactly the same information as Eq. (1). From the factor graph representation, it is possible to reinterpret LDA using a Markov random field framework and thus infer messages for words in a document using loopy belief propagation. The essence of their paper can be summarized by a message updating equation of the form

$$\mu_{d,w}^k(t+1) \propto \frac{\sum_{w'=1, w' \neq w}^W x_{d,w'} \mu_{d,w'}^k(t) + \alpha}{\sum_{k=1}^K (\sum_{w'=1, w' \neq w}^W x_{d,w'} \mu_{d,w'}^k(t) + \alpha)} \times \frac{\sum_{d'=1, d' \neq d}^D x_{d',w} \mu_{d',w}^k(t) + \beta}{\sum_{w=1}^W (\sum_{d'=1, d' \neq d}^D x_{d',w} \mu_{d',w}^k(t) + \beta)}. \quad (5)$$

Our goal in this paper is to connect the factor graph approach of LDA with the neighborhood aggregation algorithm.

### 3.2 Neighborhood Aggregation Algorithm

The goal of node embedding is to represent nodes as low-dimensional vectors summarizing the positions or structural roles of the nodes in a network [Hamilton *et al.*, 2017]. The neighborhood aggregation algorithm is a recently proposed algorithm used in node embedding [Dai *et al.*, 2016; Hamilton *et al.*, 2017], which tries to overcome limitations of the more traditional direct encoding [Perozzi *et al.*, 2014] approaches. The heart of a neighborhood aggregation algorithm lies in designing encoders that summarize information gathered by a node's local neighborhood. In a neighborhood aggregation algorithm, it is easy to incorporate a graph structure into the encoder, leverage node attribute information and add nonlinearity to the model. The parameters defining the encoder can be learned by minimizing the supervised loss [Dai *et al.*, 2016; Hamilton *et al.*, 2017]. In [Dai *et al.*, 2016], it was shown that these algorithms can be seen as replacing

message passing operations with differential neural networks. We refer to [Dai *et al.*, 2016] for a more detailed explanation.

The essence of neighborhood aggregation algorithms is characterized by three steps: aggregation, combination and normalization. In the aggregation step, we gather information from a node’s local neighborhood. A concatenation, sum-based procedure or elementwise mean is usually employed. In the combination step, we combine a node’s attribute information with the gathered information from the aggregation step. After passing the combined information through a nonlinear transformation, we normalize the message so that the new updated messages can be further used by neighboring nodes.

The overall process is succinctly summarized as

$$\mu_i(t+1) = \text{Norm}(\sigma(\text{Comb}(v_i, \text{Agg}(\mu_j(t); \forall j \in \text{Nei}(i)); W_A, W_C))), \quad (6)$$

where  $\mu_i(t+1)$  denotes the message of node  $i$  at iteration  $t+1$ ;  $\text{Nei}(i)$  denotes neighboring nodes of node  $i$ ;  $\text{Agg}$ ,  $\text{Comb}$  and  $\text{Norm}$  respectively denote aggregation, combination and normalization functions;  $W_A$  and  $W_C$  denote parameters used in the combination step (e.g.,  $\text{Comb}(v_i, x_i) = W_C v_i + W_A x_i$ ) and  $\sigma$  is an elementwise nonlinear transformation. With enough update iterations, the model converges and the desired messages are learned.

Suppose that we are given a training dataset  $D = \{X, y_{1:D}\}$ , where  $X$  is the observed network and  $y_{1:D}$  is the supervisory signal attached to each node. For a classification problem  $\{y_d \in 1, \dots, S\}$ , we use a simple neural network with softmax output to transform the learned messages to probabilities and to minimize the cross-entropy loss. This is summarized as

$$\begin{aligned} \text{score}_d &= S_C \sigma(S_B \sigma(S_A \mu_d + T_A) + T_B) + T_C \\ p_d^s &= \frac{\exp(\text{score}_d^s)}{\sum_{s=1}^S \exp(\text{score}_d^s)} \\ \min_{\{W_A, W_C, S_A, S_B\}} & - \sum_{d=1}^D \sum_{s=1}^S y_d^s \log(p_d^s) \end{aligned} \quad (7)$$

where  $S_A, S_B, S_C$  are either  $H_1 \times K, H_2 \times H_1$  and  $S \times H_3$  weight matrices transforming the messages and  $T_A, T_B$  and  $T_C$  are bias vectors of either size  $H_1, H_2$  and  $S$ ,  $y_d^s = 1$  if the label of node  $d$  is  $s$  and zero otherwise and  $\sigma$  is an elementwise nonlinear transformation. We use the classic sigmoid function in this paper.

For a regression problem  $y_n \in R$ , parameters can be learned by minimizing the sum-of-squared loss:

$$\min_{\{W_A, W_C, S_A, S_B\}} \sum_{d=1}^D (y_d - \text{score}_d)^2. \quad (8)$$

## 4 Model

### 4.1 LDA and Neighborhood Aggregation

The message-passing equation (Eq. (5)) of LDA can be seen as taking an elementwise product of two neighborhood aggregation operations and normalizing the result for

probabilistic interpretation. We clarify this point with an illustrative example. Figure 3 shows a hypothetical corpus with three documents “platypus, clinic, ill”, “platypus, Australia, east” and “platypus, eat, shrimp”. Each document–word pair in the corpus has an associated message like the upper-right vector depicted next to the word “ill” in document 1. The red bold lines indicate a relationship describing co-occurring words in a document while the blue dotted lines indicate a relationship describing the same word in the corpus.

Suppose that we want to update the message for the word “platypus” found in document 2. According to the message passing equation (Eq. (5)), this message is updated by the elementwise product of messages gathered from edges describing co-occurring words in document 2 (as denoted by the red bold arrows) and messages gathered from edges describing the same word in the corpus (as denoted by the blue dotted arrows). In fact, assuming that the aggregation function in a document is

$$\text{Agg}_d^k(\mu_j(t); \forall j \in \text{Nei}_d(d, w)) = \sum_{w'=1, w' \neq w}^W x_{d, w'} \mu_{d, w'}^k + \alpha, \quad (9)$$

where  $\text{Nei}_d(d, w)$  denotes all other words in the same document, there are no additional node features to combine and, using  $\text{Agg}_d$  to denote  $[\text{Agg}_d^1 \cdots \text{Agg}_d^K]$ , the neighborhood aggregation operation from co-occurring words in a document can be written as

$$NA_d(t) = \text{Norm}_K(\text{Agg}_d(\mu_j(t); \forall j \in \text{Nei}_d(d, w))), \quad (10)$$

where  $\text{Norm}_K$  is defined to be a normalization function dividing by the sum over topics  $1 \leq k \leq K$ .

Following similar reasoning, it is assumed that the aggregation function for the edges describing the same word in the corpus is

$$\text{Agg}_w^k(\mu_j(t); \forall j \in \text{Nei}_w(d, w)) = \sum_{d'=1, d' \neq d}^D x_{d', w} \mu_{d', w}^k + \beta, \quad (11)$$

where  $\text{Nei}_w(d, w)$  denotes all document–word pairs in the corpus that is the same word as  $w$  (i.e. nodes in the blue dashed square shown in Figure 3) and there are no additional node features to combine (such as pre-trained word embedding vectors described below). The neighborhood aggregation operation for this neighborhood system can be summarized as

$$NA_w(t) = \text{Norm}_W(\text{Agg}_w(\mu_j(t); \forall j \in \text{Nei}_w(d, w))), \quad (12)$$

where  $\text{Norm}_W$  is defined as a normalization function dividing by the sum over words  $1 \leq w \leq W$ .

The message update equation for a document–word pair  $(d, w)$  can thus be summarized by

$$\mu_{d, w}(t+1) = \text{Norm}_K(NA_d(t) \odot NA_w(t)), \quad (13)$$

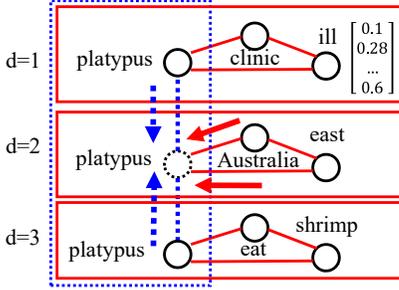


Figure 3: Schematic of our network view of LDA

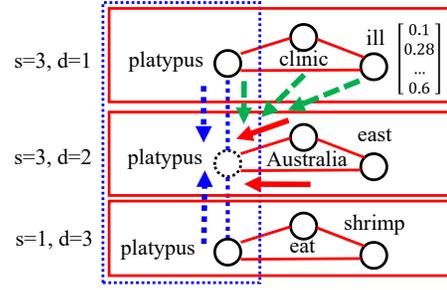


Figure 4: Schematic of our network view of supervised LDA

where  $\odot$  denotes elementwise multiplication. The messages are normalized in the final step so that they can be used as a proper probability distribution.

This explanation shows that the message passing equation (Eq. (5)) of LDA can be seen as an element-wise product of two neighborhood aggregation operations with an additional normalization step for probabilistic interpretation.

## 4.2 Supervised LDA and Neighborhood Aggregation

We extend the above formulation to incorporate supervisory signals. Supervisory signals such as label information can be thought of as defining an additional neighborhood system in the above formulation. We therefore add edges reflecting this additional neighborhood system. For example, in a classification problem where we have label information for each document, we add edges among document-word pairs that belong to other documents with the same label. Figure 4 illustrates the same hypothetical corpus as shown in Figure 3. The green dashed arrows are the new edges added by the supervisory signal.

We define the aggregation function for this neighborhood system as

$$Agg_s^k(\mu_j(t); \forall j \in Nei_s(d, w)) = \sum_{d'=1, d' \neq d}^D \sum_{w=1}^W x_{d', w} \mu_{d', w}^k 1_{l(d')=l(d)}, \quad (14)$$

where the indicator function  $1_{l(d')=l(d)}$  is used to select documents with the same label and  $l$  is a function that maps a document index  $d$  to its corresponding label  $s$ . Neighborhood systems can easily be extended to the regression case where documents do not necessarily have exactly the same output value (e.g., real numbers). In this case, we replace the indicator function as  $1_{|l(d')-l(d)| < \epsilon}$  for a given  $\epsilon > 0$ .

From this formulation, the neighborhood aggregation operation from the supervisory signal can be written as

$$NA_s(t) = Norm_K(W_s Agg_s(\mu_j(t); \forall Nei_s(d, w))), \quad (15)$$

where  $Nei_s(d, w)$  denotes the neighborhood system of  $s$  for document-word pair  $(d, w)$  and  $W_s$  denotes a diagonal matrix with positive entries only. We train  $W_s$  using a supervisory signal as is done in neighborhood aggregation algorithms [Hamilton *et al.*, 2017].

In the standard sum-product algorithm, we take the product of all messages from factors to variables. However as was noted in [Zeng *et al.*, 2013], a product operation cannot control the messages from different sources (i.e. supervised part and unsupervised part) and we, therefore, take the weighted sum of the two neighborhood aggregation operations to rebalance the two effects. The whole message updating equation can be summarized as

$$Norm_K(\eta NA_s(t) + (1 - \eta) NA_d(t)) \odot NA_w(t), \quad (16)$$

where  $\eta$  controls the strength of the supervisory signal in inferring the topic label configuration. This type of rebalancing of the effect from the supervised part and unsupervised is also taken in the traditional supervised LDA approaches where it is well known that the effect of the supervised part is reduced for documents with more words in the standard formulation [Katharopoulos *et al.*, 2016]. Hence, Eq. (16) is quite a standard technique, contrary to what it might appear at first glance.

## 4.3 Pretrained Word Embedding Vectors

It is natural to assume that there is an association between pre-trained word embedding vectors [Mikolov *et al.*, 2013] and topics, so long as topic models are used to summarize the semantic information in a document. Hence, learning the association between pre-trained word embedding vectors and topics is important especially at the time of testing when there might be many unseen words in the held-out documents. If these unobserved words are included in the pre-trained word embedding dictionary and we have trained a function mapping the word embedding vectors to the topics, we can leverage the pre-trained word embedding vectors to predict the topic label configuration more accurately. This issue was addressed in several recent papers [Das *et al.*, 2015].

Within the neighborhood aggregation framework, the pre-trained word embedding vectors can be modeled as a node attribute vector  $v_i$  in the notation of Eq. (6). We define the word embedding in the topic distribution transformation as

$$u_{g(d, w)} = Norm_K(\sigma(W_C v_{g(d, w)})), \quad (17)$$

where  $g$  is a function that maps the document–word index to the word index in the pre-trained word embedding vector dictionary,  $W_C$  is the  $K \times E$  weight matrix transforming word embedding vectors (which we assume to have dimension  $E$ ) to the topic and  $\sigma$  is an element–wise nonlinear transformation.

Using the transformed vector  $u_g(d, w)$ , we define the combine function as

$$Comb(u_g(d, w), NA_w(t), NA_d(t), NA_s(t)) = u_{g(d,w)} \odot (\eta NA_s(t) + (1 - \eta) NA_w(t)) \odot NA_d(t). \quad (18)$$

The entire message updating equation is now summarized as

$$Norm_K(Comb(u_g(d, w), NA_w(t), NA_d(t), NA_s(t)))^{\mu_{d,w}(t+1)} = \quad (19)$$

This update equation is a particular instance of Eq. (6) assuming identity mapping for  $\sigma$  and applying elementwise multiplication for the combine function.

#### 4.4 Training

Inference of the messages (i.e.,  $\mu_{d,w}^k$ ) and estimation of the parameters (i.e.,  $W_S, W_C, S_A, S_B, S_C, T_A, T_B$  and  $T_C$ ) can be performed by alternately inferring the topic label configurations given parameters of the model and updating the parameters minimizing the supervised loss given a topic label configuration. In each iteration, we first infer the topic label configuration for the randomly sampled document–word pairs, then update all the parameters by creating a mini–batch of the document–word pairs and update them using standard stochastic gradient descent technique. The number of the document–word pairs are set to 5000. For the regularization parameter governing  $W_S$  and  $W_C$ , we set it to 0.001, and for the output function, we set a dropout probability of 0.5 for regularization. We also set  $\eta$  in our model to be 0.2 for the economic watcher survey and 0.05 for the rest, and set the number of hidden units in Eq.(7) to be  $H_1 = 50$  and  $H_2 = 50$ . We repeat this iteration until convergence. We could also use sparse message passing in the spirit of [Pal *et al.*, 2006] to enhance the coherence of the topics, but we leave this for future works.

### 5 Experiments

#### 5.1 Datasets

We conduct experiments showing the validity of our approach. We use three datasets in our experiments. Two of our datasets are data of multiclass-label prediction tasks focusing on sentiment (economic assessments and product reviews) while the other dataset is data of a binary label prediction task focusing on subjectivity. We summarize the datasets below.

- The economic watcher survey, in the table abbreviated as EWS, is a dataset provided by the Cabinet Office of Japan<sup>1</sup>. The purpose of the survey is to gather

<sup>1</sup>The whole dataset is available at [http://www5.cao.go.jp/keizai3/watcher\\_index.html](http://www5.cao.go.jp/keizai3/watcher_index.html)

economic assessments from working people in Japan to comprehend region-by-region economic trends in near real time. The survey is conducted every month, and each record in the dataset consists of a multi-class label spanning from 1 (i.e., the economy will get worse) to 5 (i.e., the economy will get better) and text describing the reasons behind a judgment. We use a subset of this dataset describing the assessment of future economic conditions. We use the 3000 most frequently used words in the corpus and further restrict our attention to records that have more than a total of 20 words in the pre-trained word embedding dictionary to perform a fair comparison among the models<sup>2</sup>. We randomly sample 5000 records for training, development, and testing. For the pre-trained word embedding vectors describing Japanese words, we use word2vec vectors provided by [Suzuki *et al.*, 2016]<sup>3</sup>.

- Amazon review data are a dataset of gathered ratings and review information [McAuley *et al.*, 2015]. We use a subset of the five-core apps for android dataset<sup>4</sup> and use the 3000 most frequently used words in the corpus. We randomly sample 5000 records for training, development, and testing focusing on reviews that have more than 20 words excluding stop words in the pre-trained word embedding dictionary<sup>5</sup>. For the pre-trained word embedding vectors describing English words, we use the word2vec embedding vectors provided by Google [Mikolov *et al.*, 2013].
- Subjectivity data are a dataset provided by [Pang and Lee, 2004], who gathered subjective (label 0) and objective (label 1) snippets from Rotten Tomatoes and the Internet Movie Database. We focus on snippets that have more than nine words and sample 1000 snippets each for training, development and testing<sup>6</sup>. Other settings are the same as those of the Amazon review data.

#### 5.2 Classification Performance

The main goal in this section is to compare the classification performance of our proposed models with that of the state-of-the-art supervised LDA implementations (which we denote as SLDA[Katharopoulos *et al.*, 2016] and MedLDA[Zhu *et al.*, 2009]), nonlinear prediction using pre-trained word embedding vectors only (which we denote as WE-MLP) and topic models incorporating pre-trained word embedding vectors (which we denote as LFLDA[Nguyen *et al.*, 2015] and LFDMM[Nguyen *et al.*, 2015]). The difference between LFLDA and LFDMM is that

<sup>2</sup>We use standard morphological analysis software [Kudo *et al.*, 2004] to annotate each word.

<sup>3</sup>These are available for download at [http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki\\_vector/](http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/)

<sup>4</sup>The term five-core implies a subset of the data in which there are at least five reviews for all items and reviewers.

<sup>5</sup>The whole dataset is available at <http://jmcauley.ucsd.edu/data/amazon/>.

<sup>6</sup>The whole dataset is available at <http://ws.cs.cornell.edu/people/pabo/movie-review-data>.

Dataset	EWS		Amazon review		Subjectivity	
	Cross entropy	Accuracy	Cross entropy	Accuracy	Cross-entropy	Accuracy
SLDA	1.541	0.336	2.398	0.415	0.661	0.650
MedLDA	-	0.465	-	0.485	-	0.821
WE-MLP	1.379	0.453	1.575	0.343	0.748	0.656
LFLDA	1.196	0.468	<b>1.222</b>	0.457	0.461	0.799
LFDMM	1.327	0.499	1.299	0.477	0.571	0.825
NA-WE-SLDA	<b>1.194</b>	<b>0.504</b>	1.304	<b>0.510</b>	<b>0.402</b>	<b>0.829</b>

Table 1: Performance of held-out document classification. EWS stands for Economic Watcher Survey.

	EWS	Amazon review	Subjectivity
LDA	0.386	0.318	0.334
SLDA	0.367	0.332	0.339
MedLDA	<b>0.389</b>	0.327	0.338
LFLDA	0.377	<b>0.338</b>	<b>0.358</b>
LFDMM	0.363	0.323	0.323
NA-WE-SLDA	0.374	0.300	0.320

Table 2: Topic coherence measurement using the  $C_V$  measure proposed in [Röder et al., 2015]. EWS stands for Economic Watcher Survey.

LFDMM assumes that all the words in a document share the same topic. We use the default settings defined in the codes provided by the authors. We also fix the number of topics to 20 for all experiments performed in this section. We compare the performance of these models which we denote as NA-WE-SLDA.

For WE-MLP, we use the average of all the pre-trained word embedding vectors found in a document as the feature describing a document and we connect this feature to our supervisory signal using a simple multilayer perceptron with softmax output. For LFLDA and LFDMM we use the document topic distributions as the feature describing a document and also connect this feature to our supervisory signal using a simple multilayer perceptron with softmax output. Parameters (e.g., the number of hidden units) of these models was found by utilizing the development dataset.

Table 1 summarizes the results. Cross-entropy denotes the cross-entropy loss and accuracy denotes the proportion of correct predictions in the held-out documents. For MedLDA we only report accuracy because the method does not use the softmax output function. We see that NA-WE-SLDA is the best performing model in all cases and sometimes beats SLDA and MedLDA substantially, especially for multi-label classification tasks when measuring with accuracy.

### 5.3 Comparison with Deep Learning

Here we compare our model with the state-of-the-art deep learning method [Kim, 2014] varying the number of documents from 1000 to 5000. We chose the filter sizes, number of filters and dropout probability by using development data set. We see that for smaller document size our method performs better than the deep learning method, but as the number of the document increases the power of

No. Docs	CNN		NA-WE-SLDA	
	Cross entropy	Accuracy	Cross entropy	Accuracy
1000	1.265	0.483	<b>1.212</b>	<b>0.498</b>
2000	<b>1.192</b>	0.502	1.217	<b>0.508</b>
3000	<b>1.114</b>	<b>0.546</b>	1.221	0.505
4000	<b>1.092</b>	<b>0.556</b>	1.221	0.513
5000	<b>1.099</b>	<b>0.553</b>	1.194	0.504

Table 3: Comparison with deep learning method for the Economic Watcher Survey.

deep learning catches up, and our model is out-beaten as expected. However, our performance for a smaller number is worth noting.

### 5.4 Topic Coherence

Our goal in this section is to compare topic coherence among the supervised topic models that we evaluated in the previous section and the plain LDA model [Blei et al., 2003]. The purpose of this experiment is that adding complex structures to topic models might have a side effect of downgrading interpretability [Chang et al., 2009]. We want to examine whether our models sacrifice interpretability to achieve better predictive performance.

Traditionally, topic coherence is evaluated on the basis of the perplexity measure, which is basically a measure of how likely a model predicts words in the held-out documents. However, as was shown in [Chang et al., 2009], higher perplexity does not necessarily correlate to better human interpretability and various topic coherence measures have been proposed to address this issue [Röder et al., 2015]. The present paper uses the  $C_V$  measure proposed in [Röder et al., 2015], which is a coherence measure that combines existing basic coherence measures. It was shown that the  $C_V$  measure is the best coherence measure among 237,912 coherence measures compared in their paper [Röder et al., 2015]. We employ the source code provided by the authors<sup>7</sup> to calculate the coherence measure in our experiment focusing on the top 15 words in the topic distribution<sup>8</sup>.

Table 2 summarizes the results. We see that our method has a price to pay for its better predictive accuracy. For all the data set, our model shows inferior coherence compared to other methods. Our second observation is that except

<sup>7</sup>The code is available at <https://github.com/dice-group/Palmetto>

<sup>8</sup>For the economy watcher survey, we translated words from Japanese to English.

for the economic watcher survey dataset and our model, models with word embedding vectors slightly outperforms the model without word embedding vectors. Our model even utilizing word embedding vectors, could not compensate for the loss made from the nonlinear output function and lack of sparsification, but still provides comparable performance.

To gain further insights into our learned topics, in Table 4, Table 5 and Table 6, we report the top six words of the 20 topics learned by LDA, SLDA, and NA-WE-SLDA respectively for the economic watcher survey dataset<sup>9</sup>. First of all we see several topics that does not appear in LDA such as “ecology, payment, automobile, subsidy, sale, car” (i.e., Topic 1) in SLDA and “ecology, subsidy, car, payment, last, year” (i.e., Topic 8) in NA-WE-SLDA highlighting the effect of the supervisory signals. It is also worth mentioning that although the topic corresponding to the great east Japan earthquake does seem to appear in LDA as in Topic 7 (i.e., “customer, last, year, situation, disaster, east”), the topic corresponding to the earthquake event is easier to spot in SLDA and NA-WE-SLDA, where the former focuses more on the impact on the car industry (i.e., Topic 13) and the latter focuses more on the nuclear incident (i.e., Topic 9). The difference between SLDA and NA-WE-SLDA could be seen in topics that focus on temporary dispatch workers<sup>10</sup>. While Topic 15 in SLDA seems to focus on dispatch workers in small and medium-sized companies, Topic 16 in NA-WE-SLDA focuses more on the end of the fiscal year which corresponds to the timing when dispatch workers are more susceptible to layoffs.

## 6 Conclusion

We showed that topic modeling with word embedding can be viewed as implementing a neighborhood aggregation algorithm where the messages are passed through a network defined over words. By exploiting the network view of topic models, we proposed new ways to model and infer supervised topic models equipped with a nonlinear output function. Our extensive experiments performed over a range of datasets showed the validity of our approach.

## Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP17K17663. We thank Glenn Pennycook, MSc, from Edanz Group ([www.edanzediting.com/ac](http://www.edanzediting.com/ac)) for editing a draft of this manuscript.

<sup>9</sup>Stop words are omitted.

<sup>10</sup>Temporary workers sent by staffing agencies.

Topic number	Top 6 words
Topic 1	company, economy, like, gain, production, job
Topic 2	person, job, hunting, situation, company, increase
Topic 3	customer, good, goods, situation, sales, continue
Topic 4	economy, company, expect, feel, continue, many
Topic 5	consumption, tax, increase, goods, customer, life
Topic 6	price, product, increase, unit, situation, impact
Topic 7	customer, last, year, situation, disaster, east
Topic 8	good, economy, months, number, company, customer
Topic 9	consumption, tax, increase, last, minute, demand
Topic 10	number, last, year, sales, visitor, future
Topic 11	customer, store, sales, increase, tourism, people
Topic 12	economy, company, yen, weaker, consumption, stronger
Topic 13	last, year, travel, percent, ratio, reservation
Topic 14	economy, look, demand, person, number, like
Topic 15	job, offer, number, last, year, tendency
Topic 16	industry, production, impact, job, offer, none
Topic 17	consumption, last, year, situation, ecology, impact
Topic 18	consumption, situation, customer, economy, continue, travel
Topic 19	months, 3, 2, good, situation, ahead
Topic 20	consumption, economy, company, recover, business, gain

Table 4: Topics learned by LDA.

Topic number	Top 6 words
Topic 1	ecology, payment, automobile, subsidy, sale, car
Topic 2	expect, increase, possible, sales, movement, target
Topic 3	consumption, tax, increase, demand, last, minute
Topic 4	good, customer, gain, economy, expect, little
Topic 5	industry, order, production, job, fiscal, year
Topic 6	Tokyo, city, sign, good, Olympic, go
Topic 7	goods, sales, product, unit, price, customer
Topic 8	good, economy, none, situation, change, customer
Topic 9	price, fee, store, soaring, continue, increase
Topic 10	yen, price, stronger, economy, impact, weaker
Topic 11	people, tourism, customer, foreign, use, expect
Topic 12	economy, customer, bad, consumption, situation, like
Topic 13	east, Japan, disaster, impact, car, commerce
Topic 14	reservation, travel, last, year, situation, customer
Topic 15	company, hire, small, medium, sized, dipatch
Topic 16	consumption, tax, increase, demand, last, minute
Topic 17	month, last, year, 10, 12, sales
Topic 18	months, 3, 2, ahead, number, situation
Topic 19	yen, economy, consumption, stronger, weaker, impact
Topic 20	job, offer, number, last, year, percent

Table 5: Topics learned by SLDA.

Topic number	Top 6 words
Topic 1	situation, economy, number, future, good, change
Topic 2	good, USA, change, expect, continue, economy
Topic 3	consumption, good, economy, customer, situation, expect
Topic 4	economy, situation, consumption, customer, company, person
Topic 5	last, year, good, number, month, customer
Topic 6	customer, good, economy, future, consumption, situation
Topic 7	consumption, customer, situation, economy, good, none
Topic 8	ecology, subsidy, car, payment, last, year
Topic 9	nuclear, Fukushima, one, place, power, generation
Topic 10	last, year, 2, ratio, 3, number
Topic 11	consumption, situation, customer, economy, company, continue
Topic 12	job, offer, last, year, months, number
Topic 13	consumption, customer, situation, impact, economy, continue
Topic 14	consumption, tax, increase, last, year, situation
Topic 15	industry, production, company, continue, economy, impact
Topic 16	fiscal, year, dispatch, worker, person, end
Topic 17	consumption, good, economy, customer, expect, company
Topic 18	consumption, tax, increase, demand, last, minute
Topic 19	last, year, ratio, month, percent, number
Topic 20	economy, consumption, situation, company, impact, continue

Table 6: Topics learned by NA-WE-SLDA.

## References

- [Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [Chang *et al.*, 2009] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.
- [Dai *et al.*, 2016] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML’16, pages 2702–2711, 2016.
- [Das *et al.*, 2015] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *ACL (1)*, pages 795–804. The Association for Computer Linguistics, 2015.
- [Hamilton *et al.*, 2017] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2017.
- [Katharopoulos *et al.*, 2016] Angelos Katharopoulos, Despoina Paschalidou, Christos Diou, and Anastasios Delopoulos. Fast supervised lda for discovering micro-events in large-scale video datasets. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM ’16, pages 332–336, New York, NY, USA, 2016. ACM.
- [Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar; A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.
- [Kudo *et al.*, 2004] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *In Proc. of EMNLP*, pages 230–237, 2004.
- [McAuley *et al.*, 2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’15, pages 43–52. ACM, 2015.
- [Mcauliffe and Blei, 2008] Jon D. Mcauliffe and David M. Blei. Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Inc., 2008.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 2013.
- [Nguyen *et al.*, 2015] Dat Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3, 2015.
- [Pal *et al.*, 2006] Chris Pal, Charles Sutton, and Andrew McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, 2006.
- [Pang and Lee, 2004] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, 2004.
- [Park *et al.*, 2015] Sungrae Park, Wonsung Lee, and Il-Chul Moon. Supervised dynamic topic models for associative topic extraction with a numerical time series. In *Proceedings of the 2015 Workshop on Topic Models*, TM ’15, pages 49–54, 2015.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, pages 701–710, 2014.
- [Ramage *et al.*, 2009] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP ’09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [Röder *et al.*, 2015] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM ’15, pages 399–408, 2015.
- [Suzuki *et al.*, 2016] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoki Okazaki, and Kentaro Inui. Wikipedia kiji ni taisuru kakucho koyuu hyougen label no tajyuu fuyo [multiple granting of extended named expression labels for wikipedia articles]. In *The Association for Natural Language Processing*, 2016.
- [Zeng *et al.*, 2013] Jia Zeng, William K. Cheung, and Jiming Liu. Learning topic models by belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1121–1134, May 2013.
- [Zhu *et al.*, 2009] Jun Zhu, Amr Ahmed, and Eric P. Xing. Medlda: Maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML ’09, pages 1257–1264, New York, NY, USA, 2009. ACM.