Joint Aspect-Sentiment N-Gram Topic Model for Customer Reviews

Anonymous Author(s)
Affiliation
Address
email

Abstract

With the increase in popularity of e-commerce, more and more customer reviews are available online. These reviews are very useful for customers to make buying decisions, and also helpful for business to improve their products based on these reviews. However, with the large amount of reviews exists for each product today, it's usually hard to go through each of them. In this paper, we propose N-Gram based topic models that can automate these process. Specifically, the unsupervised model can jointly learn the aspects and sentiments from the customer reviews. While the unigram LDA model is effective, simple and computationally efficient, it does not capture some meaningful phrase from aspect, like "Pa Thai", "working distance",etc. On the other hand, our model is able to capture more meaning phrases, and also capture the sentiment distribution for each aspect. Our experimental results showed that the proposed model works better than state-of-arts models in terms aspect mining and sentiment analysis.

1 Introduction

Nowadays, there are large amount of reviews for products available online, ranging from books, restaurants, electronic devices to many others. In the review, customer will give some opinions for different aspects. For example, people who rate restaurant may provide some opinions for food, service, price, location and etc, which we call *aspects*. And finally, they will provide a rating based on their overall sanctification.

Different customers may talk about the different aspects for the same product. A user who is looking for computer may care about its computational power and maximum throughput, while other cares more about graphical capability. Another big problem is understanding how the sentiments for different aspects are expressed. The sentence, "The computer is great for designers" shows the positive sentiment for the graphical capability, while the sentence, "It takes forever to get the results" shows the negative sentiment for the computational power. One simple solution is to find all the positive and negative words from the sentence, and classify them based on these words. Till now, there have been lots of work done for sentiment classification using supervised learning[11,12]. But this approach typically requires a lot of labeled data, which is expensive to obtain in practice.

As we can see, there are two big problems for review mining, aspect extraction and sentiment classification. Most of the existing methods[] use two-step approach: firstly, they build a model to automatically extract the aspects from the reviews. Then for each aspect, they look for sentiment words that express those aspects.

In this paper, we propose a new topic model that can jointly model aspects and sentiments. More specifically, we extend the N-gram topic model to handle the sentiment as well. In addition to this, we learn the aspect for each sentence level, not on word level. This assumption is reasonable for review mining, since each sentence is likely related to only one aspect. The comparison between N-gram model and sentence-based N-gram model will be shown in later section.

2 Previous Work

There have been quite a lot of efforts put into aspect mining and sentiment classification. For sentiment classification, Turney and Littman[13] uses an unsupervised learning algorithm to classify the semantic orientation in the word/phrase level based on mutual information, and Choi et al[14] uses conditional random fields and a variation of Autoslog to classify the sentiments. Figure 1-(b) shows the google product reviews. Where large amount of reviews are automatically classified by different aspects[15]. However, these aspects are predefined for different product types.

Comparing to the sentiment classification task, extracting aspects from the reviews seems more challenging. The earliest attempts for detecting aspects are based on frequently occurring noun phrases [9]. This approach works well when aspect are strongly tied to the single word, but less useful when aspects uses many low frequency terms. One common solution is to use clustering techniques to group the terms that associated with the same aspects. After that, they search for opinions associated with those aspects.

With the popularity of topic model(LDA)[10], there are many variations of LDA model have been applied to the product review mining. Titov et al.[7] propose a model for modeling two types of topics in reviews: global topic and local topics. The global topics correspond to a global property of review such as brand, and local topics corresponding to the product aspects. This is because aspects are fundamentally different from the global property of products. Brody and Elhadad [8] proposed a local LDA model, where they extract the aspects from sentence instead of the whole review text. Wang et al [1] uses bootstrap methods to extract the aspects first, and then use latent models to analyze the opinions. All these methods focus on extracting aspects, or dealing with the sentiment analysis as a separate step.

While most of these work focus either on aspect mining or sentiment analysis, there are few of recent attempts that jointly model aspects and sentiments. The work that is closely related to ours is []. They also argued that training for sentence level performs better than training on word level. However, all of their approaches are based on bag of words assumption that use unigram models. While unigram model is simple and computationally efficient, it lack of capability to find out meaningful phrase which consists of more than one word.

3 Preliminary

Let's first take a look at N-gram topic model. as shown in Figure 1, which was proposed by []. While the bi-gram model always generate bi-grams, the N-gram model is flexible enough that can generate phrase that consists of arbitrary length of words. The main idea is to introduce new latent variable X, which is the indicator variable to denotes whether we need to combine the current word with previous one to make bi-gram. The whole generative process looks like this:

1. Draw Discrete distribution ϕ_k from Dirichlet prior β, for each topic k.
2. Draw Bernoulli distribution ψ_{kw} from a Beta prior γ for each topic k and each word w
3. Draw Discrete distribution σ_{kw} from Dirichlet prior η for each topic k and each word w.
4. For each document d, for $d = 1, ...D$
   (a) draw document distribution θ_d from Dir(α)
   (b) for each word i in document d
      i. Draw z_i from multinomial(θ_d)
      ii. Draw indicator variable x_i from Bernoulli $\psi_{z_{i-1},w_{i-1}}$
      iii. Draw word w_i from ϕ_{z_i} if $x = 0$, else draw w_i from $\sigma_{z_i,w_{i-1}}$

As we notice, we draw each word either from unigram model or bi-gram model, depends on the indicator variable. This is very intuitive and makes model to generate any meaningful phrases. The derivation of gibbs sampling for N-gram topic model is straightforward, although it's a little bit complicated that uni-gram LDA model.

We first assume that

$$c(\alpha) = \frac{\sum_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$$

For collapsed gibbs sampling we just need to calculate the following quantity:

$$p(z_i, x_i | z_{-i}, x_{-i}, w, \alpha, \beta, \gamma, \delta) = \frac{p(z, x, w|\alpha, \beta, \gamma, \delta)}{p(z_{-i}, x_{-i}, w|\alpha, \beta, \gamma, \delta)} \tag{1}$$

For the diving term, we can expand it to

$$p(z, x, w|\alpha, \beta, \gamma, \delta) = p(z|\alpha)p(w|z, x, \beta, \delta)p(x|z, w, \gamma)$$

The first term $p(z|\alpha)$ remains the same as in LDA. We have

$$p(z|\alpha) = \prod_{d=1}^{D} \frac{c(\alpha + \sum_{j:\delta_j = d} I_K(z_j))}{c(\alpha)}$$

for $p(w|z, \beta)$, we need to consider two different cases.

When $x_i = 0$

$$p(w|z, \beta) = \int p(w|z, \phi)p(\phi|\beta)d\phi$$

$$= \int \prod_{k=1}^{K} \prod_{i:z_i=k, x_i=0} \prod_{v=1}^{V} \phi_{k,v}^{I_{w_i=v}} \times \prod_{k=1}^{K} \frac{1}{c(\beta)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} d\phi$$

$$= \int \prod_{k=1}^{K} \frac{1}{c(\beta)} \prod_{v=1}^{V} \phi_{k,v}^{\sum_{i:z_i=k,x_i=0} I_{w_i=v} + \beta_v - 1} d\phi$$

$$= \prod_{k=1}^{K} \frac{c(\beta + \sum_{i:z_i=k, x_i=0} I_V(w_i))}{c(\beta)}$$

When $x_i = 1$

$$p(w|z, \beta) = \int p(w|z, \sigma)p(\sigma|\beta)d\sigma$$

$$= \int \prod_{k=1}^{K} \prod_{w=0}^{V} \prod_{i:z_i=k, w_{i-1}=w, x_i=1} \prod_{v=1}^{V} \phi_{k,w,v}^{I_{w_i=v}} \times \prod_{k=1}^{K} \prod_{w=0}^{V} \frac{1}{c(\beta)} \prod_{v=1}^{V} \sigma_{k,w,v}^{\delta_v - 1} d\sigma$$

$$= \prod_{k=1}^{K} \prod_{w=0}^{V} \frac{c(\delta + \sum_{i:z_i=k, w_{i-1}=w, x_i=1} I_V(w_i))}{c(\delta)}$$

For the last term, we have

$$p(x|z, w, \gamma = \int p(x|z, w, \psi))p(\psi|\gamma)d\psi$$

$$= \int \prod_{k=1}^{K} \prod_{w=0}^{V} \prod_{i:z_i=k, w_{i-1}=w} \prod_{s=0}^{1} \psi_{k,w,s}^{I_{x_i=s}} \times \prod_{k=1}^{K} \prod_{w=0}^{V} \prod_{s=0}^{1} \psi_{k,w,s}^{\gamma_s - 1}$$

$$= \prod_{k=1}^{K} \prod_{w=0}^{V} \frac{c(\gamma + \sum_{i:z_{i-1}=k, w_{i-1}=w} I_S(x_i))}{c(\gamma)}$$

Finally we can get

$$p(z,x,w|\alpha,\beta,\gamma,\delta) = \prod_{d=1}^{D} \frac{c(\alpha + \sum_{j:\delta_j=d} I_K(z_j))}{c(\alpha)} \times \prod_{k=1}^{K} \frac{c(\beta + \sum_{i:z_i=k,x_i=0} I_V(w_i))}{c(\beta)} \times$$
$$\prod_{k=1}^{K}\prod_{w=0}^{V} \frac{c(\delta + \sum_{i:z_i=k,w_{i-1}=w,x_i=1} I_V(w_i))}{c(\delta)} \times \prod_{k=1}^{K}\prod_{w=0}^{V} \frac{c(\gamma + \sum_{i:z_{i-1}=k,w_{i-1}=w} I_S(x_i))}{c(\gamma)}$$

4 Proposed Models

4.1 Sentence N-gram Model

Unfortunately, topic model does not work well for short text, like reviews, comments and etc. If we apply the classical unigram or N-gram model to short text like reviews, it tends to give us higher level aspects. The problem was mentioned in [(paper from columbia), (paper from KAIST)] as well. The first model we proposed in this paper is sentence n-gram model, where we assume that all the words within one sentence comes from same aspect. We split the text into sentences based on conjunction symbols.

The Figure 1.b shows the sentence N-gram topic model. The generative process is as follows:

1. Draw Discrete distribution ϕ_k from Dirichlet prior β, for each topic k.
2. Draw Bernoulli distribution ψ_{kw} from a Beta prior γ for each topic k and each word w
3. Draw Discrete distribution σ_{kw} from Dirichlet prior η for each topic k and each word w.
4. For each document d, for $d = 1, ...D$
   (a) draw document distribution θ_d from Dir(α)
   (b) for each sentence m in document d
      i. Draw z_m from multinomial(θ_d)
      ii. For each word w_i in sentence m
         A. Draw indicator variable x_i from Bernoulli $\psi_{z_m,w_{i-1}}$
         B. Draw word w_i from ϕ_{z_m} if $x = 0$, else draw w_i from $\sigma_{z_m,w_{i-1}}$

4.2 Aspect-Sentiment N-gram Topic Model

Next,we introduce our proposed model, where we incorporate sentiment information. Besides all of the existing variables, we also draw sentiment distribution for each document. By adding sentiment, we add another layer for each hyperparameters. The whole generative process is as follows:

1. Draw Discrete distribution π from Dirichlet prior $lamda$, for each sentiment type.
2. Draw Discrete distribution $\phi_{s,k}$ from Dirichlet prior β for each sentiment s, and each topic k
3. Draw Bernoulli distribution $\psi_{s,k,w}$ from a Beta prior γ for each sentiment s, and each topic k and each word w
4. Draw Discrete distribution $\sigma_{s,k,w}$ from Dirichlet prior η for each sentiment s, and topic k and each word w.
5. For each document d, for $d = 1, ...D$
   (a) draw document distribution $\theta_{d,s}$ from Dir(α), for each sentiment s
   (b) for each word i in document d
      i. Draw sentiment l_i from multinomial(π_d)
      ii. Draw z_i from multinomial(θ_{d,l_i})
      iii. Draw indicator variable x_i from Bernoulli $\psi_{z_{i-1},w_{i-1},l_{i-1}}$
      iv. Draw word w_i from ϕ_{l_i,z_i} if $x = 0$, else draw w_i from $\sigma_{l_i,z_i,w_{i-1}}$

4.3 Sentence Aspect-Sentiment N-gram Topic Model

1. Draw Discrete distribution π from Dirichlet prior $lamda$, for each sentiment type.
2. Draw Discrete distribution $\phi_{s,k}$ from Dirichlet prior β for each sentiment s, and each topic k
3. Draw Bernoulli distribution $\psi_{s,k,w}$ from a Beta prior γ for each sentiment s, and each topic k and each word w
4. Draw Discrete distribution $\sigma_{s,k,w}$ from Dirichlet prior η for each sentiment s, and topic k and each word w.
5. For each document d, for $d = 1, ...D$
   (a) draw document distribution $\theta_{d,s}$ from Dir(α), for each sentiment s
   (b) for each sentence i in document d
      i. Draw sentiment l_i from multinomial(π_d)
      ii. Draw z_i from multinomial(θ_{d,l_i})
      iii. Draw indicator variable x_i from Bernoulli $\psi_{z_{i-1},w_{i-1},l_{i-1}}$
      iv. For each word w_j in sentence i.
         A. Draw word w_j from ϕ_{l_i,z_i} if $x = 0$, else draw w_i from $\sigma_{l_i,z_i,w_{i-1}}$

5 Experiments

In this section, we give some experimental results based on real-word data sets. We use yelp review data sets, it contains 27000 customer reviews for restaurants. The data set comes from the paper[], and we use it for comparison purpose.

5.1 Preprocessing

Stop words removal + stemming words + split into sentences (”.”,”?”,”!”)

5.2 Aspect Mining

5.3 Sentiment Classification

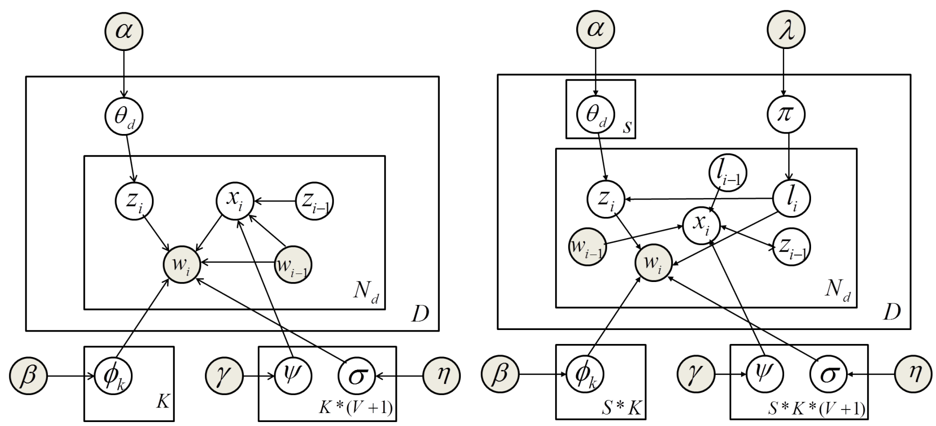
5.4 Perplexity Score

In order to measure the goodness of the model, we need some measurement tool. Perplexity score is widely used in language modeling to assess the predictive power of the model. Since the documents in the corpora are treated as unlabeled, we will do density estimation, and we hope to get the high likelihood for the test data set. In particular, we use the perplexity score, which is monotonically decreasing in the likelihood of the test data, and is equivalent to the inverse of the geometric mean per-word likelihood [10]. More formally, for the test documents set,

$$\text{perplexity}(D_{test}) = exp\{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\} \tag{2}$$

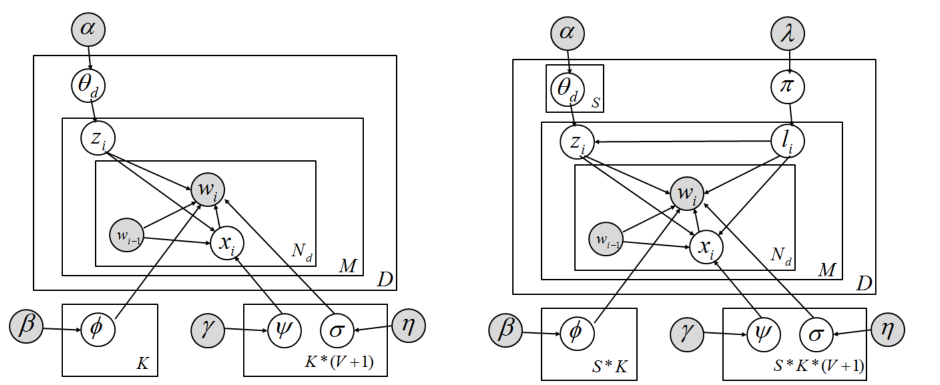
In our experiment, we randomly select the 90% of the data as training samples, while the remaining 10% for testing samples.

Upcoming soon



(a) N-Gram Topic Model

(b) Joint Sentiment-Aspect N-Gram Topic Model



(c) Sentence based N-Gram Topic Model

(d) Sentence Based Joint Sentiment-Aspect N-Gram Topic Model

Figure 1: Four different variations of N-Gram topic models. (a). N-gram topic model (b) Joint Sentiment N-Gram topic model, where we integrate sentiment layout to the model. (c) Sentence N-Gram topic model, where we assume the words in the single sentence draws from the same topic. (d). Sentence joint apsect-sentiment topic model, where we add sentiment layout to the sentence N-gram model

References

[1] Wang.H, Lu. Y and Zhai.C. Latent aspect rating analysis on review text data: a rating regression approach. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp.783–792 Washington, DC, USA 2010

[2] Jo.Y and Oh.A.H. Aspect and sentiment unification model for online review analysis. *Proceedings of the fourth ACM international conference on Web search and data mining*. pp. 815–824, Hong Kong, China 2011.

[3] Zhao.W, Jiang.J, Yan.H.F and Li.X. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. pp. 56–65. Cambridge, Massachusetts, 2010.

[4] Li.F, Huang.M and Zhu.X. Sentiment Analysis with Global Topics and Local Dependency *AAAI*. 2010.

[5] He.Y, Lin.C and Alani.H. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* pp.123–131. Portland, Oregon, 2011

[6] Lin.C and He.Y. Joint sentiment/topic model for sentiment analysis. *Proceedings of the 18th ACM conference on Information and knowledge management*. pp.375–384, Hong Kong, China 2009

[7] Titov.I and McDonald.R. Modeling online reviews with multi-grain topic models. *Proceedings of the 17th international conference on World Wide Web* pp. 111–120 Beijing, China 2008

[8] Brody.S and Elhadad.N. An unsupervised aspect-sentiment model for online reviews *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. pp.804–812. Los Angeles, California. 2010.

[9] Hu.M and Liu.B Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 168–177, Seattle. WA 2004

[10] David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research* pp. 993-1022 2003

[11] Pang.B, Lee.L and Vaithyanathan.S Thumbs up?: sentiment classification using machine learning techniques *Proceeding EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing* - Volume 10 Pages 79-86

[12] Kim.S.M and Hovy.E. Determining the sentiment of opinions. *Proceeding COLING '04 Proceedings of the 20th international conference on Computational Linguistics*. Article No. 1367 .

[13] P.D. Turney and M.L.Littman. Unsupervised learning for semantic orientation from a hundred-billion-word corupus. *CoRR*, cs.LG/0212012, 2012

[14] Y.Choi, C.Cardie, E.riloff, and S.Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* pages 355-362, Vancouver, British Columbia, Canada, October 2005.

[15] Sasha Blair-goldensohn and Tyler Neylon and Kerry Hannan and George A. Reis and Ryan Mcdonald and Jeff Reynar Building a sentiment summarizer for local service reviews. *In NLP in the Information Explosion Era*. 2008

[16] Gregor Heinrich. Infinite LDA CImplementing the HDP with minimum code complexity. *Technical note* TN2011/1, 2011

[17] Tom Griffiths. Gibbs sampling in the generative model of Latent Dirichlet Allocation. *Note*. 2007.

[18] http://www.yelp.com/dataset_challenge/

[19] M. Hoffman, D. Blei, J. Paisley, and C. Wang. Stochastic variational inference. *Journal of Machine Learning Research* 2013