Classification of Yelp Review with Latent Dirichlet Allocation Model

Group 15

Dec 2018

Leader: Yifan Zhang (yifan8)
Members: Xingyu Fu(xingyuf2), Haonan Wang(haonan3), Yixiang Yu(yyu50), Qiaoge Zhu(qzhu18)

ABSTRACT

In many different fields we are faced with a ton of information and we need algorithmic tools to organize, search, and understand this information. The central goal of our project is to provide a "thematic summary" of a collection of yelp reviews. In this project, we intend to use LDA in classifying yelp review's business category. We apply LDA to classify Yelp reviews into 20 categories (topics), such as Restaurant, Bar, and so on. We intend to explore the applications of the methods in computational statistics for our LDA model. We use random number generating methods such as Accept - Reject and sampling methods such as MCMC to help generate the latent topics; in EM part, we use optimization methods to achieve convergence. As the result, we find that we can get good and reasonable results as about 82% using LDA by setting category number as 20, while we could only get about 57% result by setting topic number as 50. It means that LDA is a great model as long as we alternate it based on the real data scenario.

INTRODUCTION

LDA(Figure 1) is a generative statistical model for discovering the abstract topics that occur in a collection of documents. Each document is viewed as a mixture of a number of topics and each document has a set of topics that are assigned to it through LDA. The goal of the LDA model is to decode these topics behind these texts. There are various applications of LDA, such as independent component analysis and information retrieval. In this project, we identified our interest and intended to explore the applications of the methods we learned in class to LDA in classifying texts. We apply LDA to classify Yelp reviews into 20 categories, such as Restaurant, Bar and so on. For LDA, We use random number generations and MCMC to generate the latent topics. Topic allocation, among other usage of LDA models are of immense importance in many applications. Therefore, a detailed study of LDA model is in need.

GOALS: We intend to implement and apply LDA in the application of Yelp reviews business classification. Each Yelp review has multiple prior distributions in its business's properties, such as Restaurant, Beauty and Spa, Bar and so on. Thus it has massive conjugate hidden values and is hard to classify using other methods. Yelp review business classification has a broad usage in various application and research directions such as sentiment analysis and business recommendation, as well as web-based applications such as search engines.

DATASET: The Yelp dataset collects about 5,996,996 Yelp reviews from 1,518,169 users, nearly evenly distributed into different business type, e.g. restaurant, Beauty Salon. Some of the reviews are correlated while others are not. We intend to use this data set for text classification. The data set is published at www.yelp.com/dataset. We tend to sample a subset from the whole yelp dataset as our data here.

METHODS IMPLEMENTATIONS

Method from Group 1:

- Random number generation.

Methods from Group 2:

- Optimization (Calculate critical points from gradient for convergence) in EM Algorithm.
- MCMC.

LDA

LDA (Latent Dirichlet Allocation) model is a generative topic bag of words model that automatically discovers topics in text documents classify text documents. Generally speaking, it is the Bayesian version of PLSA where topic coverage and topic word distributions can be inferred using Bayesian inference. What's special in the LDA method than in the normal PLSA method is we will use a surrogate distribution to approximate the posterior and find the best surrogate distribution from a certain parametric family by minimizing the KL-divergence from the surrogate distribution to the true posterior distribution, which transforms inference into a deterministic optimization.

LDA is the most popular topic model in application. For this project we will employ LDA to classify yelp reviews to 20 categories.

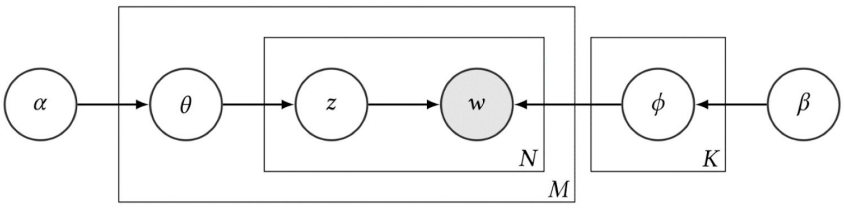


Figure 1

LDA describes the latent distribution of a word in a dataset. In our project, a dataset is a collection of D reviews. But what is a review? It's a collection of words. So our LDA model describes how each review obtains its words. Initially, let's assume we know K review categories

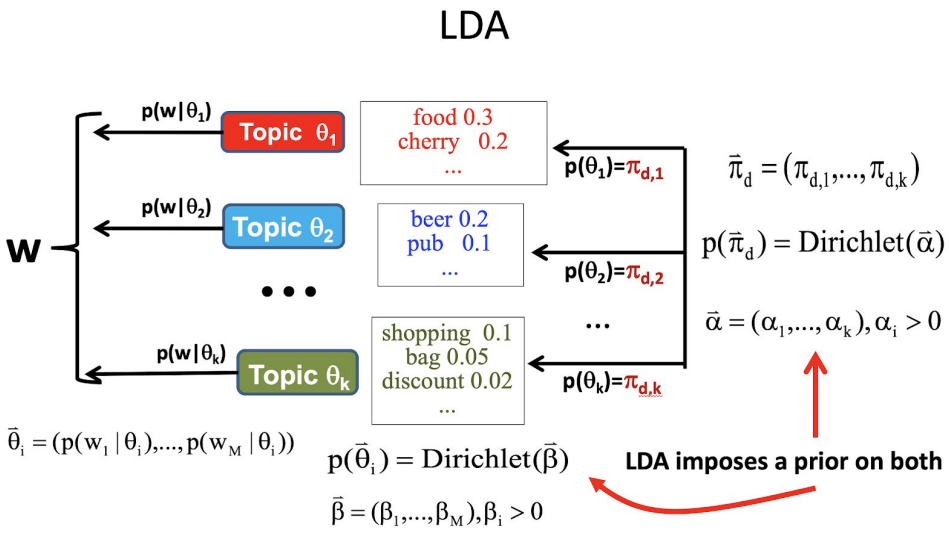
distributions for our dataset, meaning K multinomials containing V elements, where V is the number of terms in our corpus. Let β_i represent the multinomial for the i-th topic, where the size of β_i is V : $|\beta_i| = V$.

To formalize LDA, let's first restate the generative process in more detail

1. For each document:
    (a) draw a topic distribution, $\theta_d \sim \text{Dir}(\alpha)$, where Dir(·) is a draw from a uniform Dirichlet distribution with scaling parameter α
    (b) for each word in the document:
        (i) Draw a specific topic $z_{d,n} \sim \text{multi}(\theta_d)$ where multi(·) is a multinomial
        (ii) Draw a word $w_{d,n} \sim \beta_{z_{d,n}}$

For reference purposes, let's formalize some notation before moving on:

1. *w* represents a word and w^v represents the vth word in the vocabulary where $w^v = 1$ and $w^u = 0$ if the v≠ u—this superscript notation will be used with other variables as well
2. **w** represents a document (a vector of words) where $\mathbf{w} = (w_1, w_2, \ldots, w_N)$
3. α is the parameter of the Dirichlet distribution, technically $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_k)$, but unless otherwise noted, all elements of α will be the same, and so in the included R code α is simply a number.
4. **z** represents a vector of topics, where if the ith element of **z** is 1 then *w* draws from the ith topic
5. β is a $k \times V$ word-probability matrix for each topic (row) and each term (column), where $\beta_{ij} = p(w_j = 1 | z_i = 1)$



Probabilities and Optimization:

$$p_d(w|\{\theta_j\},\{\pi_{d,j}\}) = \sum_{j=1}^{k} \pi_{d,j} p(w|\theta_j)$$

$$\log p(d|\vec{\alpha},\{\theta_j\}) = \int \sum_{w\in V} c(w,d) \log[\sum_{j=1}^{k} \pi_{d,j} p(w|\theta_j)] p(\vec{\pi}_d|\vec{\alpha}) d\vec{\pi}_d$$

$$\log p(C|\vec{\alpha},\vec{\beta}) = \int \sum_{d\in C} \log p(d|\vec{\alpha},\{\theta_j\}) \prod_{j=1}^{k} p(\theta_j|\vec{\beta}) d\theta_1 ... d\theta_k$$

$$(\hat{\alpha},\hat{\beta}) = argmax_{\hat{\alpha},\hat{\beta}} \log p(C|\hat{\alpha},\hat{\beta})$$

Since we randomly generated P(Topic) at first, and we'd like to maximize the log likelihood including P(w|Topic) for each w -- word, and each topic. We need to use EM to find the converging value for both variables on the designated yelp corpus to get the maximum likelihood.

EM/Optimization

EM_Algorithm{

Initialize probability that document d is in topic j = p(PI d, j), and probability that word w occurs under topic j = p(w|j) with random values.

Iteratively improve them using E-step & M-step until likelihood doesn't change{

E step: calculate probability that w in document d is generated from topic j = p(Z d,w = j) based on p(PI d, j), p(w|j)

M step: Re-calculate p(PI d, j) and p(w|j) based on updated p(Z d,w = j).

}

}

EM Algorithm for LDA: E-Step

Hidden Variable (=topic indicator): $z_{d,w} \in \{B, 1, 2, ..., k\}$

Probability that **w in doc d** is generated from **topic** θ_j

Use of Bayes Rule

$$p(z_{d,w} = j) = \frac{\pi_{d,j}^{(n)} p^{(n)}(w|\theta_j)}{\sum_{j'=1}^{k} \pi_{d,j'}^{(n)} p^{(n)}(w|\theta_{j'})}$$

$$p(z_{d,w} = B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k} \pi_{d,j}^{(n)} p^{(n)}(w|\theta_j)}$$

Probability that **w in doc d** is generated from **background** θ_B

EM Algorithm for LDA: M-Step

Hidden Variable (=topic indicator): $z_{d,w} \in \{B, 1, 2, ..., k\}$

Re-estimated **probability** of **doc d** covering **topic** θ_j

ML Estimate based on "allocated" word counts to topic θ_j

$$\pi_{d,j}^{(n+1)} = \frac{\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{j'}\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j')}$$

$$p^{(n+1)}(w|\theta_j) = \frac{\sum_{d\in C} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{w'\in V}\sum_{d\in C} c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)}$$

Re-estimated **probability** of **word w** for **topic** θ_j

RNG

Random number generation is a device that can generate a sequence of random numbers with no patterns and cannot be reasonably predicted. Random number generation is very useful in developing Monte Carlo method simulations, cryptography and so on.

The widely used Accept-Reject (AR) method is described as the following:
AR algorithm to generate an i.i.d. sample of size N from density function f.

1. Choose a density function g that is easy to sample from such that f <= Cg on the support of f for some constant C
2. While the generated sample size is less than N, do:
    a. Sample y ~ g(Y)
    b. Sample u ~ Unif(0, 1)
    c. If u < f(y)/(Cg(y)), then accept y to the sample; else reject y and go to a-step

It can be shown that the generated sample is i.i.d. g(Y), and the acceptance rate is 1/C.

In LDA model, we use **AR method** to initialize the initial background probability and topic distribution, which is ranged between 0 and 1. We intend to use the random number generation to randomly generate numbers between 0 and 1. Suppose there are N topics in total. We first generate background probability P_B from random number generator, and then set the initial distribution for each topic to a Dirichlet prior, which again generated from a RNG. However, we found that a randomized P_B actually does not help a lot in fast convergence and allocation accuracy. In practice, we finally dropped RNG step and initialized the probability manually.

RESULTS

Business Category number	5	20	50
Precision on Sampled data by manual evaluation	91%	82%	57%
Human Judgement	Category Too Broad	Accurate Categories & Good Precision	Precision Too Low

*Due to lack of data and time, we evaluated manually by sampling from training data and check if the lda model predicts correctly. Each person of our group evaluated about 30 reviews from the training data. When we check if lda's output is consistent with the real category, we let the score be 1 if we think they're consistent, 0 if not, and 0.5 if we're not sure. Then we take the average score of all reviews as the precision.

DISCUSSION

In practice, we first made business category number very small as 5, and we also made topic number big as 50. When business category number is 5, we could achieve high precision as about 91% by manual evaluation. However, we think the business category number should be far beyond 5 considering the business category list extracted, so the business categories here would be too general in practice. When we set it 50, we could barely achieve good result and precision is about 57%. Finally, when we use 20 business categories, it is about right, the model achieves accuracy of 82% -- the model seems to assign each review into a properly designed category. Therefore, we realized that LDA model has limitations - we need to try a few times and manually find a good topic number as input to the LDA model that fits with our data. Also, we noticed that the number of business categories in the dataset are based on some distribution by sampling which is subjective and thus doesn't always highlight the true distribution of business categories.

APPENDIX

A. Code

EM optimization

```
    def train(self):
        print (self.topic_K, "Topics, Lambda =", self.lambda_B, ",", self.n_process, "processes")
        print ("Start training ...")

        stdout.flush()
        ll = self.likelihood_multi()
        assert (ll!=0), "Initial likelihood is zero"
        print ("Initial likelihood: " + str(ll))
        self.ll.append(ll)

        for iteration in range(1, self.max_iter+1):
            t_start = time.time()
            self.E_step_multi()
            self.M_step()
            rel_ll = self.rel_likelihood()
            self.rel_ll.append(rel_ll)
            t_end = time.time()
            t = t_end - t_start
            print("Iter #" + str(iteration) + "\tLL " + str(self.ll[-1]) + "\tRel_LL " + str(rel_ll) \
+ "\ttime " + str(t))
            stdout.flush()

            self.check()

            if rel_ll < self.min_rel_ll:
                print("Training stop! Relative likelihood:", rel_ll)
                break

        self.plot_ll(rel=False)
        self.plot_ll(rel=True)
        self.topic_words()
        self.save_model()
```

Randomized initializer

```
def init_random(dim1, dim2, norm=1):
    rand = numpy.random.random([dim1,dim2])
    if norm == 1:
        for row in range(dim1):
            rand[row, :] /= sum(rand[row, :])
    else:
        for col in range(dim2):
            rand[:,col] /= sum(rand[:, col])
    return rand
```

B. Original Data Example


```
{"review_id":"x7mDIiDB3jEiPGPHOmDzyw","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"iCQpiavjjPzJ5_3gPD5Ebg","stars":2,"date":"2011-02-25","text":"The pizza was okay. Not the be$
{"review_id":"dDl8zu1vWPdKGihJrwQbpw","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"pomGBqfbxcqPv14c3XH-ZQ","stars":5,"date":"2012-11-13","text":"I love this place! My fiance An
{"review_id":"LZp4UX5zK3e-c5ZGSeo3kA","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"jtQARsP6P-LbkyjbO1qNGg","stars":1,"date":"2014-10-23","text":"Terrible. Dry corn bread. Rib t
{"review_id":"Er4NBWCmCD4nM8_p1GRdow","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"elqbBhBfElMNSrjFqW3now","stars":2,"date":"2011-02-25","text":"Back in 2005-2007 this place wa
{"review_id":"jsDu6QEJHbwP2Blom1PLCA","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"Ums3gaP2qM3W1XcA5r6SsQ","stars":5,"date":"2014-09-05","text":"Delicious healthy food. The ste
{"review_id":"pfavA0hr3nyqO61oupj-lA","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"vgfcTvK81oD4r50NMjU2Ag","stars":1,"date":"2011-02-25","text":"This place sucks. The customer
{"review_id":"brokEno2n7s4vrwmmUdr9w","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"AxeQEz3-s9_1TyIo-G7UQw","stars":5,"date":"2011-10-10","text":"If you like Thai food, you have
{"review_id":"kUZWBVZvhWuC8TWUg5AYyA","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"zdE82PiD6wquvjYLyhOJNA","stars":5,"date":"2012-04-18","text":"AMAZING!!!\n\n I was referred h
{"review_id":"wcqt0III88LEcm19IxFFyA","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"EAwh1OmG6t6p3nRaZOW_AA","stars":4,"date":"2011-02-25","text":"Ribs = amazing\n2 hour wait tim
{"review_id":"LWUtqzNthMM3vpWZIFBlPw","user_id":"msQe1u7Z_XuqjGoqhB0J5g","business_id":"atVh8viqTj-sqDJ35tAYVg","stars":2,"date":"2012-11-09","text":"Food is pretty good, not gonna
{"review_id":"STiFMww2z31siPY7BWNC2g","user_id":"TlvV-xJhmh7LCwJYXkV-cg","business_id":"yFumR3CWzpfvTH2FCthvVw","stars":5,"date":"2016-06-15","text":"I have been an Emerald Club mem
```


C. Extracted business categories from whole yelp dataset

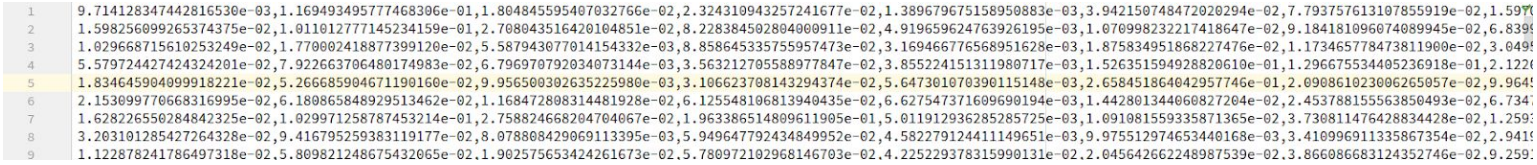
1. **Restaurants**
2. **Shopping**
3. **Beauty & Spas**
4. **Bars**
5. **Home Services**
6. **Health & Medical**
7. **Local Services**
8. **Automotive**
9. **Event Planning & Services**
10. **Active Life**
11. **Coffee & Tea**
12. **Hair Salons**
13. **Home & Garden**
14. **Auto Repair**
15. **Hotels & Travel**
16. **Arts & Entertainment**
17. **Professional Services**
18. **Doctors**

19. **Real Estate**
20. **Pets**
21. Hair Removal
22. Bakeries
23. Dentists
24. Grocery
25. Skin Care
26. Cafes
27. Education
28. Desserts
29. Financial Services
30. Contractors
31. Women's Clothing
32. Pet Services
33. General Dentistry
34. Massage
35. Ice Cream & Frozen Yogurt
36. Hotels
37. Japanese
38. Chicken Wings
39. Day Spas
40. Seafood
41. Salad
42. Apartments
43. Waxing
44. Gyms
45. Caterers
46. Tires
47. Sushi Bars

***** We noticed that the first 20 business categories could properly represent all business categories in the dataset, and the following business categories are more or less subset of the first 20 business categories.

D. Output

First 9 rows



E. CASE STUDY (Example Output)

Category -- Review:

Restaurant "... if you want a great atmosphere no attitude mixed crowd fantastic service and reasonably priced food bev taste phoenix's newest cherry and tell me what you think ..."

LDA Outputs:

1. [1.597065138605689993e-01,9.71412834744281653e-03,1.16949349577746830e-01,1.80484559540703276e-02,2.32431094325724167e-02,1.38967967515895088e-03,3.94215074847202029e-02,7.79375761310785591e-02,2.81932855827616733e-03,2.28386760962764542e-03,2.67154984093204367e-03,5.70774958894796485e-02,1.27900771297067311e-01,5.46600619016533468e-03,8.81656749013220580e-02,4.51813518568008887e-02,1.31942092017348411e-01,3.89109363746727859e-02,4.82952397636299060e-02,2.87536523731875964e-03]

   First Column is Restaurant. -- Correct

REFERENCES

[1] http://times.cs.uiuc.edu/course/510f18/schedule.html
[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:9931022, 2003.
[3] Chase. Geigle. The EM Algorithm: An Optimization View. Sep 30, 2016, http://times.cs.uiuc.edu/course/510f18/notes/em-algorithm.pdf.
[4] LDA Implementation from Sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html