Gaussian generative model based Topic Detection using Factor Analysis

Qian Chen[1,2] , Zhiguo Gui[1,3,4,*] , Xin Guo[2] , Yang Xiang[5]

1 School of Information and Communication Engineering, North University of China, Taiyuan, Shanxi,030051, China
2 School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, 030006, China
3 Key Laboratory of Instrumentation Science and Dynamic Measurement, North University of China, Taiyuan, Shanxi, 030051, China
4 National Key laboratory for Electronic Measurement Technology, Taiyuan,030051, Shanxi, China
5 School of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China
** Corresponding author: gui_zg@163.com*

Abstract
Topic detection is one of the center tasks in information retrieval especially public opinion monitoring. As large scaled documents updating every minute in web2.0, topic detection should be done in second level. However, traditional probabilistic generative model using Gibbs sampling cost long time to reach convergence, and is inappropriate for online documents. In this paper, we presented a probabilistic generative model based on Gaussian assumption using factor analysis for large document collections, and an EM algorithm for factor analysis based topic detection as lower dimensional decomposition was designed to gain faster convergence. Empirical results showed that our algorithm performed detection task in much fewer seconds compared to LDA in both research literature and newsgroup, and the topic quality generated by our model also outperformed LDA in both subjective and objective view.
Keywords: TOPIC DETECTION, GAUSSIAN GENERATIVE MODEL, FACTOR ANALYSIS, EM

1. Introduction

With the development of web2.0, for many areas including science, industry, and culture, Understanding and navigating large online collections of documents has become an important activity in many fields[1]. Topic detection, which is a fully unsupervised problem with no prior knowledge of category structure, is one of the key tasks in text mining, information retrieval especially pulic opnion monitoring. With topic one can get direct intuition about a corpus or an article[2], and with which a large enterprise can manage its own documents, such as news article, research literature, microblog[3], twitter[4], and email[5]. From 2003, many topic models such as LDA[6], one of the most classical probabilistic generative models in the field of text mining, Online LDA[7] et.al. were emerging, and it is becoming the hotspot of research in the last one decade even in image process[17]. However, topic model often uses Gibbs sampling, and the complexity is rather high. Experimental result shows that a topic model with corpus size of 3000, and 10000 words cost almost 3

minutes. For many problems in real time situation, this can be hardly to apply into practice.

We image such situation that an author named mike is trying to compose a paper for some problems using some skills or tools, and some aspect such as motivation, problem to be solved, technique or tools, all of which can be regarded as topics. We see that topics formed the basic elements in a paper. In this paper, we present a novel generative model based on latent factor analysis using Gaussian assumption. Traditional topic models or algorithms are basically mixture model, which treat document as a mixture of topic space, and topic as a mixture of word space[7]. One problem with these mixture models is that they use a single latent variable to generate the document. That is to say, each observation can only come from one of K clusters, thus the model is limited in its representational power. We choose factor analysis as our model since the latent variable in FA is a vector of real-valued numbers, thus can deal with multiple latent variable mapping problem. Besides, for Gaussian mixture model, we face singular problems when we have data in which the size of samples is less or much less than that of features. The maximum likelihood estimates of the mean and covariance for Gaussian mixture may be poor, however, this is quite common in the situation of text mining, and especially there are millions of words in a given vocabulary. Therefore, we choose FA to compose our topic detection algorithm.

This paper is arranged as follows: In section 2, we review related work on topic detection; we introduced the basic theory of FA in section 3,; a novel topic detection approaches based on FA (FA-TD) is proposed in section 3; we analyze the time complexity of FA-TD in section 4, and the experiment is designed and performed in section 5; A conclusion was drawn in section 6.

2. Related Work

From 1980s, Topic detection and tracking dated back to TDT project by Allan[8], and much work has been done on text categorization, which is a supervised problem, and should be labeled artificially beforehand, with the assignment of texts into a given set of categories, and thus it cost much in labor power and time. Some classifier algorithms in machine learning are used to train a model on that set of manually categorized documents. Later, lots of work has been focused on text clustering[9-10], which has already found practical applications, and the core drawback of clustering is that there was no semantic interpretation about each cluster. H. Li proposed a topic analysis method based on finite mixture model[11]. Wartena performed topic detection by clustering words, and overcame the shortcoming of topic representation loss[12]. Later, many generative models such as LDA were boosting and dominating in the last one decade due to its solid mathematical theory and semantic representation, however, it cost much time for model's complexity.

3. Methodology

We review factor analysis, focusing on the classical model using maximization likelyhood extimation.

FA is a Bayesian generative model in which each data point x is generated by sampling a latent multivariate Gaussian variable z, and later projecting to a k-dimensional vector by linear transformation $\mu + \Lambda z$. Finally, the i-th sample $x^{(i)}$ is obtained by a Gaussian noise with covariance Ψ. In our problem, we define the EM algorithm which can be applied to topic detection in section 4. In common setting, we usually imagine that we have sufficient data to be able to discern the multiple-Gaussian structure in the data. However, this would not be the case that the dimension n of the data was significantly larger than training set size m.

Our generative process is described as follows[14]:

$$\begin{aligned} &z \sim N(0, I) \\ &\varepsilon \sim N(0, \Psi) \\ &x \mid z \sim N(\mu + \Lambda z, \Psi) \end{aligned}, \tag{1}$$

therefore, a joint distribution on (x, z) is posited as

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim N\left(\begin{bmatrix} \mu \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Lambda \\ \Lambda_T & I \end{bmatrix} \right) \tag{2}$$

where $z \in R^k$ is a latent random variable: The corresponding graphical model is demonstrated in Figure 1.

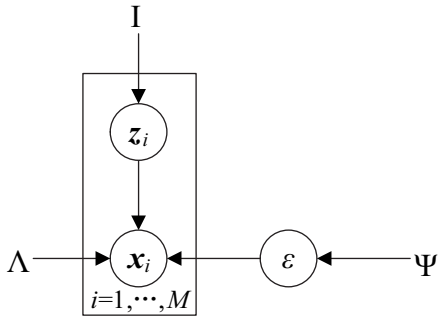

Figure 1. Graphical model for factor analysis

In this model, x, μ, and ε are all N-dimensional vectors, $\Psi = diag(\varphi_1, \varphi_2, ...\varphi_N)$, Λ is a factor loading matrix with size of N *K, and z is a K-dimensional vector. Note that the loading matrix plays the same role as in the basis matrix in Principal Components Analysis. This linear generative model with Gaussian

latent variables differs from PCA in that the variance of the noise variable ε is a diagonal matrix rather than a unit matrix, and thus it contains a richer interpretation in the noise space ψ. For there is no offset in our model, we assume that x has zero mean μ.

Substitute multivariate Gaussian into the conditional probabilistic in equation group 1, we have

$$p(x \mid z) = \exp(-\frac{1}{2}(x-\Lambda z)^T \Psi^{-1}(x-\Lambda z)\} \quad (3)$$

In our model, the only observed variables are x_i, and we have to inverse this generative process and evaluate three parameters Λ, μ, ψ, and the latent variables z_i. Since marginal distributions of Gaussians are themselves Gaussian, we therefore have that the marginal distribution of x is given by[13]

$$x \sim N(\mu, \Lambda\Lambda + \psi) \quad (4)$$

Given the training data set{ $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$,…, $\boldsymbol{x}^{(M)}$}, we can easily have our log likelihood functions, which, however, is intractable to maximize for no close form solution is existed. In this paper, we use EM algorithm instead since EM is an elegant and powerful method for finding maximum likelihood solutions for models with latent variables[13].

4. Topic detection using factor analysis

We can see in the algorithm from the graphical model that we assume the observed words in each document are independent given the latent variable z, which is in common with LDA. In EM algorithm for factor analysis, we have E step[13]:

$$E[z_i] = (I + \Lambda^T \Psi^{-1}\Lambda)\Lambda^T \Psi^{-1}(x_i - \overline{x}) \quad (5)$$

$$E[z_i z_i^T] = I + \Lambda^T \Psi^{-1}\Lambda + E[z_i]E[z_i]^T \quad (6)$$

Note that this can be evaluated through inversion of matrices of size $K\times K$. This is quite efficient since K is often much smaller than N.

Similarly, in the M step, we update these three parameters as follows[13]:

$$\mu' = 1/M \cdot \sum_{i=1}^{M} x_i \quad (7)$$

$$\Lambda' = [\sum_{i=1}^{M}(x_i - \mu')E[z_i]^T][\sum_{i=1}^{M} E[z_i z_i^T]]^{-1} \quad (8)$$

$$\Psi' = diag\{S - \Lambda' \frac{1}{M}\sum_{i=1}^{M} E[z_i](x_i - \mu')^T\}, \quad (9)$$

where, μ' won't change as parameters vary, and the diag(.) is a function that ignore the nondiagonal elements as zeros. S is the sample covariance matrix defined by

$$S = \frac{1}{M}\sum_{i=1}^{M}(x_i - \mu')(x_i - \mu')^T . \quad (10)$$

It is interesting to note the close relationship between equation(5) and the normal equation that one derived for least squares regression, $\theta^T = (y^T X)(X^T X)^{-1}$.

Factor analysis based topic detection(FA-TD) is in fact a type of matrix factorization, where we have a document sets **X** with matrix size of $M\times N$, each row of which represent a sample with N-dimensional features, factorized into a low dimensional matrix with size of $M\times K$. our model regarded K factors as topics that composed a particular paper. Since the mean vector μ is set to zero, the factorization process is shown in

$$\Lambda_{N\times K} \bullet \begin{bmatrix} | & | & \dots & | \\ y_1 & y_2 & \dots & y_M \\ | & | & \dots & | \end{bmatrix} = \begin{bmatrix} | & | & \dots & | \\ x_1 & x_2 & \dots & x_M \\ | & | & \dots & | \end{bmatrix}, \quad (10)$$

where the element vector y_i in matrix Y is K-dimensional, while x_i is N-dimensional vector. Thus we have $X_{N\times M}=\Lambda_{N\times K}Y_{K\times M}$. our factor analysis based topic detection can be summarized in algorithm 1.

Algorithm 1. FA-TD algorithm.

Input: M N-dimensional word count matrix $\boldsymbol{W}$
Output: topic no. and top L words in each topic.

1. Initialize the diagonal noise ψ
2. Get the original word frequency $\boldsymbol{W}$
3. Transform $\boldsymbol{W}$ into *tf-idf* matrix X which is composed of data points $X=\{x_1, x_2, \dots, x_M\}$
4. initialize Λ, μ, ψ using random generators
5. $K = 3$ // according to real corpus.
6. **while** terminiation criterion not reached **do**
7. //E step
8. Evaluate $E[z_i]$, $E[z_i z_i^T]$ using equation 5,6
9. //M step
10. Update Λ, μ, ψ using equation 7, 8, 9
11. Transform Λ to get Y
12. **for** m= 1,2,....,M
13. **for** i = 1, 2, ...K
14. select top L weight words in Λ_i
15. **print** topic for each text entity
16. return Y, Λ,

The terminiation criterion is that either the parameters or the log likelihood does not change.

5. Experiment

5.1. Corpus

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups, which of these groups can be invided into 4 major classes as in Table 1. Approximately 4% of the articles are crossposted. The articles are typical post-

ings and thus have headers including subject lines, signature files, and quoted portions of other articles.. it was originally collected by Ken Lang[15], though he does not explicitly mention this collection. The 20 Newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and clustering.

Table 1. Detailed topic information in corpus newsgroup20.

Topic1	Topic2	Topic 3	Topic 4
alt.atheism	comp.sys.ibm.pc.hardware	rec.autos	sci.crypt
talk.politics.guns	comp.graphics	rec.motorcycles	sci.electronics
talk.politics.mideast	comp.os.ms-windows.misc	rec.sport.baseball	sci.space
talk.politics.misc	comp.sys.mac.hardware	rec.sport.hockey	sci.med
talk.religion.misc	comp.windows.x		
soc.religion.christian			misc.forsale

NIPS12 research literature is our second test corpus, which include abstractions and original paper content from 2000 to 2012, 13 years of papers in International Conference on Neural Information Processing Systems(NIPS). For convenience, we download from url *http://www.cs.nyu.edu/~roweis/* for the word-frequency matrix data *nips12raw_str602.mat* which was prepared for further research purpose, and there are 1740 papers in 13 years in total, and the size of word list is 13649.

We used these two corpus as experimental data for two reasons: (1) These are two different types, i.e. newsgroup corpus and research literature,; (2) the sample size is larger than the dimensional size in newsgroup20 while smaller in NIPS12. both corpus can be used to vervify the effeciency of our method.

5.2. Experimental Result

To perform topic detection on text documents, we have to turn the corpus into numerical feature vectors. Since text data is a high dimentional sparse matrix, we saved much memory by storing data into sparse matrix. We used scikit-learn toolkit based on python to perform text preprocessing, tokenizing and filtering of stopwords which can build features dictionary and transform documents to feature vectors for us[16].

All of the codes were written in python, and we select python for three reasons: first, it is simple and easy to program; second, the rich machine learning libary is strongly supported, third, python support sparse matrix computation and it iseasy to draw statistic plot with matplotlib module. We set L=20, K=4 for newsgroup and L=10, K=5 for NIPS12 and the experimental result is shown in table 2 and 3 for newsgroup20 and NIPS12 respectively.

Table 2. topic results for newsgroup20.

T no.	Top 20 words
topic 1	god jesus bible does christian faith people christians christ believe life true church heaven religion sin lord say belief human
topic 2	thanks windows edu mail file know does drive card help software use using hi advance program problem pc email need
topic 3	just don think like people know good time year way really ll car did make ve years got say new
topic 4	key chip government clipper use encryption keys law public enforcement secure data nsa people phone used going citizens clinton security

Table 3. topic results for NIPS12 in 2011.

T no.	Top 10 words using FA-TD	Top 10 words using LDA
topic 1	network units input neural hidden output training learning weights layer	model results set noise single function hidden output approach vector
topic 2	cells model cell neurons neuron firing visual spike synaptic response	function output networks neural figure time visual weights models layer
topic 3	learning state policy reinforcement action control states mdp sutton reward	neural weight paper single hidden class time layer results systems
topic 4	data model algorithm function error distribution training gaussian probability linear	neural system recognition parameters problem matrix noise visual fig order
topic 5	recognition image speech word images system training object hmm features	large units neurons values algorithm parameters space figure models time

6. Analysis

From table 2, we can see that topic 1 is about politics and religion, and topic 2 is about computer, and that topic 3 is about car, and topic 4 is about science. It is obvious that our algorithm fail to detect topic on sports, this may be due to that our topic number is selected as 4, and there may be some information missing. However, for data with size of 2000*1000, our algorithm is done in 3.04 seconds, while for NIPS12 the input data with size of 1740*13649 we complete the topic detection in 6.68 seconds using 5 topics and top 10 words. Our FA-TD outperformed LDA model in terms of time complexity since LDA needed 40.68 and 148.29 seconds.

From Table 3, we can see that 5 topics detected in NIPS12 which are neural network, neurons synaptic modeling from biology, reinforcement learning, linear gaussion model and applications containing image, speech, face and other recognition tasks. This is almost the same as NIPS call-for-paper. However, the result from LDA seems a little poorer from subjective view.

7. Conclusion

Compared with the traditional LDA, our algorithm's time complexity is lower, EM iteration algorithm much faster, and topic effect better in both research literature and news corpus in terms of both objective and subjective view; We can easily extended to multilayer topic detection; Compared with other matrix decomposition method, we can get the semantic interpretation for each dimension with meaningful words that related to the topic; Compared to other matrix decomposition method, the difference is that our hypothesis is that the factor of each document as a linear transformation of a Gaussian, while the topic on the assumption of LDA is multinomial distribution; Like LDA, our method can also solve big *p* small *n* problem.

Acknowledgements

This work is supported by (1) the National Natural Science Foundation of China (61403238, 61502288, 61071192, 61271357, 61171178), (2) Natural Science Foundation of Shanxi Province (2014021022-1), (3) Outstanding Graduate Innovation Project of Shanxi Province (20123098), and (4) International S&T Cooperation Program of Shanxi Province(2013081035).

References

1. Kleinberg, Jon. «Temporal dynamics of online information streams.» Data stream management: Processing high-speed data streams (2006).
2. Takahashi, Tatsuro, Ryota Tomioka, and Kenji Yamanishi. «Discovering emerging topics in social streams via link-anomaly detection.» Knowledge and Data Engineering, IEEE Transactions on 26.1 (2014): 120-130.
3. Chen Y, Amiri H, Li Z, et al. Emerging topic detection for organizations from microblogs[C]// Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013: 43-52.
4. Benhardus, James, and Jugal Kalita. «Streaming trend detection in twitter.» International Journal of Web Based Communities 9.1 (2013): 122-139.
5. Tang, Guanting, Jian Pei, and Wo-Shun Luk. «Email mining: tasks, common techniques, and tools.» Knowledge and Information Systems 41.1 (2014): 1-31.
6. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. «Latent dirichlet allocation.» the Journal of machine Learning research 3 (2003): 993-1022.
7. AlSumait, L.; Barbar´a, D.; and Domeniconi, C. 2008. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In ICDM, 3–12. IEEE.
8. Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study final report.
9. M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In Proceedings of Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00), pages 109–110, August 20–23 2000.
10. S. Meyer zu Eissen and B. Stein. Analysis of clustering algorithms for web-based search. In D. Karagiannis and U. Reimer, editors, PAKM, volume 2569 of Lecture Notes in Computer Science, pages 168–178. Springer, 2002.
11. H. Li and K. Yamanishi. Topic analysis using a finite mixture model. Inf. Process. Manage, 39(4):521–541, 2003.
12. Wartena C, Brussee R. Topic detection by clustering keywords[C]//Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on. IEEE, 2008: 54-58.
13. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

14. Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.
15. Lang, Ken. «Newsweeder: Learning to filter netnews.» Proceedings of the 12th international conference on machine learning. 1995.
16. scikit-learn homepage: http://scikit-learn.org/stable/
17. Xu, Ke, et al. «Unsupervised satellite image classification using markov field topic model.» Geoscience and Remote Sensing Letters, IEEE 10.1 (2013): 130-134.

Design of cognitive algorithm based on data traffic control for underground accurate positioning system

Lijun Tang[1]*, Wei Wu[2]

1. Chongqing Vocational Institute of Engineering, Chongqing 402260, China
2. Chongqing city management college, Chongqing, 401331, China

Abstract
In the existing accurate positioning system, a large amount of positioning data are uploaded to the ground control center, which often cause link congestion for the positioning system. The link congestion further degrades the positioning performance like real-time and reliability. In order to overcome the above disadvantages about the existing accurate positioning system, a novel cognitive algorithm based on data traffic control is proposed to reduce data traffic and avoid link congestion for the positioning system. The algorithm uploads location data only when target nodes are in motion, and the control center uses V-T algorithm to predict the complete trajectory of the target nodes simultaneously. Finally the simulation results show that the proposed algorithm applying to the accurate positioning system for mine coal can greatly reduce data traffic and don't degrade the trajectory tracking performance.
Keywords: ACCURATE POSITIONING; DATA TRAFFIC CONTROL; MINE COAL; COGNITIVE ALGORITHMS.

1. Introduction

With accurate positioning systems widely used in coal mines, they makes the ground control center can monitor the accurate location where miners are in real-time. When an accident occurred, the positioning system can provide important information to rescuers for rescuing the trapped miners.

Due to the harsh environment of the coal mine, data transmission rate is very low, even only a few tens of kB, between the ground control center and base station [1]. However, for tracking the target