Classification of Distributed Data Using Topic Modeling and Maximum Variation Sampling

Robert M. Patton
Oak Ridge National Laboratory
pattonrm@ornl.gov

Justin M. Beaver
Oak Ridge National Laboratory
beaverjm@ornl.gov

Thomas E. Potok
Oak Ridge National Laboratory
potokte@ornl.gov

Abstract

From a management perspective, understanding the information that exists on a network and how it is distributed provides a critical advantage. This work explores the use of topic modeling as an approach to automatically determine the classes of information that exist on an organization's network, and then use the resultant topics as centroid vectors for the classification of individual documents in order to understand the distribution of information topics across the enterprise network. The approach is tested using the 20 Newsgroups dataset.

1. Introduction

Current business and government enterprises rely heavily on electronically stored data for most aspects of their operations. The controls implemented to manage the data are typically focused on flexibility and freedom for individual users of the enterprise's computer systems. Users are permitted to create, modify, delete, and transfer data with a high degree of autonomy. While access controls are implemented at the host and account levels, there are no toolsets available that provide insight into user activities or data interactions within local accounts. Thus, organizations have very little visibility into the nature and distribution of their enterprise data at the user level. In addition, there is a lack of technology that reliably automates the collection, classification, and presentation of data distribution information.

From a management perspective, understanding the information that exists on a network and how it is distributed provides a critical advantage. It provides knowledge of which users have access to which data, how similar information topics are being pursued by which organizational units, and allows an enterprise to defend its highest value computational assets more vigorously. Current approaches center on identifying those computational assets that manage large volumes of data such as file shares and mail servers. However, a computer's role provides no insight into the data it contains. For example, does the computer of a staff engineer contain valuable information? Typically, the answer varies with the person, their projects, and their role on those projects. Unfortunately, the fluid nature of staff members, projects, and roles in an organization makes it challenging to determine the value of information on an employee's computer based on these criteria. Given the flux of information on a given host, role-assignment approaches seem impractical. We believe that an automated means to discover a hosts' contained information is needed.

In the context of information security, users' freedoms with documents present several challenges. Users have the ability to change permissions, redistribute files, and allow access to the downloaded data in a manner that may be inconsistent with the original intent, or any established security policies. Furthermore, users may edit and borrow text from sensitive documents to create new documents that are still inherently sensitive, yet may no longer be subject to access restrictions. While such user actions are more often for convenience than explicit malicious acts, the result is that an organization's inventory of enterprise data across their networked computers is unreliable a best.

This work explores the use of topic modeling as an approach to automatically determine the classes of information that exist on an organization's network, and then use the resultant topics as centroid vectors for the classification of individual documents in order to understand the distribution of information topics across the enterprise network. This approach addresses the lack of visibility into user-level data by automatically discovering what topics are most relevant in an organization and the how those topics are distributed across the enterprise network.

2. Related Works

In the work of [13], a topic-oriented approach for distributed retrieval is presented. Their approach relies on clustering documents in a distributed environment, and then generating topic models for each cluster using unigrams. The generated topics are then used for retrieval purposes. In the work of [6], topic modeling with network regularization is proposed. Their effort seeks to leverage the strengths of topic modeling and social network analysis with the goal of not only identifying topics, but also mapping them to a network and discovering topical communities. The work of [4] presents distributed topic map architecture. This architecture enables enterprise knowledge management of information distributed across a network, and enables clients to transparently query topics in a network. Finally, in the works of [2][8], distributed algorithms for the topic models Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP) are developed. Both of these approaches involve the extension of LDA and HDP models to large parallel machines. A general description of topic modeling is provided in [12].

3. Approach

What the related works have not addressed is the volume of data required for topic modeling, which is a critical bottleneck for automated topic discovery and document classification across a network. In a large enterprise, the volume of data may range from hundreds of terabytes to a few petabytes. At this scale, a large supercomputer would be needed in order to perform even parallel versions of topic modeling algorithms, which is a resource not likely to be available to an enterprise class user. Consequently, what is needed is a means of effectively reducing the volume of data for training purposes while minimizing the loss of accuracy in classification.

To meet this need, the work described here proposes the use of maximum variation sampling to reduce the volume of data required for training. Maximum variation sampling (MVS) is a nonprobability-based sampling [9]. This form of sampling is based on purposeful selection, rather than random selection, and seeks to identify a particular sample of data that will represent the diverse data points in a data set. According to [9], "This strategy for purposeful sampling aims at capturing and describing the central themes or principle outcomes that cut across a great deal of [data] variation." The MVS is naturally implemented as a genetic algorithm (MVS-GA). This algorithm including the fitness function was first described in [10], and has since been modified and successfully applied to medical reports [11].

In a distributed, networked environment, the MVS-GA would be applied locally to each node's data for sampling. Samples from the nodes would then be sent to a master node, which would then execute an LDA-based topic modeler on the sample data collected from all the nodes across the network. The generated topics would then be distributed to each node on the network for classification of the data on that node. Results of the classification would then be returned to the master node, and could then be used for fine-tuning cyber defenses.

To better understand the feasibility of this approach, several tests were conducted to measure both the reduction in training data as well as the impact of that reduction on the accuracy of classification.

4. Tests

Six tests were conducted using the 20 Newsgroups data consisting of 18,846 documents [1]. For each test, twenty topics were generated as defined by the user, while the sample size was varied between the tests. Topics were generated using MALLET [1]. The MVS technique is compared to random sampling and using all the documents in the corpus.

The first test involved simply using the entire document set as a training set. Twenty topics were generated, and were then used as centroid vectors to create twenty clusters of the entire corpus. The overall entropy of the twenty clusters was then computed.

The second, third, and fourth tests used random sampling of the entire corpus. The second test used a sample size of 1,884 documents, which is approximately ten percent of the corpus size. If statistical inference were to be performed, this would be an appropriate sample size. The third test used a sample size of 150, which is slightly less than one percent of the corpus size. Finally, the fourth test used a sample size of 40. This was chosen for comparison purposes to the MVS-GA and to observe entropy values at extremely low sample sizes. As with the first test, twenty topics were generated from these samples, and used as centroid vectors to create twenty clusters of the entire corpus. The overall entropy of the twenty clusters was then computed.

The fifth and sixth tests used MVS-GA to sample the entire corpus. For these tests, the entire corpus was randomly split into two groups of approximately equal size. This was done in order to simulate data that was distributed across two machines. For both tests, the

MVS-GA was run on each group of data, and the samples from each group were then combined to provide a single overall sample. The fourth test used a sample size of 150 consisting of two samples of 75 from each group of data, while the fifth test used a sample size of 40 consisting of two samples of 20 from each group of data. The sample size of 40 was used because the MVS technique tends to perform better and is less computationally intensive with small sample sizes. In addition, for an enterprise environment of distributed data, smaller samples can be effectively used to sample individual nodes on the network. As with the previous tests, twenty topics were generated from these samples, and used as centroid vectors to create twenty clusters of the entire corpus. The overall entropy of the twenty clusters was then computed.

The parameters for the MVS-GA were the following: population size of 500, 25 generations, 0.7 crossover rate, 0.03 mutation rate, and tournament selection. In addition, the fitness evaluation of the MVS-GA is a multi-threaded master / slave parallelization implementation to take advantage of multi-core processors and reduce wall clock time.

5. Results

Tests were performed using dual 3 GHz Quad-Core Intel Xeon processors. The results of the five tests are shown in the tables below. Table 2 shows the average runtime of the sampling and training algorithms. The total runtime does not include the time required for classifying the entire corpus based on the twenty centroid vectors. Table 3 shows the percent change of the random and MVS techniques over the baseline of using the entire corpus for training.

Table 1. Test results

Test	Sample Size	Entropy	Avg # words per topic	# Categories represented in sample
All (baseline)	18,846	0.93	19	20
Random 1	1,884	1.04	19	20
Random 2	150	1.16	16.85	20
Random 3	40	1.21	8.05	14
MVS 1	150	1.11	19	20
MVS 2	40	1.18	18.65	15

Table 2. Runtime results

Test	Sampling Runtime (sec)	Training Runtime (sec)	Total Runtime (sec)
All (baseline)	0	358	358
Random 1	1.48	75	79.39
Random 2	0.8	44	44.8
Random 3	0.72	32	32.72
MVS 1	137.5	45	182.5
MVS 2	12.76	31	43.76

Table 3. Percent Change over Baseline

Test	Sample Size	Entropy	Avg # words per topic	# Categories represented in sample
Random 1	-90.0	11.8	0	0
Random 2	-99.2	24.7	-11.3	0
Random 3	-99.8	30.1	-57.6	-30
MVS 1	-99.2	19.4	0	0
MVS 2	-99.8	26.9	-1.8	-25

The entropy was computed according to the equations shown in Equation 1 and Equation 2.

$$E_i = -\sum_j \left(\frac{N_{ij}}{N_i}\right) \log\left(\frac{N_{ij}}{N_i}\right)$$

Equation 1. Entropy per cluster *i*

$$E = \sum_i \left(\frac{N_i}{N}\right) E_i$$

Equation 2. Overall entropy

The entropy values shown in Table 1 were computed using Equation 2. For Equation 1, N_{ij} is the number of documents from class j in cluster i. N_i is the number of documents in cluster i. For Equation 2, N is the number of documents in the entire corpus. A lower entropy value indicates better cohesion of the documents to the topics, and vice versa.

6. Discussion

As expected, there is a clear trade-off between entropy and reduction in the amount of data used for

training and classification. One encouraging aspect of the results is that the reduction in data is considerably more than the corresponding increase in entropy. However, there are many more variables and characteristics of the algorithms and data to consider before any conclusions can be drawn.

First, sampling techniques, in general, often suffer from the lack of representative documents from the various classes (i.e., subpopulations or strata within the document corpus). For this particular data set, a stratified sampling technique would have performed better in this regard as the broad subpopulations were known *a priori*. However, in an enterprise environment with distributed data where the classification is not known *a priori*, stratified sampling would be difficult to implement effectively. Stratified sampling could be performed across the network to ensure adequate sampling of *locations* of the data, however, this would still not address adequate sampling of the *content* of the data. Given that the classifications would not be known *a priori* in an enterprise environment, the MVS technique performs slightly better than the random sampling, especially at very low sample sizes. Future work will explore hybrid sampling consisting of stratified and maximum variation sampling techniques to adequately and effectively sample both the location and content of data.

Second, the sample size is intimately connected with the sampling technique in terms of precision and run-time execution. As can be seen from the *"Random 2"* and *"MVS 1"* tests and from the "*Random 3*" and "*MVS 2*" tests, the same sample size provided very different outcomes in entropy, representativeness, and runtime. This is a clear reflection of the characteristics of the sampling techniques. MVS pursues diversity in the sample and is computationally expensive as the sample size increases. Random sampling tends to neglect diversity for the sake of representativeness of the *proportion* of the data, and is not computational intensive. As a result of the MVS pursuing diversity in the sample, it inadvertently achieves better representativeness of the *classes* of the data (especially at very low sample sizes), which is what the topic modeling algorithms need to achieve better topic models. Unfortunately, better diversity in the sample does not guarantee better representativeness of the classes of data. For example, a single class of data may be highly diverse while the other classes are homogenous, and class separation is low. In this example, the MVS would most likely sample exclusively from the highly diverse class of data. However, in an enterprise environment, this case is not likely to occur.

Finally, the nature of the genetic algorithm plays a significant role in the performance of the MVS technique. As discussed in [3], there are numerous parameters for a canonical genetic algorithm. Different parameter settings can dramatically alter the outcome. Additional testing with different GA parameters is needed as well as additional work to explore more advanced evolutionary techniques such as memetic algorithms [7].

6. Acknowledgements

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid- up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

7. References

[1] 20 Newsgroups data. [Online]. Available: http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] A. Asuncion, P. Smyth, and M. Welling, "Asynchronous distributed learning of topic models," in Advances in Neural Information Processing Systems 21, 2008, pp. 81-88.

[3] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. 1989.

[4] A. Korthaus, S. Henke, M. Aleksy, and M. Schader, "A Distributed Topic Map Architecture for Enterprise Knowledge Management," Proc of the 5th IEEE/ACIS International Conference on Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR 2006.

[5] MALLET [Online]. Available: http://mallet.cs.umass.edu

[6] Q. Mei, D. Cai, D. Zhang, and C. Zhai, "Topic modeling with network regularization." In Proceeding of the 17th international Conference on World Wide Web (Beijing, China, April 21 - 25, 2008). WWW '08. ACM, New York, NY, 101-110.

[7] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms". Caltech Concurrent Computation Program (report 826), 1989.

[8] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed Algorithms for Topic Models," in Journal of Machine Learning Research, 10, 2009, pp. 1801-1828.

[9] M.Q. Patton, Qualitative Evaluation and Research Methods, Second Edition. Newbury Park, CA: Sage Publications, Inc., 1990.

[10] R. M. Patton, and T. E. Potok, "Adaptive Sampling of Text Documents," Proc. of the 13th International Conference on Intelligent and Adaptive Systems and Software Engineering, July 2004.

[11] R.M. Patton, T.E. Potok, B.G. Beckerman, and J.N. Treadwell, "A Genetic Algorithm for Learning Significant Phrase Patterns in Radiology Reports." Proceedings of the 5th GECCO Workshop on Medical Applications of Genetic and Evolutionary Computation (MedGEC), Montreal, Canada, July 2009. ACM Press, New York, NY, 2061-2064.

[12] M. Steyvers and T. Griffiths, *Probabilistic Topic Models.* Lawrence Erlbaum Associates, 2007. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1410615340

[13] J. Xu, and W.B. Croft, "Topic-Based Language Models for Distributed Retrieval" in Advances in Information Retrieval, vol. 7, Springer US, 2000, pp. 151-172.