Unsupervised academic curricula evaluation through Latent Dirichlet allocation

Jean Michel Rouly
jrouly@gmu.edu

Huzefa Rangwala
rangwala@cs.gmu.edu

January 25, 2015

1. INTRODUCTION

Programs of study at institutions of higher education are a chain composed of the courses required to complete a degree. These component courses in turn are composed of the topics or concepts they are intended to cover. Evaluation of the courses within a particular program is a key process in the evaluation of an overall academic curriculum. Analyzing the structure of a program's prerequisite chain, for example, requires an understanding of each constituent course and the overlap, if present, of covered topics between courses and their prerequisites. Additionally, inter-institutional curricular comparison requires an aggregate evaluation of the courses within each institution's relevant program. However, comparing and evaluating different courses requires expert knowledge in the relevant field. Placing, for example, two courses, one in data mining and one in database theory, under evaluation may not yield any inherent similarity unless a domain expert can identify the concepts expected from each class.

Automating the information retrieval process to identify core concepts covered in any particular course removes the need for a domain expert. By analyzing course syllabi from a corpus spanning fields and institutions, topic modeling can provide a method to generate a semantic representation of core course concepts. Specifically, unsupervised latent variable models present a method of identifying the core concepts (ie. topics) covered in a course. This introduces the possibility of applications in automated course and program evaluation methods.

2. Background on Latent Dirichlet allocation (LDA)

Topic modeling, a form of latent variable modeling, is an unsupervised machine learning method which attempts to recreate the distribution of so-called “topics” an author used to generate a corpus of documents. In this case, a topic is a frequency distribution of terms within a vocabulary. The topics discovered in a corpus can be used to categorize documents and provide structure to an otherwise unknown dataset.

LDA is a specific type of topic modeling which assumes that multiple topics exist within a single document (ie. were used to generate that document). [2] LDA assumes a generative process where, for each word in the document, it selects a distribution over topics, selects a topic, and then selects a vocabulary term. [2] By picking a distribution over topics, multiple possible topics can be blended into a single document. Reversing this generative process is significantly more difficult because the topic distributions are unknown. This is what the “hidden model” or “latent model” refers to.

LDA can best be understood through its generative process. Given the set of distributions as input, generating the corpus topics is a probabilistic process. Taking the variables $\theta_{d,k}$ (topic proportion for topic k in document d), $\beta_{1:k}$ (topic k), $z_{d,n}$ (topic assignment for word n in document d), and $w_{d,n}$ (the n^{th} word in document d), LDA calculates the posterior probability in Equation 2.1. [1]

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}) = \frac{\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}}{w_{1:D}} \tag{2.1}$$

Using a variety of probabilistic methods to calculate or approximate the denominator (ie. evidence) LDA results in a usable set of vocabulary frequency distributions or topics. Specifically, Gibbs Sampling, a variety of Bayesian Inference, is used to approximate the LDA posterior probability.[2]

Additional unsupervised machine learning tools can be applied to the same problem. Simple clustering algorithms (eg. K-Means) when given the same bag of words corpus representation as input act to identify groupings of similar documents according to their term frequency vector Euclidean distance. [3] Additional, similar clustering algorithms can be applied in a similar manner.

3. Research Question

At a high level, courses can be compared by the core concepts they address. By extrapolating, programs of study are assumed to be the union over the set of all concepts covered in the composite courses. Comparing these courses or programs requires expert knowledge. There is no simple inherent or natural relationship between two separate courses in distinct domains; however, an expert in the domains would perhaps argue for a link via common conceptual topics.

Through the application of probabilistic machine learning methods, specifically LDA topic modeling, a corpus of unstructured course syllabi can be digested and mined for topics. In this scenario, each topic represents a core concept covered by the courses. Knowledge of the topics covered in courses as set out by course syllabi informs the design of prerequisite chains and, more generally, the academic program built on these courses.

The primary goal of this work is the development of a system to digest large quantities of data, specifically academic course syllabi, and to process and ultimately generate interactive descriptions of the topics covered within institutional programs as illustrated by core course concepts.

A secondary goal of this work is the identification of an unsupervised means of generating labels for discovered topics. A primary weakness in LDA and unsupervised learning methods in general is an inability to name or otherwise identify the probability distributions generated as topics.

4. Methods

4.1. Data Acquisition

The primary data manipulated in this study are university course syllabi. Currently the experimental data include documents obtained from the George Mason University departments of Computer Science and Statistics as well as the Portland State University department of Computer Science. Simple web scrapers were written using Python and BeautifulSoup to download publicly available syllabi from departmental web pages. Syllabi occurred in a number of different formats, most commonly HTML, Portable Document Format, Microsoft Word Documents, and Rich Text Format. A parser was written for each of these formats to acquire the contained, unstructured text, which was then passed through a cleaning procedure to remove abbreviations and non-English characters.

The Python scraping framework employed is structured to allow pluggable web scrapers tooled to specific syllabus repositories. A major future goal is to expand the breadth of data collection across different institutions. Development is in progress for new scraping engines tooled to new repositories.

4.2. Preliminary Data Exploration

Preliminary exploratory results are promising. We applied K-Means clustering to a sample dataset of syllabi scraped from the GMU Computer Science online archive of syllabi. Using course sections across semesters as ground-truth labels, we obtained results summarized in Table 4.1 and Table 4.2. A distributed implementation of K-Means available in the Python toolkit `scikit-learn` was used to perform the clustering.

The high completeness values are promising: this indicates that many of the same course are assigned under the same cluster prototype. The low value of homogeneity is unsurprising given the initialization parameters used: K-Means was forced to detect only 20 clusters, a far smaller number than the magnitude of distinct course sections available. The number 20 was chosen arbitrarily as a smaller count than the true number of distinct course sections in order to increase cluster size. Cursory visual inspection of the most common frequencies in the first few clusters (Table 4.2) also supports the structured nature of the dataset with semantically related terms grouped together in a logical fashion corresponding with an obvious, known GMU course.

Execution time	0.144568s
Homogeneity	0.415
Completeness	0.877
V-measure	0.563

Table 4.1: Preliminary clustering metrics

Cluster 1	Cluster 2	Cluster 3
intelligence	chapter	software
artificial	project	swe
agents	sipser	testing
learning	networks	web
tecuci	layer	interfaces
expert	savitch	construction
knowledge	data	design
reasoning	dlc	constructing
semantic	experimental	professor
intelligent	design	quality

Table 4.2: Structure of clustering results

4.3. Topic Modeling

After the exploratory clustering process, we passed cleaned data into a topic modeling framework. The Java MALLET library is used to perform Latent Dirichlet allocation (LDA) on the combined syllabus data set. A data pipeline is constructed using the MALLET API that reads input data, tokenizes it, strips stop words, and trains an LDA topic model. Initial topic modeling results are later piped into a simple, interactive user interface.

Sample preliminary results are illustrated in Table 4.3, showing the breakdown of documents into component topics, and Table 4.4, showing topic definition. Further study is needed to evaluate the success of this method and its application to the data set. Additionally, further work is needed to construct labels for the data set. That is, each course syllabus is currently an anonymous, independent data point. No metadata is considered, specifically semester or section name.

Doc	Topic	Proportion	Topic	Proportion
0	33	0.7666641741676518	62	0.230550274742815
1	44	0.5776374037067855	8	0.3152509716025131
2	86	0.8143297134325639	62	0.18015768047706127
3	9	0.9491106700671812	5	0.034876914241398584
4	82	0.5736690412365056	53	0.39539480434234386

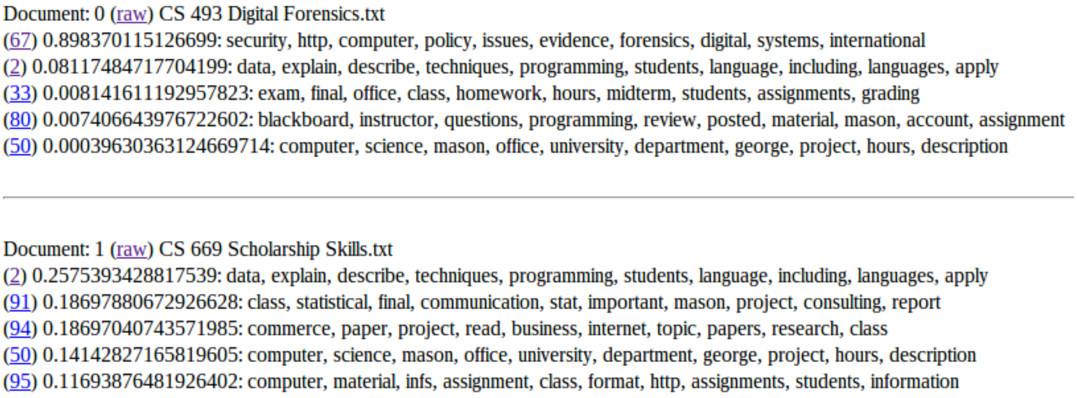
Table 4.3: Structure of per-document topic breakdown

Topic	Term	Term	Term	Term
0	lisp (98)	june (57)	prolog (46)	http (45)
1	systems (154)	operating (119)	system (101)	programming (92)
2	systems (304)	operating (252)	students (189)	projects (146)
3	randomization (66)	trials (57)	clinical (57)	outcome (30)
4	database (389)	relational (151)	design (133)	model (124)

Table 4.4: Structure of topic term-frequency definition

4.4. Visualization

A basic, interactive user visualization of syllabus topics has been prototyped. Output data is piped from the topic model and stored a simple relational database to store topic definitions and document composition information. This data is then combinatorially expanded into two cross-referenced HTML documents using a simple Python script. Images of the visualization are included for reference, see Figure 4.1 and Figure 4.2. It will be a subject of future work to create a richer browsing experience to investigate learned document topics.



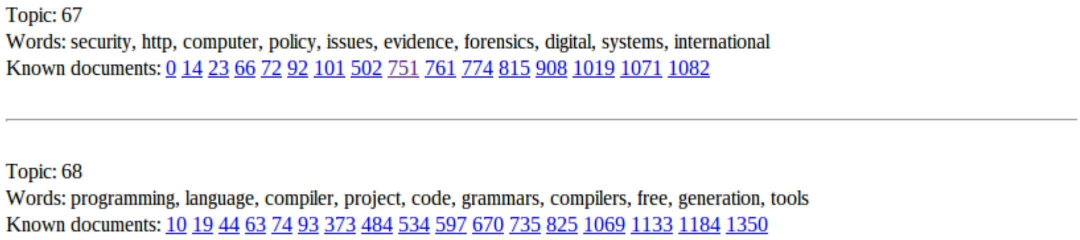
Document: 0 (raw) CS 493 Digital Forensics.txt
(67) 0.898370115126699: security, http, computer, policy, issues, evidence, forensics, digital, systems, international
(2) 0.08117484717704199: data, explain, describe, techniques, programming, students, language, including, languages, apply
(33) 0.008141611192957823: exam, final, office, class, homework, hours, midterm, students, assignments, grading
(80) 0.007406643976722602: blackboard, instructor, questions, programming, review, posted, material, mason, account, assignment
(50) 0.00039630363124669714: computer, science, mason, office, university, department, george, project, hours, description

Document: 1 (raw) CS 669 Scholarship Skills.txt
(2) 0.2575393428817539: data, explain, describe, techniques, programming, students, language, including, languages, apply
(91) 0.18697880672926628: class, statistical, final, communication, stat, important, mason, project, consulting, report
(94) 0.18697040743571985: commerce, paper, project, read, business, internet, topic, papers, research, class
(50) 0.14142827165819605: computer, science, mason, office, university, department, george, project, hours, description
(95) 0.11693876481926402: computer, material, infs, assignment, class, format, http, assignments, students, information

Figure 4.1: Per-document topic breakdown

4.5. Evaluation

Results evaluation will require two components: the generation of predicted topics and a set of known-good, reliable benchmark topics. The benchmark topics can be classified “known-good” if they encode domain expertise, a characteristic which is fundamentally lacking from the unsupervised topic modeling process. Once a set of proposed topics is obtained from the syllabus data set, the known-good categorization of concepts with labels will be used to propose a second set of topics. Overlap between these two groups of topics will be used to share labels from the known-good set to the experimentally proposed group with some value of confidence. Those common topics will be strongly endorsed or validated, while outliers



Topic: 67
Words: security, http, computer, policy, issues, evidence, forensics, digital, systems, international
Known documents: 0 14 23 66 72 92 101 502 751 761 774 815 908 1019 1071 1082

Topic: 68
Words: programming, language, compiler, project, code, grammars, compilers, free, generation, tools
Known documents: 10 19 44 63 74 93 373 484 534 597 670 735 825 1069 1133 1184 1350

Figure 4.2: Topic-word definitions

may be trimmed if under a certain confidence threshold.

External parties occasionally maintain detailed descriptions of expectations for specific hypothetical courses. For example the ACM maintains an annual writeup of guidelines for undergraduate education. [4] These course descriptions could be employed as the secondary corpus to include implicit expert knowledge.

4.6. Resources

The primary software libraries employed to support this project are the Python machine learning toolkit `scikit-learn`[1], the Java toolkit `Mallet`[2], and the Python web scraping library BeautifulSoup[3]. Other libraries may be incorporated as needed.

Syllabus data has been collected from the George Mason University departments of Computer Science[4] and Statistics[5] along with the Portland State University department of computer Science[6]. Additional institutions targeted for future scraping include Colorado, Rice, UNCG, and Chaminade. All of the listed schools host publicly available syllabus archives, at least for some departments. It may become necessary to contact other universities directly to acquire their syllabus data. Additionally, the Open Syllabus Project[7] may prove a useful resource or collaborator in the future.

5. Discussion

The primary results of this initial study have been positive. Exploratory K-Means clustering immediately resulted in clusters with a high level of completeness. Even superficial manual analysis of clusters indicated that the collected data were being appropriately grouped. Similar findings were encountered after the application of LDA topic modeling. Formal evaluation is still missing, although possible techniques have been discussed.

[1] `http://scikit-learn.org`

[2] `http://mallet.cs.umass.edu`

[3] `http://www.crummy.com/software/BeautifulSoup/`

[4] `http://cs.gmu.edu/courses/`

[5] `http://statistics.gmu.edu/courses/`

[6] `http://www.pdx.edu/computer-science/courses`

[7] `http://opensyllabusproject.org`

Many components of this study are still missing and will need to be implemented in the future. First, there is no metadata processing. The immediate result of this is that documents cannot be labeled or grouped based on their section number, meaning the dataset is simply a set of anonymous documents. This is less than satisfactory, given that the purpose of applying topic modeling was the extraction of meaningful data. Another direct consequence of metadata-blindness is the inability to construct prerequisite chains. If documents are not identified by their section number, it will be impossible to list prerequisites, let alone correlate the intersection of topics between prerequisites and sequels. This is a core issue that needs to be addressed in a future version of the work.

A possible solution would include further development of the database schema system introduced for visualization. Perhaps heavier use of a structured relational database would naturally allow for the collection of metadata in a structured manner. One approach to this solution would be the creation of an intermediary database layer between the scraping and learning libraries instead of writing directly to disk. Drawbacks to this solution include the need to refactor large portions of the scraping library as well as the need to develop a software or network interface between the database layer and the MALLET library.

Another major missing component is the rich visualization library. With the creation of the structured database, visualizations were made possible. However, no significant work has been performed to create a usable visualization tool. In the same vein, much of the interface in general is lacking in usability and would greatly benefit from a design overhaul.

The final missing component is the evaluation suite. This will only be possible once the previously mentioned metadata-awareness is implemented.

6. Conclusion

Programs of study in higher education differ widely between departments and universities. In a perfect world, the same program of study at two institutions would provide students with the same core concepts. Indeed, this is all too often not the case. Because of these discrepencies, program accreditation and evaluation methodologies are employed with the goal of standardizing the contents of a program of study. However, this process generally requires manual inspection by a domain expert to extract information from large quantities of course syllabi. Automating the digestion and processing of these syllabi will greatly reduce the time and effort required. Topic Modeling presents a statistical machine learning method to extract the hidden topics behind a document set, in this case the core concepts covered by a course syllabus. Latent Dirichlet allocation (LDA) results in a feasible breakdown of textual syllabi into component concepts. More work will be required to fully utilize the power of this tool, but the results highlighted in this study show great promise.

REFERENCES

[1] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[3] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, Mar 1982.

[4] ACM/IEEE-CS Joint Task Force on Computing Curricula. Computer science curricula 2013. Technical report, ACM Press and IEEE Computer Society Press, December 2013.

A. APPENDIX

Source code of current archive is available online at https://github.com/jrouly/trajectory.