Towards rapidly developing database-supported machine learning applications

Frank Rosner[1] and Alexander Hinneburg[2]

[1] Global Data and Analytics, Allianz SE, Germany
[2] Computer Science, Martin-Luther-University Halle-Wittenberg, Germany

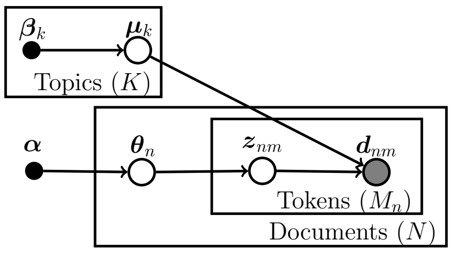
Abstract. The development of a big data analytics application benefits from a conceptual model that jointly represents aspects about data management as well as machine learning. We demonstrate a recently proposed method to translate a Bayesian network into a usable entity relationship model using the real world example of the TopicExplorer system. TopicExplorer is an interactive web application for text mining that uses Bayesian topic models as a core component. Further, we sketch a vision of a conceptual framework that eases machine learning specific development tasks during building big data analytics applications.

1 Introduction

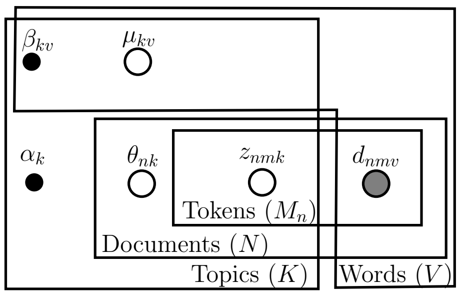
The implementation of a big data analytics application requires to join data management software with machine learning tools. However, the fields of data management and machine learning developed quite different models and notations. The former frequently uses entity-relationship models (ERM) [5] while the latter uses probabilistic graphical models in particular Bayesian networks (BN) to communicate key concepts during development. Even while both kinds of graphical notations show many details of the data, information explicit on one side remains implicit on the other one and vice versa — there is no natural understanding of the two worlds. However, a common conceptual description of the contributions from both worlds is crucial for the successes of big data analytics development projects.

Recently, we proposed a translation [13, 14] from a graphical BN model in plate notation into an entity relationship model. Such ERM can be easily integrated into the overall ERM of the whole application. Thus, we gain the advantage of a formal conceptual view of the machine learning part that is integrated into the conceptual view of the data management side. Thereby, developers from the data management side understand the basic in- and outputs of the machine learning part that remains no longer as a black box behind an abstract API in the data management model.

We demonstrate the method in the real world example of the TopicExplorer in Section 2. Based on this, we describe our vision of a conceptual framework that uses pre-translated BNs as a library of ERM snippets in Section 3. Such library could be used by data management developers to conceptually include machine



(a) LDA plate model



(b) Atomic LDA plate model

Fig. 1: Transformation of the LDA plate model to an APM.

learning methods into analytics applications. We believe that software development of big data analytics applications could benefit from machine learning implementations that are attached to the pre-translated BNs. Last, we discuss related work in Section 4 and conclude the paper in Section 5.

2 Case Study: Text Topic Modeling

We demonstrate the recent method [13, 14] to translate BN to ERM using the example of the TopicExplorer [10, 11], an application to explore document collections using probabilistic topic models [4]. We explain how the translated LDA topic model is represented as ERM. Furthermore, we show use cases for typical analyses supported by the translated ERM.

2.1 Translation of Latent Dirichlet Allocation to ERM

LDA [4] models a collection of documents that is indexed by the set N. Each document $n \in N$ consists of a set of tokens M_n that represents the words occurring in this document. The document specific token index sets M_n partition the total index set of tokens M. A token $m \in M_n$ corresponds to exactly one word type v from a vocabulary V. In the Bayesian network, Figure 1a, this information is coded as a bit vector $\boldsymbol{d}_{nm} \in \{0,1\}^{|V|}$ that has exactly a single 1 at the index associated with the respective word $v \in V$. Each token $m \in M_n$ is also assigned to a topic $k \in K$. This assignment is coded by the bit vector $\boldsymbol{z}_{nm} \in \{0,1\}^{|K|}$, which has a single 1 at the respective topic index. Furthermore, each topic has its own word distribution parameterized by a vector of positive real number $\boldsymbol{\mu}_k \in \mathbb{R}^{|V|}$. The topic proportions per document are represented by a similar vector $\boldsymbol{\theta}_n \in \mathbb{R}^{|K|}$. The hyper-parameter vectors $\boldsymbol{\alpha} \in \mathbb{R}^{|K|}$ and $\boldsymbol{\beta}_k \in \mathbb{R}^{|V|}$ regulate the prior distributions for the respective hidden parameters $\boldsymbol{\theta}_n$ and $\boldsymbol{\mu}_k$.

The translation [13, 14] delivers the ERM shown in the right part of Figure 2 for the given LDA plate model (Figure 1a). The translation employs several intermediate steps, one of which is the transformation of the plate model into

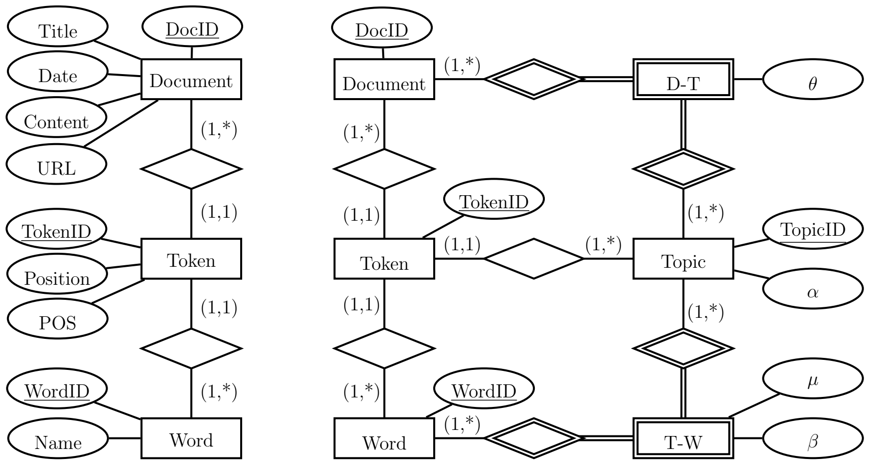


Fig. 2: ERM of given Data (left) and translated ERM for LDA (right).

an atomic plate model (APM), see Figure 1b. The APM represents implicit relational information hidden in the original plate model in an explicit way using the plate notation. By the transformation and reduction rules [13, 14, 8], the APM is translated into a sequence of several intermediary ERMs and then reduced into a usable final ERM.

Such ERM is close to a manually designed ERM for LDA. A document consists of one or more tokens which are of exactly one word type. Each token is assigned to a topic, while one topic can have multiple tokens assigned. The inferred topic mixture for each document is stored in `D-T.`θ, while `T-W.`μ holds the word probabilities for each topic. The hyper-parameter α of the prior for the topic mixture resides as an attribute of the topic entity type. The parameters for the individual priors for the word distributions are stored in `T-W.`β.

2.2 TopicExplorer

TopicExplorer is a web application that helps users from the humanities to work with topic models, e.g. in a collaboration with the institute for Japanese studies at Martin-Luther-University, we analyzed blog posts about the Fukushima disaster. After crawling relevant blogs, each blog entry is preprocessed by computer linguistic software to extract tokens from full text and store them in their lemmatized forms together with their part-of-speech tags (e.g. noun, verb or adjective) and their string positions in the text.

TopicExplorer interactively visualizes the topic structure of the documents. The visualizations require to join the data about documents and words together with results from the topic model. An ERM that would integrate the left and the right part of Figure 2 is obtained by merging the matching entities from both sides. It gives the application developer a good idea how to access those data, without needing to understand the machine learning details of a topic model.

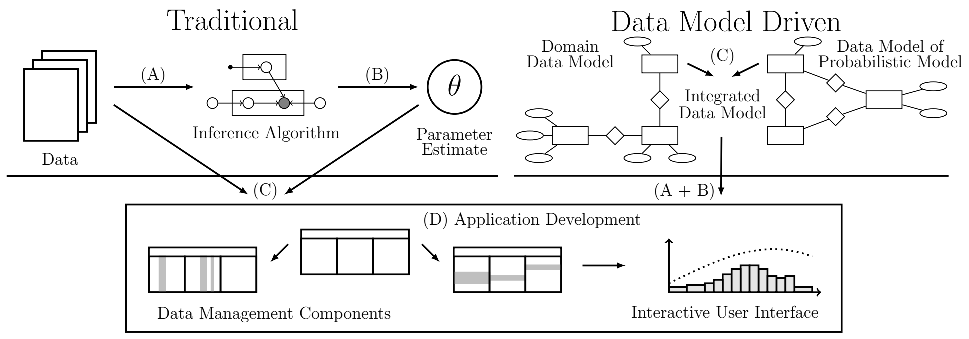


Fig. 3: Comparison of traditional development versus data model driven development of big data analysis applications.

We present how to derive a few visualizations that are part of the current version of TopicExplorer [10].
Document topic mixture. LDA assigns a vector of topic probabilities stored in `D-T.`θ for each document, called a topic mixture. This could be presented as a list of topics with decreasing order of probabilities.
Topic Documents. Reversing the idea behind the document topic mixture, one can visualize a topic as a list of the most representative documents for this topic. This is done by joining `Document`, `Token` and `Topic`, grouping by both IDs of documents and topics and counting the number of tokens in each group. For each topic the entries are sorted with decreasing token count, yielding a list of representative documents.
Topic words. As stated above, a topic can be represented as a list of words sorted in decreasing probability (`T-W.`μ). Furthermore, the topics appear linearly ordered by similarity in the user interface to allow browsing in a semantically uninterrupted way. Computing all pair-wise similarities between topics is well supported by a relational database using table `T-W`.
Topic frames. Another visualization of topics uses the concept of frames. A topic frame consists of a noun and a verb that are assigned to the same topic and appear close together in the same documents. Topic frames can be computed using `Token`, `Word` and `Topic`, grouping by topic ID and word IDs of the frame tokens, and counting the number of frames using the same words.
Topic time. To analyze how discussions in blogs evolve, TopicExplorer allows to visualize the number of tokes assigned to topics over time. This analysis is also directly supported by joining documents with topics, grouping by date and topic ID and then counting the tokens.

3 Conceptual Modeling Framework

Based on the method for translating probabilistic models to ERMs and our experiences with the development of the TopicExplorer system, we propose a first idea for a new data model driven development approach dedicated to big data analytics applications. Figure 3 visualizes the traditional and our proposed data model driven development approaches. Both address four different tasks, namely (A) gather the data sources and make them available to a probabilistic model, (B) run machine learning components, (C) integrate the data sources with the machine learning output and (D) build the application consisting of data management components and an interactive user interface.

The traditional approach addresses the tasks mainly in sequential order. The first three steps implement data mining process following the CRISP model [7], while the last step addresses standard application development.

The translation method [13, 14] from BN to ERM allows an alternative, data model driven approach. We assume that for a wide spectrum of machine learning problems abstract, readily developed BNs already do exist. Those BNs could be pre-translated to ERMs to build a library. Thus, conceptual information about the machine learning component is already available when integrating the data sources, task (C). The BN could be treated as just another data source. The entities corresponding to observed variables in the BN, including their respective relationships, have to be matched with those from other available data sources. The matching conceptually defines the interface between data sources and machine learning, task (A). Furthermore, translating the integrated ERM into a (relational) model for a big data framework, e.g. Flink [2] or Spark [3], conceptually defines the API between the output of machine learning and the rest of the application, task (B). Depending on the framework, the tasks (A) and (B) could be supported by generation of efficient code for interfaces to access the given data as well as machine learning implementations.

As a consequence, application developers just need knowledge about in- and outputs, and the relationships among the variables in the Bayesian model, but not about probabilistic distributions and dependencies. Thus, our new data model driven approach eliminates unnecessary complexity caused by a lack of compatible conceptual languages on both sides of machine learning and data management. Thereby, it makes the collaboration between both sides more direct and offers potential for optimization.

4 Related Work

There are several approaches that combine data management with machine learning, however, none of them reaches a comparable conceptual level like ERMs. Hazy [12] provides programming, infrastructure and statistical processing abstractions, the latter are based on Markov logic [6]. This requires a deeper understanding of the machine learning algorithms in order to combine them effectively with data management. Several approaches combine machine learning APIs with SQL [1–3, 9] or with their own declarative language [16].

Last, data management is combined with machine learning at the level of user interfaces. An recent example is scikit-learn [15], which enable users to quickly select data sources and try different algorithms. Both do not offer an easy way to integrate machine learning results with domain specific meta data. Our approach also contrasts with statistical programming languages and software like R and SAS that just offer programming APIs to data sources and machine learning algorithms.

5 Conclusion

Knowledge about the machine learning side of a project helps developers to build a big data analytics application. The subsequently proposed framework gives guidelines how to effectively build an integrated conceptual model that includes details about domain specific aspects as well as the machine learning side of a big data analytics application. Future work includes the implementation of the framework and optimizing efficiency when translating integrated conceptual models to a particular implementation.

References

1. Akdere, Cetintemel, Riondato, et al. The case for predictive database systems: Opportunities and challenges. In *CIDR*, pp. 167–174, 2011.
2. Alexandrov, Bergmann, Ewen, et al. The stratosphere platform for big data analytics. *The VLDB Journal*, 23(6):939–964, 2014.
3. Armbrust, Xin, Lian et. al. Spark sql: Relational data processing in spark. In *SIGMOD*, pp. 1383–1394, 2015.
4. Blei, Ng, and Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
5. Chen. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1):9–36, 1976.
6. Domingos and Richardson. Markov logic: A unifying framework for statistical relational learning. *Introduction to statistical relational learning*, pp. 339–371, 2007.
7. Han, Kamber, and Pei. *Data Mining: Concepts and Techniques.* MK Pub., 2011.
8. Heckerman, Meek, and Koller. Probabilistic entity-relationship models, prms, and plate models. *Introduction to statistical relational learning*, pp. 201–238, 2007.
9. Hellerstein, Ré, Schoppmann, et al. The madlib analytics library: or mad skills, the sql. *VLDB*, 5(12):1700–1711, 2012.
10. Hinneburg, Oberländer, Rosner, et al.. Exploring document collections with topic frames. in *CIKM* 2014, pp. 2084–2086, 2014.
11. Hinneburg, Preiss, and Schröder. Topicexplorer: Exploring document collections with topic models. In *PKDD*, Part II, pp. 838–841, 2012.
12. Kumar, Niu, and Ré. Hazy: Making it easier to build and maintain big-data analytics. *Communications of the ACM*, 56(3):40–49, 2013.
13. Rosner and Hinneburg. Translating bayesian networks into entity relationship models. In *35th Int. Conf. on Conceptual Modeling, ER*, 2016. to appear.
14. Rosner and Hinneburg. Translating Bayesian Networks into Entity Relationship Models, Extended Version. *ArXiv e-prints, 1607.02399*, July 2016.
15. Scikit. scikit-learn, 2014. Machine Learning in Python.
16. Sparks, Talwalkar, Smith, et al. Mli: An api for distributed machine learning. In *ICDM*, pp. 1187–1192, 2013.