DeepVizdom: Deep Interactive Data Exploration

Carsten Binnig[1,2] Kristian Kersting[1] Alejandro Molina[1] Emanuel Zgraggen[2]
[1] TU Darmstadt, Germany [2] Brown University, USA

ABSTRACT

We make a case for a new generation of interactive data exploration systems that seamlessly integrate deep models as first-class citizens into the data exploration stack. Based on three case studies, we argue that this not only enables users to gain much deeper insights into a broader range of data sets but also helps to improve the performance and quality of existing data exploration systems.

ACM Reference format:
Carsten Binnig[1,2] Kristian Kersting[1] Alejandro Molina[1] Emanuel Zgraggen[2]
. 2018. DeepVizdom: Deep Interactive Data Exploration. In *Proceedings of SysML, Stanford, CA, USA, February 2018 (SysML'18),* 3 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There exists an ever-growing set of data exploration tools that allow data scientists of to intuitively explore new data sets. For instance, systems like DICE [8], Vizdom/IDEA [6], or Tableau [15] provide visual interfaces to quickly skim through the new data sets and look for patterns. However, these tools typically only focus on simple aggregate queries over structured data, and thus their applicability is limited to a small set of potential questions and data sets.

Consider e.g. a medical doctor who is exploring a patient data set that contains information about the patient itself such as sex and age but also other information about the medication, potential diagnoses, as well as information about whether or not a particular treatment was successfully applied or not. With existing exploration tools, the doctor could easily investigate the data at hand and find out that patients with age over 50, when diagnosed with disease X, received two different treatments — A and B — and overall had a chance of 60% to be successfully cured. Moreover, by drilling down, she would find out that treatment B had a higher ratio of successfully cured patients than treatment A. During data exploration, however, the doctor might immediately have some more speculative questions in mind such as "What if all these patients would have received only treatment B?". Unfortunately, using existing data exploration tools, the doctor has no simple support to answer those questions. Furthermore, important information is often not only available in well-curated structured data sets but is often available in other formats such as texts (e.g., medical reports) or images (e.g., MRI scans).

Putting deep learning [9] into the data exploration stack seems to achieve the desired flexibility, at least at first sight. Deep learning has gained popularity and yielded astonishing results in a wide range of applications ranging from standard classification and regression tasks to more complicated tasks such as image or text understanding. However, deep models typically need to be trained explicitly with large amounts of data in advance before they can be used to solve a particular task and applied to an existing data set. This is in contrast to the main challenge of data exploration: the questions users might want to answer using data are not known in advance; i.e., users typically ask ad-hoc questions or refine queries about the data set while browsing the data. Thus a naïve integration of deep models into the data exploration stack would not be fruitful.

We therefore put forward a vision for a new generation of data exploration systems that integrate deep models in an efficient manner: We envision that (1) deep models are integrated in a seamless manner — i.e., the user is not aware of using or even building deep models at all — and (2) the class of models supported should support a wide range of different questions a user might have to efficiently support exploratory workloads. Interestingly, we also believe that (3) the integration of deep models also helps existing data exploration systems to improve their performance as well as the quality of computing approximate aggregate query results. To substantiate our vision, we present the results of three initial case studies where we have integrated deep models into our own interactive data exploration tool Vizdom/IDEA [6]. In the first case study (Section 2), we show how deep models can be used to enhance existing approximate query processing techniques that are used to enable interactive response times for data exploration. In a second case study (Section 3), we then discuss how deep models can be further leveraged to enable users to ask more complex inference-based "what-if" questions such as the ones discussed before without the need to explicitly build a model for each new question at hand. In the last case study (Section 4), we present an extension to interactively explore text data without the need to explicitly construct a curated structured knowledge base before. Finally, we conclude by discussing further avenues for integrating deep models into the data exploration stack and building a complete system for deep interactive data exploration.

2 CASE STUDY I: MODEL-BASED AQP

Several commercial and academic systems support interactive data exploration workloads. Different from classical analytical database systems such as SAP HANA, these tools typically use sampling techniques to return an approximate answer for simple aggregate queries without scanning the entire dataset [4]. That way, they can often better guarantee interactive response times even on large data sets. Examples in this category include BlinkDB [1], ApproXimate DB (XDB) [10] or Vizdom/IDEA [6] as well as commercial systems such as SnappyData [12]. However, existing sample-based approximate query processing (AQP) approaches fall short under varying circumstances. Consider e.g. random sampling. It can be used to

SysML'18, February 2018, Stanford, CA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

support simple ad-hoc queries efficiently. Unfortunately, random samples may yield results of low quality with missing groups or large confidence intervals per group if the user wants to explore the tail of a distribution or analyze outliers as the sample might not include any of those data items. At the same time, stratified samples can better support rare cases but need to know the workload in advance to include the appropriate data items in the sample, which is not a realistic assumption for exploratory data analysis where the workload is not known in advance. Furthermore, another problem that existing sample-based approaches face is data quality. Existing AQP techniques rely on the fact that the data is cleaned; e.g., missing values are replaced and data entry errors are corrected etc. If not, existing techniques might return estimates of low-quality [17].

Deep generative models can be used to tackle these problems since they learn to represent the probability distribution over multiple variables from a given data set. While classical deep neural networks may discard any information unrelated to the desired prediction, deep generative models learn to represent the data in its entirety by capturing its underlying distribution. As such, they can naturally be applied for data exploration and also help to tackle other significant problems such as imputing missing values, repairing damaged data or detecting anomalies or even for constructing stratified samples online for a given selected subset of the data following the original data distribution. We have therefore integrated a first prototype model-based AQP engine in IDEA, which uses sum-product networks (SPNs) — a particular class of deep generative models [13] — instead of data samples to compute approximate aggregate query results. SPNs are particularly interesting for AQP not only because of the properties mentioned above but also since they can be estimated without deciding a-priori the parametric form of the random variables while still being expressive enough to approximate any distribution effectively and permitting efficient learning and tractable inference [11], e.g., for computing confidence intervals on the results. We have implemented two query processing strategies, one which uses an SPN to analytically compute the result of simple aggregate queries using the network architecture itself. A second query processing strategy, that can be used for arbitrary queries, creates stratified samples for a selected sub-population (e.g., the male patients over 50) using the SPN on the fly. Initial results show that we can compute approximate results with higher quality faster than existing purely sample-based AQP strategies.

3 CASE STUDY II: WHAT-IF QUERIES

As an extension of the previous model-based AQP engine, we have implemented a second case study which allows users to formulate speculative " What-if" queries. The main idea is to use the ability of generative models to "predict" any attribute or to generate new samples from the learned data distribution.

For example, as outlined in the introduction, the doctor might want to know the average success rate for male patients over 50 "if all these patients would have received only treatment B?". To this aim, we could use the SPN to either answer this query directly or sample a set of new data items for this speculative query from the model to compute an approximate answer. Moreover, the SPN could also be used as a "normal" prediction model. For example, the doctor might want to know the best treatment for a new incoming patient. In that case, she might want to predict the medication based on the patient data such as gender, sex, and age.

4 CASE STUDY III: TEXT EXPLORATION

A typical problem in many domains is that user want to get a high-level overview of a text corpus by deriving some aggregated information over those documents. Imagine a medical doctor who starts at a new hospital and wants to get an overview of all the medical reports. For example, she might want to know the average age of patients with different diagnosis from the texts. Knowledge base construction (KBC) from text data is a long-standing problem [14, 18] where the structured information that should queried is extracted from the text corpus. A major problem of all the existing KBC approaches is, however, that they do not support on-the-fly KBC construction and thus require well curated extract-transform-load pipelines that fill a structured database from the unstructured content. To that end, existing KBC approaches do not support ad-hoc exploratory analysis where we do not know the queries the user might want to ask in advance.

We have therefore extended our IDEA backend such that it can run ad-hoc queries such as "`SELECT AVG(age), disease FROM text_reports GROUP BY disease`" that computes an approximate aggregate result to visualize a histogram of patient ages per disease over a given text corpus without the need to construct a curated knowledge base before. The main idea is that we rely on existing open information extraction models [2] to derive a sparse table with the required attributes for the given query on-the-fly while streaming over the text documents and apply approximate query processing (AQP) techniques to answer simple aggregate queries such as the ones mentioned before. One problem clearly is data quality and in particular missing values that need to be taken into account. Another issue is how to sample from the text documents to get a representative set of entries in the sparse table. Moreover, very different from existing AQP approaches, and a key challenge in our case is that we do not know the population (i.e., the number of instances for each entity type) in advance, which makes it hard to estimate an over all aggregate value (such as COUNT or AVG). In our initial prototype we therefore rely on domain knowledge (e.g., one patient data set per medical report) but we also plan to use more clever statistical estimators such as the ones presented in [5] for this purpose.

5 FUTURE DIRECTIONS

The main future avenue is to integrate all our initial results into one joint system, called DeepVizdom, and generalize them to support seamless deep interactive data exploration. One should also investigate other components in the data exploration stack not considered here where deep models can be seamlessly be integrated to provider richer data exploration capabilities such as supporting deep query translation. Examples are natural language [3] or chat-bot like data exploration interfaces [7] that allow novice users to ask complex questions without knowing the schema of the data set or the technical details of a query language. Other interesting future avenues are the integration of an automated statistician, able to predict the statistical types and parametric forms of variables on-the-fly [16], and to learn from user interactions to better support information needs or to provide additional model-based explanations that help users to understand query results better.

REFERENCES

[1] S. Agarwal et al. Blinkdb: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 29–42. ACM, 2013.

[2] G. Angeli et al. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354, 2015.

[3] C. Binnig et al. Voice-based data exploration: Chatting with your database. In *SCAI@ICTIR*, 2017.

[4] S. Chaudhuri et al. Approximate query processing: No silver bullet. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 511–519, 2017.

[5] Y. Chung et al. Estimating the impact of unknown unknowns on aggregate query results. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 861–876, 2016.

[6] A. Crotty et al. Vizdom: Interactive analytics through pen and touch. *PVLDB*, 8(12):2024–2027, 2015.

[7] R. J. L. John et al. Ava: From data to insights through conversations. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[8] N. Kamat et al. Distributed and interactive cube exploration. In *ICDE*, pages 472–483. IEEE, 2014.

[9] Y. LeCun et al. Deep learning. *Nature*, 521:436 – 444, May 2015.

[10] F. Li et al. Wander join: Online aggregation via random walks. In *ACM SIGMOD*, pages 615–629. ACM, 2016.

[11] A. Molina et al. Mixed sum-product networks: A deep architecture for hybrid domains. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[12] B. Mozafari et al. Snappydata: A unified cluster for streaming, transactions and interactice analytics. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[13] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 337–346, 2011.

[14] X. Ren et al. Building structured databases of factual knowledge from massive text corpora. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1741–1745, 2017.

[15] P. Terlecki et al. On Improving User Response Times in Tableau. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, 2015.

[16] I. Valera and Z. Ghahramani. Automatic discovery of the statistical types of variables in a dataset. In *Proceedings of the 34th International Conference on Machine Learning, (ICML)*, pages 3521–3529, 2017.

[17] J. Wang et al. A sample-and-clean framework for fast and accurate query processing on dirty data. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 469–480, 2014.

[18] C. Zhang et al. Deepdive: declarative knowledge base construction. *Commun. ACM*, 60(5):93–102, 2017.