Automating Exploratory Data Analysis via Machine Learning: An Overview

Tova Milo
Tel Aviv University
milo@cs.tau.ac.il

Amit Somech
Tel Aviv University
amitsome@mail.tau.ac.il

ABSTRACT

Exploratory Data Analysis (EDA) is an important initial step for any knowledge discovery process, in which data scientists interactively explore unfamiliar datasets by issuing a sequence of analysis operations (e.g. filter, aggregation, and visualization). Since EDA is long known as a difficult task, requiring profound analytical skills, experience, and domain knowledge, a plethora of systems have been devised over the last decade in order to facilitate EDA.

In particular, advancements in machine learning research have created exciting opportunities, not only for better facilitating EDA, but to fully automate the process. In this tutorial, we review recent lines of work for automating EDA. Starting from recommender systems for suggesting a single exploratory action, going through kNN-based classifiers and active-learning methods for predicting users' interestingness preferences, and finally to fully automating EDA using state-of-the-art methods such as deep reinforcement learning and sequence-to-sequence models.

We conclude the tutorial with a discussion on the main challenges and open questions to be dealt with in order to ultimately reduce the manual effort required for EDA.

ACM Reference Format:
Tova Milo and Amit Somech. 2020. Automating Exploratory Data Analysis via Machine Learning: An Overview. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD'20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3318464.3383126

1 TUTORIAL BACKGROUND & SCOPE

Exploratory Data Analysis (EDA) is an essential process performed by data scientists in order to examine a new dataset up-close, better understand its nature and characteristics, and extract preliminary insights from it. Typically, EDA is done by interactively applying various analysis operations (such as filtering, aggregation, and visualization) – the user examines the results of each operation, and decides if and which operation to employ next.

EDA is long known to be a complex, time consuming process, especially for non-expert users. Therefore, numerous lines of work were devised to facilitate the process, in multiple dimensions [10, 24, 25, 31, 42, 46]: First, simplified EDA interfaces (e.g., [25], Tableau[1]) allow non-programmers to effectively explore datasets without knowing scripting languages or SQL. Second, numerous solutions (e.g., [6, 14, 22]) were devised to improve the interactivity of EDA, by reducing the running times of exploratory operations, and displaying preliminary sketches of their results. Last, several more systems simplify the query formulation for non-expert users (e.g., [15, 24]), as well as introducing query-by example systems (e.g. [8, 41]) that allow non-technical users to specify their information needs by selecting a set of example tuples.

While these works greatly facilitate the exploration process in terms of response time and ease of use, EDA is still a difficult process, as the user remains in the "driver's seat", having to decide which exploratory operation to employ at each step of the exploratory process. Therefore the scope of this tutorial is an emerging body of research works that are aimed at automating EDA, whether partially or fully. We start by surveying recommender-systems designed for EDA, aimed to assist users in choosing the next exploratory step to take, or suggest dataset segments and tuples that are likely to be of interest. We then discuss how machine learning techniques can improve upon such EDA recommender systems, by predicting users' interests in order to generate better, more personalized recommendations. Finally, we explain how deep learning methods can be used to fully automate EDA i.e. by auto-generating entire exploratory sessions and visualizations, hereby significantly reducing the time and effort data scientists dedicate to EDA.

Tutorial Outline. The proposed tutorial begins with a brief introduction to modern-day EDA, overviewing commonly

[1] https://www.tableau.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMOD'20, June 14–19, 2020, Portland, OR, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6735-6/20/06…$15.00
https://doi.org/10.1145/3318464.3383126

used interactive EDA interfaces, methodology and optimization tools. Then, the core of the tutorial consists of the following three modules, which are depicted in more details in Section 2. The first module describes recommender systems designated for EDA, whereas the latter two modules explain how automated EDA can be vastly improved using machine learning and deep learning methods. Table 1 summarizes selected works along the three modules.

1. EDA Recommender Systems: Data-driven, Log-based, and Hybrid Systems. The first section surveys the major developments in the field of recommender systems dedicated to EDA that provide the user with suggestions for specific, high-utility exploratory operations. Such systems can be roughly divided into two categories: (1) *data-driven* (also known as discovery-driven) systems, which use heuristic notions of *interestingness* and employ them, e.g., to find data subsets conveying interesting patterns ([12]), data visualizations [46], and data summaries [43]. (2) *Log-based* systems [1, 13, 49] leverage a log of former exploratory operations, performed by the same or different users, in order to generate more personalized EDA recommendations. Last, hybrid methods such as [29, 31] allow effectively utilizing both the log and the dataset currently being explored.

2. ML for Predicting Users' Interestingness Preferences. Measuring the interestingness of exploratory operations is a crucial component in many of the EDA systems described above, as well as for fully automating EDA (as described further below). Many heuristic measures have been proposed for assessing the interestingness of analysis operations, each capturing a different facet of the broad concept [16, 27]. However, interestingness is often subjective [7] and dynamically changing, even in the same exploratory session [28]. We examine two lines of recent works that utilize machine learning techniques to solve the problem: (1) *Dynamic selection of interestingness measures*, in which systems such as [28] predict which measure most accurately captures the user's interest, and (2) *ML-based models for users' interest* using, e.g., active-learning [10, 18] and *learning-to-rank* [26] techniques.

3. Fully-automated EDA Using Deep Learning Methods. While the recommender systems and interestingness prediction techniques described above accelerate and improve the exploration experience, to ultimately reduce the manual effort in EDA, it has been recently suggested to fully automate the exploration process, relying on advancements in deep neural networks. In particular, [9] presents a system for auto-generating data visualizations based on a *sequence-to-sequence recurrent neural network* model. Also, [2, 30] suggest generating entire EDA sessions, given an input dataset, that capture dataset highlights and interesting aspects. This is done by formulating EDA as a control problem, and solving it using *deep reinforcement learning*. Such generated sessions, when presented in EDA notebooks, allow allow users to gain preliminary insights on their dataset, before they begin exploring it themselves [3].

As automated EDA is an emerging field of research, it conveys many exciting opportunities for future work, as well as challenges and open questions. We review questions such as *How to make the auto-generated sessions personalized, reactive to users' information needs? How to design a better user experience and interface for automated EDA? How to build an effective, reproducible, experimental framework to evaluate the quality of auto-generated sessions?*

Tutorial Duration. The tutorial is 1.5 hour long. It begins with an introduction to EDA, in which we review commonly used EDA interfaces, and present common data models and required definitions. We then overview selected works in each module: EDA recommender systems, predicting users' interestingness, and finally - we brifly touch recent works in fully automated EDA. We conclude the tutorial with an elaborate discussion on the potential future endeavors of automated EDA.

Target Audience. The tutorial is primarily intended for researchers, students, and industry practitioners interested in data exploration and automatic knowledge discovery, but also addresses those who are interested in employing machine learning tools for data management applications. The tutorial is self-contained and requires no prior knowledge except for some familiarity with basic databases and machine learning terminology. Advanced techniques such as deep reinforcement learning and sequence-to-sequence networks are explained from the basics.

Related Tutorials. A comprehensive tutorial on data exploration [19] was presented in SIGMOD '15. It surveyed multiple lines of works, all dedicated to enhance the exploration process, from adaptive database storage and indexing mechanisms, to gesture-based database interfaces. In contrast, our tutorial is focused particularly on more recent developments in *automated EDA*, and the only overlap is in the introductory part of our tutorial, where we briefly overview selected materials from [19].

Also, [32] and [33], presented in VLDB '17 and SIGMOD '19, focus solely on systems for *query by example*, which allow non-technical users to circumvent query languages, and express their information needs by selecting example tuples. Such systems are out of the scope of this tutorial.

2 DETAILED OVERVIEW

As mentioned above, our tutorial begins with an introductory overview on modern EDA, briefly covering example use

Module	System Type	Exploration Type	Personalization
EDA Recommender Systems	Data-Driven	Tuples Recommendation [11], Data Cube/OLAP [20, 38], Visuazliations [46, 48]	No
	Log-Based	SQL [13] OLAP [1, 17, 49],	Yes
	Hybrid	Generic EDA [29, 31]	Yes
Predicting/Modeling Users' Interest	Dynamic Measure Prediction (kNN-based Classification)	Generic EDA [28]	Yes
	Modeling (Active Learning)	SQL/ Tuples Recommendations [10, 18]	Yes
	Modeling (Learning-to-Rank)	Visualizations [26]	No
Fully-Automated EDA	Seq2seq RNN	Visualizations [9]	No
	Deep Reinforcement Learning	Generic EDA [2, 30]	No

Table 1: Overview of works on automated EDA

cases, commonly used interfaces and optimization tools. We next describe the three core modules of our tutorial, then the concluding discussion.

2.1 EDA Recommender Systems

The first module of our tutorial surveys recommender systems designated for EDA. Such systems typically reside between the dataset and the EDA interface, and used for recommending high-utility EDA operations (e.g., visualizations, drill-downs, OLAP/SQL queries, and general EDA suggestions), without relying on any particular input or specifications from the user. We examine two major types of EDA recommender systems, *data-driven* and *log-based*, then how they can be combined in *hybrid systems*

Data-driven EDA recommendations. These systems derive recommendations relying solely on the data at hand (either the entire dataset or the results set of the last operation employed by the user). These tools use a notion of *interestingness*, defined a-priori, which is employed to grade the utility of the results of a specific EDA operation (e.g., OLAP drill-down [20] and roll-ups [39], data visualization [46], and data summaries [43]) or directly evaluate the interestingness of specific tuples [12] or data cube subsets [37, 38]. For example, [38] defines a notion of *surprisingness* that favors data cube cells with unexpected values, and in [46] the authors define visualizations that demonstrate a larger "deviation" from the original dataset.

Recommendations are typically generated by efficiently searching the space of EDA operations (supported by the system), pruning operations with low interestingness scores.

Log-based EDA recommendations. Such systems take as input the current state of an ongoing EDA session of a certain user, and generate recommendations for promising exploratory operations to be performed next, based on (previous) sessions of the same or different users. Drawing inspiration from *collaborative-filtering* recommender systems, the log-based EDA recommender systems, such as [1, 13, 17, 31, 49], utilize a repository of previous exploratory sessions, under the assumption that if two session *prefixes* are similar, their *continuation* is likely to also be similar.

To generate recommendations given an *ongoing* user's session, the system first retrieves the top-k most similar session prefixes from the repository, using a dedicated similarity measure for exploratory sessions. Then, it analyzes the continuation of the retrieved prefixes and uses the gathered information to construct recommendations for the next step in the ongoing user session.

Hybrid EDA recommendations. The data-driven and the log-based approaches each have a significant drawback: data-driven systems, relying only on the dataset at hand, produce generic recommendations with limited personalization to the end user, whereas log-based systems, examining the entire session of the current user, offer more personalized recommendations yet rely on the existence of high-quality, relevant previous EDA sessions that may not be fully available (e.g., since the goals and datasets of different users rarely overlap). To that end, the *hybrid* EDA recommender system presented in [31] attempts to bridge the gap between log-based and data-driven systems. Just like log-based systems, it uses the session log to retrieve similar prefixes to the ongoing session, and harvests a set of candidate "next-actions". However,

rather than directly constructing recommendations for concrete EDA operations, the system *generalizes* the set of candidate next-actions into a set of *abstract-actions* (i.e., with some missing parameters). The abstract actions can be further instantiated to concrete recommendations of EDA operations using a data-driven approach, i.e., by selecting the concrete operations that yield high interestingness values according to a pre-defined interestingness measure.

2.2 ML for Predicting Users' Interest

As mentioned above, measuring the interestingness of EDA operations is an important component in EDA recommender systems, as well as for fully-automated EDA systems (as described next in Section 2.3). However, a multitude of interestingness measures are suggested in previous work (see [16, 27] for surveys), each capturing a different aspect of the broad concept, which means that an operation ranked as highly-interesting by one measure, may be ranked as non-interesting by a different measure. Therefore, it is challenging yet highly important to dynamically select the appropriate measure. To that end, we discuss two different approaches for tackling the problem using machine learning: (1) Dynamic selection of interestingness measures and (2) ML-based models for users' interest.

Dynamic Interestingness Measure Selection. By analyzing EDA sessions logs, the authors in [28] show that interestingness preferences not only vary across users and datasets, but they often change dynamically within the same user's session. To that end, given a predefined set of interestingness measures, they formulate the dynamic interestingness measure selection as a multi-class classification problem, and build a kNN based classifier that predicts which interestingness measure best captures the current user's interest in each step of an ongoing EDA session.

ML-based models for users' interest. Rather than using existing, predefined notions for interestingness, systems such as [10, 18, 26] are designed to directly build a model for users' interestingness preferences. For example, [10, 18] take an *active learning* approach, i.e., they harvest feedback on presented tuples ("interesting" or "not interesting"), and use it to construct a model for an individual user's interest and gradually improve its accuracy as more feedback is collected. Another example is [26], in which the authors use a *learning-to-rank* model to assess the quality of data visualizations. By utilizing a training dataset containing user-annotated data visualization, they build a learned ranking-model that is able to determine, given two data visualizations, which one is more interesting.

2.3 Fully-Automated EDA using Deep Learning

In the last module of the tutorial, we survey very recent lines of work [2, 3, 9, 30], with the potential to ultimately reduce the manual effort devoted to EDA, by fully automating the exploration process. We introduce this novel approach by examining two different frameworks: one for automatically generating visualizations based on supervised learning, and another framework that employs deep reinforcement learning to auto-generate complete exploratory sessions.

Auto-generated data visualizations. The authors in [9] consider the problem of auto-generating visualizations as a *translation problem*, between data specifications (i.e., fields an values in a JSON format) to Vega-Lite [40] visualization specifications. To that end, they use a corpus of Vega-Lite specifications [35], and build a supervised translation model using an *encoder-decoder* network architecture: The encoder is a bi-directional recurrent neural network (RNN) that takes the data specifications as input and outputs a low-dimensional vector, and the decoder, also an RNN, takes the encoded vector as input and sequentially outputs Vega-Lite tokens (i.e., the visualization specification). The two networks are jointly trained to maximize the probability of generating a correct visualization specification given an input data specification.

Auto-generated entire EDA sessions. In [2, 3, 30], the authors propose a framework for automatically generating an entire EDA session given a dataset as input. The auto-generated session is presented to users in an EDA notebook [23], guiding them through the dataset's highlights and important characteristics, without requiring further input. To do so, the authors formulate EDA as a *control-problem*, i.e., by devising (1) a machine-compatible EDA interface, which provides parameterized, atomic composition of exploratory operations, as well as produces machine-readable encodings for their result sets; (2) an objective function that favors exploratory sessions that are: *interesting* (by employing a notion of *interestingness* as explained above), but also *diverse* (using a distance metric between actions' result sets) and *human understandable* (using an external classifier).

To solve the EDA control-problem, a dedicated *deep reinforcement learning* (DRL) network architecture is proposed, which can effectively handle the large number of possible EDA operations in each state of the session. The network is trained by self-interactions only, without requiring any labeled training data.

2.4 Concluding Discussion: The future of automated EDA

We conclude the tutorial with an elaborate discussion about the future challenges and opportunities in the emerging field of automated EDA. Example open questions are:

How to personalize automatic EDA? Current fully-automated systems for EDA (as described in Section 2.3) do not generate personalized output. However, since different users have different information needs, it is important to develop solutions for the personalization and interactivity of the auto-generated exploratory sessions. For example, the auto-generated visualizations system [9] could benefit from including a user preferences vector encoded alongside the data specification, and the DRL system for auto-generating EDA notebooks [3] can be enhanced with a personalized component to its objective function. Additionally, utilizing systems for predicting and modeling user interest, as presented in Section 2.2, may also be a promising direction for future work.

Which evaluation methods and benchmarks are adequate for EDA? EDA may be performed in different settings and in light of different exploration goals, therefore devising a standard for evaluation and benchmark automated EDA systems remains a challenge. The performance of EDA recommender systems are often measured using *predictive evaluation* (i.e., given a prefix of a user's EDA session, predict the subsequent operation performed). This is done using publicly available repository of EDA sessions, such as [44].

Evaluating the quality of entire EDA sessions [2, 3], as is the case for many generative models [45], is particularly challenging. To that end, an evaluation benchmark for auto-generated EDA sessions was developed by [3], drawing inspiration from existing methods to evaluate generative textual models (e.g., Inception Score [36] for image generation, BLEU score [34] for machine translation, CIDEr [47] for image description, etc).

How to facilitate richer (but safer) automatic EDA? Users often perform EDA using several types of EDA operations (e.g., data manipulation, visualizations, mining) [**?**], and across several datasets. Automatic EDA should support different types of operations, as well as work on multiple datasets. The latter could be supported by integrating systems for finding related/joinable datasets such as [5, 51]. Also, as the field of automated EDA develops, it becomes increasingly important to protect users from false-discoveries (See, e.g., [50])

Can we develop completely "Hands-free" system for EDA? While the automated EDA systems hold great promise, considerable effort is required to make the auto-generated output accessible and fully comprehensible to the end-user, as well as dynamically responsive to users requests and commands – as recently described in the visionary paper in [4]. A full-fledged "hands-free" EDA systems should be enhanced with means for voice commands (inspired by [21]) to allow for an easy, interaction with the system. An additional important feature is providing an explanation for the auto-generated exploratory steps, describing particularly what is interesting in each results-screen/visualization, hereby explicitly surfacing potential insights.

3 PRESENTERS BIOGRAPHY

Tova Milo is a full professor in the school of computer science in Tel Aviv University, and holds the Chair of Information Management. Her research focuses on large-scale data management applications such as data integration, semi-structured information, Data-centered Business Processes and Crowd-sourcing, studying both theoretical and practical aspects. Tova served as the Program Chair of multiple international conferences, including PODS, VLDB, ICDT, XSym, and WebDB, and as the chair of the PODS Executive Committee. She served as a member of the VLDB Endowment and the PODS and ICDT executive boards and as an editor of TODS, IEEE Data Eng. Bull, and the Logical Methods in Computer Science Journal. Tova has received grants from the Israel Science Foundation, the US-Israel Binational Science Foundation, the Israeli and French Ministry of Science and the European Union. She is an ACM Fellow, a member of Academia Europaea, a recipient of the 2010 ACM PODS Alberto O. Mendelzon Test-of-Time Award, the 2017 VLDB Women in Database Research Award, the 2017 Weizmann award for Exact Sciences Research, and of the prestigious EU ERC Advanced Investigators grant.

Amit Somech is a 5th year Ph.D student in Tel Aviv University under the supervision of Professor Tova Milo. His research focuses primarily on the facilitation of interactive data exploration, with the ultimate goal of making it more accessible to non-expert users. Prior to his studies, Amit practiced data science and big data analytics in the industry, and still engages in consulting work in these areas. He has published in multiple international conferences and journals including SIGMOD, VLDB, KDD, ICDE, CIKM and EDBT.

REFERENCES

[1] J. Aligon, E. Gallinucci, M. Golfarelli, P. Marcel, and S. Rizzi. A collaborative filtering approach for recommending olap sessions. *ICDSST*, 2015.

[2] O. Bar El, T. Milo, and A. Somech. Atena: An autonomous system for data exploration based on deep reinforcement learning. In *CIKM*, 2019.

[3] O. Bar El, T. Milo, and A. Somech. Automatically generating data exploration sessions using deep reinforcement learning. In *SIGMOD*, 2020.

[4] O. Bar El, T. Milo, and A. Somech. Towards autonomous, hands-free data exploration. In *CIDR*, 2020.

[5] F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire. Data polygamy: the many-many relationships among urban spatio-temporal data sets. In *SIGMOD*, 2016.

[6] A. Crotty, A. Galakatos, E. Zgraggen, C. Binnig, and T. Kraska. The case for interactive data exploration accelerators (ideas). In *HILDA*. ACM, 2016.

[7] T. De Bie. Subjective interestingness in exploratory data mining. In *Advances in Intelligent Data Analysis XII*, pages 19–31. Springer, 2013.

[8] D. Deutch and A. Gilad. Qplain: Query by explanation. In *ICDE*. IEEE, 2016.

[9] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications*, 39(5):33–46, 2019.

[10] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Aide: An active learning-based approach for interactive data exploration. *TKDE*, 2016.

[11] M. Drosou and E. Pitoura. Ymaldb: a result-driven recommendation system for databases. In *ICDT*, 2013.

[12] M. Drosou and E. Pitoura. Ymaldb: exploring relational databases via result-driven recommendations. *VLDBJ*, 22(6), 2013.

[13] M. Eirinaki, S. Abraham, N. Polyzotis, and N. Shaikh. Querie: Collaborative database exploration. *TKDE*, 2014.

[14] O. Elbaz, T. Milo, and A. Somech. Incremental based top-k similarity search framework for interactive-data-analysis sessions. In *EDBT*, 2020.

[15] J. Fan, G. Li, and L. Zhou. Interactive sql query suggestion: Making databases user-friendly. In *ICDE*. IEEE, 2011.

[16] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *CSUR*, 2006.

[17] A. Giacometti, P. Marcel, and E. Negre. *Recommending multidimensional queries*. Springer Berlin Heidelberg, 2009.

[18] E. Huang, L. Peng, L. D. Palma, A. Abdelkafi, A. Liu, and Y. Diao. Optimization for active learning-based interactive database exploration. *Proceedings of the VLDB Endowment*, 12(1):71–84, 2018.

[19] S. Idreos, O. Papaemmanouil, and S. Chaudhuri. Overview of data exploration techniques. In *SIGMOD*. ACM, 2015.

[20] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. Interactive data exploration with smart drill-down. In *ICDE*, 2016.

[21] R. J. L. John, N. Potti, and J. M. Patel. Ava: From data to insights through conversations. In *CIDR*, 2017.

[22] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed and interactive cube exploration. In *ICDE*. IEEE, 2014.

[23] M. B. Kery, M. Radensky, M. Arya, B. E. John, and B. A. Myers. The story in the notebook: Exploratory data science using a literate programming tool. In *CHI*, 2018.

[24] N. Khoussainova, Y. Kwon, M. Balazinska, and D. Suciu. Snipsuggest: Context-aware autocompletion for sql. *Proc. VLDB Endow.*, 4(1):22–33, Oct. 2010.

[25] T. Kraska. Northstar: An interactive data science system. *PVLDB*, 11(12), 2018.

[26] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. ICDE, 2018.

[27] K. McGarry. A survey of interestingness measures for knowledge discovery. *The knowledge engineering review*, 20(1):39–61, 2005.

[28] T. Milo, C. Ozeri, and A. Somech. Predicting "what is interesting" by mining interactive-data-analysis session logs. In *EDBT*, 2019.

[29] T. Milo and A. Somech. React: Context-sensitive recommendations for data analysis. In *SIGMOD*, 2016.

[30] T. Milo and A. Somech. Deep reinforcement-learning framework for exploratory data analysis. In *AIDM (at SIGMOD)*, page 4, 2018.

[31] T. Milo and A. Somech. Next-step suggestions for modern interactive data analysis platforms. In *KDD*, 2018.

[32] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. New trends on exploratory methods for data analytics. *PVLDB*, 10(12), 2017.

[33] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. Exploring the data wilderness through examples. In *SIGMOD*, 2019.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*. Association for Computational Linguistics, 2002.

[35] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum*, volume 36, pages 353–363. Wiley Online Library, 2017.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[37] S. Sarawagi. User-adaptive exploration of multidimensional data. In *VLDB*, 2000.

[38] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of olap data cubes. In *EDBT*, 1998.

[39] G. Sathe and S. Sarawagi. Intelligent rollups in multidimensional olap data. In *VLDB*, 2001.

[40] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016.

[41] T. Sellam and M. Kersten. Cluster-driven navigation of the query space. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1118–1131, 2016.

[42] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *PVLDB*, 10(4), 2016.

[43] M. Singh, M. J. Cafarella, and H. Jagadish. Dbexplorer: Exploratory search in databases. *EDBT*, 2016.

[44] A. Somech and T. Milo. React: Interactive data analysis recommender system. https://github.com/TAU-DB/REACT-IDA-Recommendation-benchmark, 2018.

[45] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models, 2015.

[46] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: efficient data-driven visualization recommendations to support visual analytics. *PVLDB*, 8(13), 2015.

[47] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[48] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG*, 2016.

[49] X. Yang, C. M. Procopiuc, and D. Srivastava. Recommending join queries via query log analysis. In *ICDE*, 2009.

[50] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska. Controlling false discoveries during interactive data exploration. In *SIGMOD*. ACM, 2017.

[51] E. Zhu, D. Deng, F. Nargesian, and R. J. Miller. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *SIGMOD*, 2019.