Research Statement

Matteo Brucato

matteo@cs.umass.edu

My research focuses on augmenting data management systems to better support *data science*, all stages of *analytics*, with special emphasis on *predictive* and *prescriptive* analytics, and faster and easier *decision making*. The main goal of my research is to *democratize* data science and decision making in a variety of practical applications, across different disciplines and industries. In my research conducted so far, I built complete and efficient data management systems able to support a broad class of decision-making problems that can be expressed as integer linear programs (ILP), on both certain and uncertain data. My current and future research aims at enlarging the scope of these systems to even broader classes of problems. Given the complexity and the broad scope of my research endeavors, my work has been largely interdisciplinary, spanning also other research areas such as natural language processing [17] information retrieval [19], text summarization [30], AI and robotics [6], it led to numerous top-tier publications and it has been recognized by various ACM awards.

Data grows at a much faster pace than the ability of existing data management systems to make full use of it. And while it grows, so does our awareness of the missed opportunity of its exploitation, pushing firms, organizations and associations to demand more efficient systems, able to perform increasingly complex operations. Embarrassingly, today's systems offer little to no support for decision making when the data is large enough for the decisions to be actually useful [32, 10, 35, 43, 7]. In business applications, making the most out the available data often means finding the most profit-maximizing decisions, which is crucial to ultimately achieve a company's growth objectives. This decision-making process travels through the whole Business Analytics pipeline, typically involving four steps: (i) Descriptive analytics, which allows businesses to analyze their massive data sets, through visualizations, summarization, and complex statistical analysis; (ii) Diagnostic analytics, which tries to explain "why" we see certain patterns, trends, or errors in the data, via complex explanation and error diagnosis tools, causality, and provenance analysis; (iii) Predictive analytics, which makes use of the insights of diagnostic analytics to build models that can predict "what is going to happen" in the future; (iv) Prescriptive analytics [9], which reaches the final goal of using predictions to make better decisions and answer the question "what to do now?" Unsurprisingly, as you move along the analytics pipeline, the complexity increases, and unfortunately, the existing support for its efficient execution decreases. My research focuses on the harder side of the analytics spectrum, hitherto mostly disregarded by modern data management systems.

In my dissertation research, I have laid out the first, most fundamental steps towards practical decision-making systems able to make use of large data sets that are crucial for modern applications. My research has revolved around a new class of database queries that naturally embody a fundamental category of constrained optimization problems: integer linear programs (ILP). The embodiment of an ILP, in a relational database setting, is what we called a "package query" [31, 20, 12, 21, 15, 13]. A table stores all the possible decisions that are available to the decision maker, in the form of records, and the package query defines which sets of records form a feasible solution, and a criterion to identify good solutions. For example, a tuple may represent an investment option, and a package query allows the decision maker to express an objective (e.g., selecting an investment portfolio that maximizes the revenue) as well as the constraints that limit the available options (e.g., budget constraints). Package queries connect interesting business applications with the data used to make decisions, in a very natural way. While package queries capture many of the interesting decision-making problems businesses face today, there are more complex ones that fall outside of their scope. I believe that the lessons learned on package queries can fundamentally change the future of research for creating scalable systems for decisions making for larger and more diverse classes of problems. My ultimate goal is to bridge the gap between the optimization problems that can most positively affect society and the ability of data management systems to support them.

1 Selected Research Accomplishments

Declarative and Scalable Prescriptive Analytics in Relational Data

In my dissertation research, I introduced "*package queries*" [14, 20], a new query model that extends traditional database queries to handle complex constraints and preferences over answer sets, allowing the declarative specification and efficient evaluation of a significant class of constrained optimization problems–integer programs–within a relational database. While traditional database queries define constraints that each record in the result must satisfy, package queries also require a *collection* of result records to satisfy constraints collectively, rather than individually. These combinatorial optimization problems arise in a variety of real-life application domains, such as coordinating fleet and crew assignments in airline scheduling to reduce delays and costs [41], managing delinquent consumer credit to minimize losses [36], crowdfunding optimization [11], optimizing organ transplant allocation and acceptance [5], planning of cancer radiotherapy treatments [42, 45], product bundles, course selection [39], team formation [8, 34], text summarization [30], vacation and travel planning [27], and computational creativity [40]. Many of these problems can be expressed as *integer linear programs* (ILP). ILP solutions alone account for billions in US dollars of projected benefits within each of these and other industry sectors [26].

Furthermore, some of these optimization problems are stochastic. Suppose each row in a table contains a possible stock trade an investor can make: whether to buy one share of a certain stock, and when to sell it back. Building a *financial portfolio* is an example of a stochastic optimization problem. Given uncertain predictions for future stock prices based on financial models derived from historical data, suppose an investor wants to invest $\$1,000$ in a set of trades (decisions on which stocks to buy and when to sell them) that will maximize the *expected future gain*, while ensuring that the *loss* (if any) will be lower than \$10 with probability at least 95%. Despite the clear need, modeling and solving these problems have relied on application-specific solutions [8, 27, 34, 39, 40], which can often be complex and error-prone, and fail to generalize. Package queries are a unified solution to enable *declarative* and *scalable* Prescriptive Analytics close to the data. The goal of my thesis research was to create a domain-independent, declarative and scalable approach for enabling Prescriptive Analytics, supported and powered by the system where the data relevant to these problems typically resides: the database.

Declarative specification and semantics of packages

SQL enables the declarative specification of properties that result tuples should independently satisfy. However, it is difficult to specify global constraints, such as the maximum total amount of money the investor is willing to invest when buying a set of trades. Expressing this SQL query is complex and inefficient. Our Package Query Language (PaQL) is a simple extension to SQL to support constraints and objectives at the package level. PaQL maintains the declarative power of SQL and its ability to express non-package constraints, while extending its expressiveness to allow for the easy specification of packages. PaQL is at least as expressive as Integer Linear Programming (ILP), which implies that evaluation of package queries is NP-hard in data complexity. sPaQL (the Stochastic Package Query Language) [18, 21] further extends PaQL to declaratively support *expectation* and *probabilistic* constraints or objectives on stochastic data. Drawing from the stochastic programming literature [4, 25, 28], we also introduced translation rules to express stochastic constraints and objectives into equivalent integer (generally, non-linear) constraints.

Evaluation of package queries

Due to their combinatorial complexity, package queries are harder to evaluate than traditional database queries [29], and we proved that package queries are as hard as integer programs. Existing database technology is ineffective at evaluating package queries, even if one were to express them in SQL. To overcome this, we extended the database evaluation engine to take advantage of external tools, such as ILP solvers, which are more effective for combinatorial problems. The core of this approach is based on the translation rules that transform a package query to an equivalent integer linear program. Unfortunately, such exact translations do not exist for the vast majority of stochastic package queries, for which the most general approach consists on using *approximate Monte Carlo translations* [38, 23, 25, 28]). We brought these approximate techniques

inside a probabilistic database [44] based on the Monte Carlo data model [33], and improved them to support queries involving large input tables [21].

Performance and scaling to large data sets

Integer programming solvers have two major limitations: they require the entire problem to fit in main memory, and they fail when the problem is too complex (e.g., too many variables or too many constraints). We overcame these limitations through sophisticated evaluation methods that allow solvers to scale to large data sizes. The core of our evaluation strategy, SketchRefine, consists of separating the package computation into multiple stages, each with small sub-problems, which the solver can evaluate efficiently. We proved that SketchRefine guarantees a $(1+\epsilon)$-factor approximation compared to Direct, where $\epsilon \geq 0$ is a flexible user-defined error parameter. A parallel version of SketchRefine [14] can also efficiently solve queries that require most of the partitions to be accessed. Monte Carlo methods often require the generation of many scenarios in order to produce feasible and close-to-optimal solutions. The data size creates new challenges for stochastic package queries: the number of required scenarios grows very fast with the size of the input table [25, 24]. Even with relatively small tables (order of thousands of tuples), the minimum number of scenarios required to have any meaningful guarantee explodes to impractical sizes that no solver can handle. Our technique, SummarySearch [21], compresses a large set of scenarios into a small set of "summaries", so that the solver can find feasible and close-to-optimal solutions very efficiently. We proved that if the state-of-the-art technique can find a $(1+\epsilon)$-approximate solution—compared to the optimal solution obtainable without summaries—in m iterations, SummarySearch is guaranteed to find a $(1+\epsilon)$-approximate solution as well in at most $3m$ iterations.

2 Current and Future Research

Enlarging the capabilities of Package Query Systems

In my thesis research, I introduced a new class of database queries—package queries—that can model linear optimization problems, with both deterministic and stochastic data. As a new area within database, my research opens a plethora of new research directions. Hereby, I summarize the main ones I plan to work on as part of my future research endeavors.

SketchRefine and SummarySearch solve two orthogonal problems: SketchRefine [31, 16, 14] deals with deterministic package optimization on large tables; SummarySearch [21] solves stochastic optimization that require too many scenarios. While the need for too many scenarios is a direct consequence of an increased data set size, it can typically happen even with relatively small sizes. When the data set size explodes, stochastic optimization becomes even more prohibitive, and SummarySearch alone may not be sufficient any more. An important question is how to combine SummarySearch with SketchRefine, for an end-to-end solution for stochastic problems at a very large scale.

In prior research [21], I developed methods to support individual probabilistic constraints, but some stochastic problems require several inner constraints to be satisfied *jointly*. Joint probabilistic constraints are a more challenging generalization of individual constraints. I plan to explore the applicability of existing techniques from the stochastic programming literature to the context of large tables, and to possibly extend SummarySearch, or develop new techniques for this challenging class of constraints.

In previous work, I addressed what is referred to as *single-stage* decision making under uncertainty, in which decisions have to be made before the values of the random variables become known. However, many applications require uncertainty to be revealed over time, i.e., in stages, allowing for remedial actions. These dynamic settings, referred to as stochastic programming with recourse, are more challenging to address than the single-stage setting. An important extension of my work is to study how to solve these complex problems at a large scale.

As part of my thesis, I created systems that sit on top of an existing DBMS. Putting the solver *inside* the DBMS is more challenging, as it requires a deeper integration with the system, such as: extensions to the relational algebra, and to query planning and optimization; automatic fine-tuning of the solver package; new fault tolerance mechanisms to deal with brittle solvers that fail due to unpredictable memory usage; support for nested queries, including complex joins before or after package-level constraints. As a first step

towards this, I plan to develop a deep integration of stochastic package queries into SimSQL [22], a database system for stochastic analytics. In SimSQL, the basic relational operators were engineered to deeply support Monte Carlo operations over relational data. In future research, I want to extend the operators to connect the system to optimization solvers.

In differential privacy, one seeks to publicly release the results of a statistical aggregation query, such as counting the number of cases a certain drug causes cancer, without giving away information about the individuals who took part in the statistic, i.e., whether a certain person has cancer or not. Accurate differential privacy is hard to achieve even for simple SQL aggregation queries. Computing the result of a package query on differentially private data poses new challenges. If packages are computed on the original data, how do we output a package that protects the input data from being inferred? As the solver typically explores a very large number of candidate solutions, it issues several SQL aggregation queries. Should the privacy budget be divided among all of the aggregate queries? Or can it be spent entirely on the final result reported to the user? In the first case, new challenges include: (1) maintaining differential privacy when the solver computes hundreds of thousands of aggregates over the same input data; (2) privatizing each aggregate can be computationally too expensive. If packages are computed on differentially private data, what are the effects on the quality of the returned packages?

Beyond packages: Data management systems for more general Decision Making

Decision making is a central component of nearly every aspect of our society. Modern applications require use of increasingly more data, rendering existing solutions inapplicable. As a result, approaches for decision making often simplify the problems so much that solutions are either infeasible or highly inaccurate. One of the main goals of a database system is to allow classes of computational problems (i.e., "queries") to be easily expressed and efficiently executed regardless of the size of the input data and the availability of special hardware. With package queries, we allowed support for decision-making problems that can be expressed as integer linear programs. In my future research, I plan to enable other, more complex kinds of applications.

A Markov Decision Process (MDP) is a decision-making framework where a decision maker (agent) has to decide what actions to take at each time step (policy), given that actions lead to uncertain outcomes (rewards). The agent's objective is to maximize the expected future reward. Problems that can be modeled as MDPs include reinforcement learning, robot planning, and self-driving cars, just to name a few, and appear in a broad range of domains, including finance, investments, agriculture, robotics, etc. My goal is to create a new data-oriented system for efficiently and scalably solving MDPs. The main challenges include: simple and declarative languages, close to the data, for expressing large and complex state spaces for MDPs; efficient and scalable algorithms for solving large MDPs [6]; support for evolving MDPs, where the agent's environment change over time, and the underlining MDP needs to be updated.

Robotics pose unique data management challenges. Consider a robot that needs to act autonomously in the world for long periods of time, without human intervention or without access to powerful computers. For example, a robot operating on Mars has to wait for several minutes before receiving a response back from Earth, due to the sheer distance between the two planets. Further, the sheer amount of sensor data of modern robotics applications are very demanding. For example, an autonomous vehicle, can produce more than 30 GB/hour of data [37]. For a robot to be autonomous, it must be equipped with the ability to store a large amount of sensor data (from its walks, interactions, and findings), create complex knowledge representations about the world, and use it to solve large analytics, optimization and decision-making problems, in order to make autonomous plans for the future.

As machine-learned software becomes more ubiquitous and accessible to decision makers who can impact our society, and robots more and more part of our everyday life, there will be a growing need for systems that ensure *fair*, *responsible*, and *sustainable* solutions. There is growing interest to develop Socially Responsible Investments (SRI) [1, 2], in line with the 2015 UN Sustainable Development Goals [3]. For example, in SRI, investors are not only interested in reducing the risk of a loss, but also that their investments meet the ESG [47] (Environmental, Social and Governance) standards. Machine learning models can exhibit undesirable behavior [46], from financial loss and unfair classification and predictions, to automated systems and robots that could potentially harm humans. Typically, a lot of training data is required to increase the confidence on the good behavior of the resulting model. Thus, an important question is how to efficiently train a model that requires complex constraints on large training sets.

References

[1] ESG investing: Where your money can reflect what matters to you. `https://investor.vanguard.com/investing/esg/`.

[2] Principles of Responsible Investment. `https://www.unpri.org/`.

[3] The SDG Investment Case. `https://www.unpri.org/sdgs/the-sdg-investment-case/303.article`.

[4] Shabbir Ahmed and Alexander Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 261–269. Informs, 2008.

[5] Oguzhan Alagoz, Andrew J. Schaefer, and Mark S. Roberts. *Optimizing Organ Allocation and Acceptance*, pages 1–24. Springer, Boston, MA, 2009.

[6] Anonymized. Anonymized title (under review). In *ICRA*, 2021.

[7] Arindam Banerjee, Tathagata Bandyopadhyay, and Prachi Acharya. Data analytics: Hyped up aspirations or true potential? *Vikalpa*, 38(4):1–12, 2013.

[8] Adil Baykasoglu, Turkay Dereli, and Sena Das. Project team selection using fuzzy optimization approach. *Cybernetic Systems*, 38(2):155–185, 2007.

[9] Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *arXiv preprint arXiv:1402.5481*, 2014.

[10] Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *Management Science*, 2019.

[11] **Matteo Brucato**, Azza Abouzied, and Chris Blauvelt. Redistributing funds across charitable crowdfunding campaigns, 2017.

[12] **Matteo Brucato**, Azza Abouzied, and Alexandra Meliou. Improving package recommendations through query relaxation. In *Proceedings of the First International Workshop on Bringing the Value of Big Data to Users (Data4U 2014)*, page 13. ACM, 2014.

[13] **Matteo Brucato**, Azza Abouzied, and Alexandra Meliou. A scalable execution engine for package queries. *SIGMOD Rec.*, 46(1):24–31, May 2017.

[14] **Matteo Brucato**, Azza Abouzied, and Alexandra Meliou. Package queries: efficient and scalable computation of high-order constraints. *The VLDB Journal*, Oct 2018.

[15] **Matteo Brucato**, Azza Abouzied, and Alexandra Meliou. Scalable computation of high-order optimization queries. *Commun. ACM*, 62(2):108–116, January 2019.

[16] **Matteo Brucato**, Juan Felipe Beltran, Azza Abouzied, and Alexandra Meliou. Scalable package queries in relational database systems. *PVLDB*, 9(7):576–587, 2016.

[17] **Matteo Brucato**, Leon Derczynski, Hector Llorens, Kalina Bontcheva, and Christian S. Jensen. Recognising and interpreting named temporal expressions. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 113–121, Hissar, Bulgaria, September 2013. Incoma Ltd. Shoumen, Bulgaria.

[18] **Matteo Brucato**, Miro Mannino, Azza Abouzied, Peter J. Haas, and Alexandra Meliou. sPaQLTooLs: A stochastic package query interface for scalable constrained optimization. In *VLDB*, 2020.

[19] **Matteo Brucato** and Danilo Montesi. Metric spaces for temporal information retrieval. In *European Conference on Information Retrieval*, pages 385–397. Springer, 2014.

[20] **Matteo Brucato**, Rahul Ramakrishna, Azza Abouzied, and Alexandra Meliou. PackageBuilder: From tuples to packages. *PVLDB*, 7(13):1593–1596, 2014.

[21] **Matteo Brucato**, Nishad Yadav, Azza Abouzied, Peter J. Haas, and Alexandra Meliou. Stochastic package queries in probabilistic databases. In *SIGMOD*, 2020.

[22] Zhuhua Cai, Zografoula Vagena, Luis Perez, Subramanian Arumugam, Peter J Haas, and Christopher Jermaine. Simulation of database-valued markov chains using simsql. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 637–648. ACM, 2013.

[23] Giuseppe C Calafiore and Marco C Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[24] Marco C Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[25] Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009.

[26] Der-San Chen, Robert G Batson, and Yu Dang. *Applied integer programming: modeling and solution*. John Wiley & Sons, 2011.

[27] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. Automatic construction of travel itineraries using social breadcrumbs. In *HyperText*, pages 35–44, 2010.

[28] Tito Homem de Mello and GÃijzin Bayraksan. Monte Carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56 – 85, 2014.

[29] Ting Deng, Wenfei Fan, and Floris Geerts. On the complexity of package recommendation problems. In *PODS*, pages 261–272, 2012.

[30] Anna Fariha, **Matteo Brucato**, Peter J. Haas, and Alexandra Meliou. SuDocu: Summarizing documents by example. In *VLDB*, 2020.

[31] Kevin Fernandes, **Matteo Brucato**, Rahul Ramakrishna, Azza Abouzied, and Alexandra Meliou. Package-builder: Querying for packages of tuples. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 1613–1614, New York, NY, USA, 2014. ACM.

[32] Davide Frazzetto, Thomas Dyhre Nielsen, Torben Bach Pedersen, and Laurynas Šikšnys. Prescriptive analytics: a survey of emerging trends and technologies. *The VLDB Journal*, pages 1–21, 2019.

[33] Ravi Jampani, Fei Xu, Mingxi Wu, Luis Leopoldo Perez, Christopher Jermaine, and Peter J Haas. MCDB: A Monte Carlo approach to managing uncertain data. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 687–700. ACM, 2008.

[34] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. In *SIGKDD*, pages 467–476, 2009.

[35] Katerina Lepenioti, Alexandros Bousdekis, Dimitris Apostolou, and Gregoris Mentzas. Prescriptive analytics: A survey of approaches and methods. In *International Conference on Business Information Systems*, pages 449–460. Springer, 2018.

[36] William M. Makuch, Jeffrey L. Dodge, Joseph G. Ecker, Donna C. Granfors, and Gerald J. Hahn. Managing consumer credit delinquency in the us economy: A multi-billion dollar management science application. *Interfaces*, 22(1):90–109, 1992.

[37] Oscar Moll, Aaron Zalewski, Sudeep Pillai, Sam Madden, Michael Stonebraker, and Vijay Gadepally. Exploring big volume sensor data with vroom. *Proceedings of the VLDB Endowment*, 10(12):1973–1976, 2017.

[38] Arkadi Nemirovski and Alexander Shapiro. Scenario approximations of chance constraints. In *Probabilistic and randomized methods for design under uncertainty*, pages 3–47. Springer, 2006.

[39] Aditya G. Parameswaran, Petros Venetis, and Hector Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM TOIS*, 29(4):1–33, 2011.

[40] Florian Pinel and Lav R. Varshney. Computational creativity for culinary recipes. In *CHI*, pages 439–442, 2014.

[41] Russell A. Rushmeier and Spyridon A. Kontogiorgis. Advances in the optimization of airline fleet assignment. *Transportation Science*, 31(2):159–169, 1997.

[42] Otto A. Sauer, David M. Shepard, and T. Rock Mackie. Application of constrained optimization to radiotherapy planning. *Medical Physics*, 26(11):2359–2366, 1999.

[43] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263 – 286, 2017.

[44] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic databases, synthesis lectures on data management. *Morgan & Claypool*, 2011.

[45] J. M. Artacho Terrer, M. A. Nasarre Benede, E. Bernues del Rio, and S. Cruz Llanas. A feasible application of constrained optimization in the IMRT system. *IEEE Transactions on Biomedical Engineering*, 54(3):370–379, 2007.

[46] Philip S Thomas, Bruno Castro da Silva, Andrew G Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.

[47] Emiel van Duuren, Auke Plantinga, and Bert Scholtens. Esg integration and the investment management process: Fundamental investing reinvented. *Journal of Business Ethics*, 138(3):525–533, 2016.