The Case for Personal Data-Driven Decision Making

Jennie Duggan
MIT CSAIL
jennie@csail.mit.edu

ABSTRACT

Data-driven decision making (D3M) has shown great promise in professional pursuits such as business and government. Here, policymakers collect and analyze data to make their operations more efficient and equitable. Progress in bringing the benefits of D3M to everyday life has been slow. For example, a student asks, “If I pursue an undergraduate degree at this university, what are my expected lifetime earnings?”. Presently there is no principled way to search for this, because an accurate answer depends on the student and school.

Such queries are personalized, winnowing down large datasets for specific circumstances, rather than applying well-defined predicates. They predict decision outcomes by extrapolating from relevant examples. This vision paper introduces a new approach to D3M that is designed to empower the individual to make informed choices. Here, we highlight research opportunities for the data management community arising from this proposal.

1. INTRODUCTION

Data-driven decision making (D3M) is changing the game in every professional endeavor, including business, medicine, and education [17, 18, 19]. Here, experts collect and analyze data to make their work more effective. In particular, D3M has seen widespread adoption in industry, where companies use it in every step of their operation, from supply chain management [20] to carving out a competitive advantage [8]. For example, banks use deep, predictive analytics to tailor their mortgage offerings. Their models take into account the creditworthiness of the borrower, valuation of the property, and even current market conditions to calculate a mortgage’s interest rate and fees.

Unfortunately, these D3M tools are only available to skilled professionals. Their techniques often use sophisticated models and rely domain expertise. Also, they leverage well-known datasets, many of which are proprietary.

Because end users don’t have access to these sophisticated tools for their decisions, they instead resort to one-off specialized calculators, such as currency converters and retirement savings planners. These applications typically require manually entered data, making them difficult to use and error-prone. Owing to their limited interface, they are not deeply personalized, rather deriving their results from industry standard formulas using generic constants.

It is our belief that a large class of decision problems can be solved for less technical users using simple models with access to lots of data. This paper proposes *personal data-driven decision making* (PD3M), a novel generalization of domain-specific techniques. PD3M differs from prior approaches in two key ways: scope and accessibility. It is designed for *general* ad-hoc decision making, rather than tools tied to a specific domain. Also, the target audience for PD3M is *untrained users*, who do not have a background in data science. This framework applies models over relevant examples from the past to guide decision makers. In lieu of deep predictive analytics, PD3M executes relatively simple models on carefully targeted data.

Everyone makes decisions in domains where they are not experts, and PD3M will empower people to make informed choices. This framework models a personal choice using a *decision query*, which builds predictions by extrapolating from outcomes of similar decisions in the past. This query consists of *context*, a*model*, and *relevant data*.

We now examine a sample decision query in this framework. In a future world, everyone’s daily activities generate structured tables which are archived in a personal data store. A user asks, “How will buying this house impact my credit rating?”. His data store contains personal bank account statements. The engine determines that the context of his decision is his financial data and the house. It automatically queries the real estate listing, bringing in details about the house and its present value. The platform uses an off-the-shelf credit score model to make predictions. The query’s relevant data consists of people similar to the hypothetical house-owning Steve.

Context personalizes a decision query by supplying information about the scenario under which a decision is being made. People are accumulating data at a greater rate and precision than ever before. This is evidenced by the Quantified Self [14] movement, where individuals use technology - often automatically - to collect data about their life, such as exercise routines and genetic testing. It is easy to imagine users seamlessly supplying rich personal data to a decision query, thereby customizing its predictions.

A decision *model* codifies how a query creates predictions. It takes as input relevant data, learning from it to predict decision outcomes. The PD3M engine contains a library of models, such as linear regression and k-nearest neighbors.

Relevant data supplies the query with examples from which it projects outcomes. The asker uses a *data portal* or search engine to find information related to his decision and inserts the source into his query. Within the data portal, decision makers interact with *brokers*, who sell queries or subscriptions to their information. Numerous businesses are organizing privately held data for public querying, includ-

Proceedings of the VLDB Endowment, Vol. 7, No. 11
Copyright 2014 VLDB Endowment 2150-8097/14/07... $ 10.00.

ing InfoChimps, DataMarket, and Windows Azure Marketplace. In addition, there are open data initiatives, such as DataHub [1] at MIT.

A *selection query*, which is distinct from a decision query, identifies the subset of the data source from which the model will learn. This filtering chooses data that is most informative for the decision to create personalized, accurate predictions. The selection also controls the financial cost of a query when it accesses a paid data source [15] and implements the broker's privacy policies by ensuring that the model only reveals results at a sufficient level of aggregation.

Personal decision making poses many research challenges not present in expert D3M applications, and it is well-suited for multi-disciplinary exploration. Personalization and usability are paramount for PD3M owing to its audience. This presents an opportunity for our field to partner with members of the human-computer interaction community to find the best ways for novice users to model their decisions.

2. PERSONAL DECISION MAKING

This section enunciates the requirements of a PD3M system. It begins with an overview of the platform. Immediately following, it outlines the mechanics of building a decision query over structured data. A data model for this framework completes the design.

We illustrate this vision with a working example. Joe, a carpenter, is debating whether to keep or sell his truck, the Model Q. He queries, "What is the estimated value of my truck in n years?" To get an accurate estimate, the framework takes into account his commute, driving record, and auto maintenance habits.

2.1 Overview

Figure 1 displays the execution of a decision query. It begins with a natural language question posed by the user, which declares the decision's broad context and outcome sought. The user composes his query on a *dashboard*, where he interactively describes and explores his decisions. Using the query text, the dashboard proposes a specific decision context, including a list of relevant attributes. Based on the context and outcome, the system picks a dataset and model.

Once the query is specified, the engine generates an execution plan, or list of steps to create decision outcomes. This plan designates where each step of the query will be performed, and dispatches work to both the user-side client for visualization and servers providing relevant data for model evaluation. The query respects privacy policies and other limitations imposed by its participants.

2.2 Query Formulation

We now examine how a user poses a decision query, focusing on its three parts: context, model, and relevant data.

Context: A decision query is personalized by its context. It has two parts: an *anchor*, specifying the exact conditions under which a decision is being made, and any number of *similarity metrics* for comparing the anchor to relevant data.

A decision maker provides rich context using his personal data store [7, 11]. The data store contains both manually entered information and data automatically collected from emails, smartphones, and other sources. Frictionless sensors integrated into everyday interactions, like Joe's truck recording its engine's condition, will make personal archiving increasingly accessible. The user browses his archive using objects, such as the truck. He also draws from a Wikipedia-like library of well-known entities, including people, places, and things.

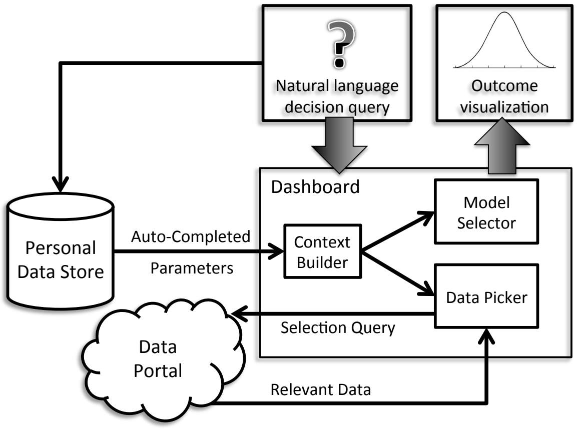

Figure 1: PD3M architecture

The anchor is typically an object from the user's archive. When Joe refers to "my truck", the database fills in the Model Q object, including its condition, mileage, and the behavior of its driver. The anchor may be modified for a proposed course of action, such Joe adding mileage to his commute if he anticipates taking a job at a distant location.

Similarity metrics shape the selection query and decision model execution. The metrics may be supplied by the user or derived statistically [21]. They quantify how closely relevant data resembles the anchor and parameterize the decision model, mapping database attributes to model inputs. The *context builder*, which determines an anchor and similarity metrics from the decision query text, brings about several novel challenges. We explore them in Section 3.1.

Decision Model: Presently PD3M is done using one-off tools, which have a narrow scope and limited personalization. A better approach supplies users with a library of models and recommends one suited for their decision.

A decision model is either general or domain specific (DS). Table 1 highlights examples of the former. DS models are usually written once by experts, and typically have complex calculations. Joe leverages a DS truck valuation model. Off the shelf offerings like this will make it easier for novices to learn how to use PD3M and help them provide meaningful context for their decisions.

Decision models are specified directly or assigned using a *model selector*. The selector infers that Joe is estimating a vehicle's value from his query text. It suggests a set of transformations to forecast his outcome. He wants to estimate the truck's value in n years. Hence, the PD3M engine models the truck's value now and at several earlier time steps. It then applies linear regression over the valuations to forecast the truck's future worth.

Relevant Data: Using the data portal, a decision query picks one or more sources from which it draws examples. To find a source, the user searches by example using the decision context. Joe's query would look for past valuations of the Model Q where it had drivers similar to himself. For a source to be relevant, it meets several expectations. The data is semantically correct, having information of the same category as the context. It is also complete, including both the decision context and outcome. Last, it is personally

Model Type	Example Query
Bin Packer	"Here are a list of my interests, the times I am available, and my budget. Suggest my schedule."
K-Nearest Neighbors	"What is my expected total cost of ownership if I get a bachelor's degree from this university?"
Probability Density	"I am planning a vacation in August, how many days each of sunny, cloudy, or rainy weather should I anticipate?"
Regression	"Are there likely to be more tech jobs in San Francisco or New York in 5 years?"

Table 1: Taxonomy of decision models

relevant, with values closely resembling the anchor.

After the user submits his context, the *data picker* returns a ranked list of potential sources, summarizing how well-suited each is to his decision. The decision maker browses proposed sources, viewing generic examples of their content to verify semantic correctness.

After selecting its data source, the PD3M engine executes a selection query, which produces a set of relevant data. The selection query negotiates the trade-off between robust modeling with more data and fine-grained personalization to the anchor.

2.3 Data Model

When the user specifies a decision's context, he uses *objects*. Recall that such objects are usually based on real-life examples, such as people and places. The context's anchor may be modified to suit a what-if decision or redacted to preserve privacy. Encapsulating context at this higher level of abstraction enables the user to easily create a complete decision scenario and smoothes the selection of relevant data. The context builder begins with the object, which it shares with the model selector and data picker. For the model execution, context is converted into a tuple with only the attributes needed.

Decision models supply a schema for their input and output. Their input is a mediated schema, spanning both the context and relevant data. Joe's query uses linear regression to estimate changes in his truck's value over time, hence the model expects dependent and independent variables and it applies the context, "in n years", for its prediction.

The query produces one or more outcomes paired uncertainty statistics. For Joe, this is a projected truck value, and a range taking into account the closeness of the relevant data to Joe and fluctuations in his vehicle's historical value.

Relevant data is structured, and will often remain on its source server. This expedites query time by reducing network transmission to aggregated results. It also simplifies the enforcement of a data broker's privacy policies and makes use of economies of scale by sending the query to a server, rather than executing the model on a personal computer.

3. RESEARCH OPPORTUNITIES

This section examines the open questions posed by a PD3M system to the data management community, with an emphasis on query personalization and ease-of-use.

We use an example query to demonstrate these research challenges. Sally is a high school student who was recently accepted to several universities. She wants to attend the school that will give her the highest projected lifetime earnings. The student starts by analyzing her prospects at the University of Enlightenment. She wants to make her decision based on the wages of students similar to her in high school GPA, standardized test scores, and class rank.

To focus on PD3M in this section, we consider a simplified version of the architecture in Figure 1. Here, the data portal uses Google Fusion Tables [12], which implements the more general data integration challenges. The system renders its visualizations using Tableau [2], because it has a robust language for displaying data. This strawman design is equipped with a basic set of models, such as the ones in Table 1.

3.1 Personalized Querying

One of the key contributions of this framework is *personalized* querying. Hence, correctly identifying and using context is crucial to realizing this system. Users may query by example [22], such as Sally asking about students like her. Based on the anchor, the database selects a set of attributes for use as similarity metrics.

One line of inquiry is selecting the most salient set of attributes for a context. Identifying the strongest predictors for the decision outcome is challenging, because not all context is independent. Finding the most informative *collection* of attributes is an open problem [6]. Sally's context builder suggests using her college major to refine its wage estimate. Another research direction is efficiently caching clusters of attributes associated with an outcome. This will speed up queries predicting the same decision for different users.

Formulating a selection query for relevant data is an important part of this personalization. This challenge navigates several competing, even contradictory, goals. The user wants enough data to have a statistically significant model, although its selection is tightly fitted to her circumstances. The selection is broad enough to protect the privacy of data broker clients. The query may be subject to limitations of the asker's budget. In addition, the decision model chosen influences the most useful samples of relevant data.

To a first approximation, this is a complex, multi-criteria decision. An integer linear program may address this challenge. This approach necessitates identifying rigorous yet usable ways for non-programmers and data brokers to convey their expectations as constraints and an objective function. The PD3M framework will formulate its solution over the available data; it determines the selection using integrated approach such as [9].

An orthogonal angle of this personalized querying is customized usability. Although user studies may provide us with general findings about how decision makers interact with their data, the PD3M engine is more effective if it is adaptive to the behavior of individuals. Sally cares about the full distribution of values in her outcome, whereas Joe prefers to focus on effect sizes. This prompts the database to tailor the output to patterns of use, rather than using the same visualization for all salary-related decisions.

In summary, we have identified three directions for query personalization. Attribute selection for context pinpoints the most promising predictors for decision outcomes. Creating the selection query finds the most useful examples for the decision model while catering to constraints. Last, adaptive interaction between the user and their decisions will present new research challenges.

3.2 Broad Challenges

We now examine several research challenges that need to be addressed to make the PD3M a reality. Security, privacy, choosing a decision model, data source selection, query composition, and outcome visualization are considered here.

Security & Privacy: For this framework, privacy is a n-way relationship, between the query writer, one or more data brokers, and the clients of the latter. Brokers must respect the privacy of their users. [3] models this relationship around the purpose of each query, and [4] programmatically verifies whether a privacy policy meets user expectations. This work does not consider a third dimension: under what conditions brokers may share information with a decision maker or other third party. This issue is complicated by two factors: whether the client is willing to sell his or her data, and the level of protection they expect. Some clients will share their data, but only if it is highly aggregated with that of others to maintain their anonymity. Others demand less protection, but for a higher price. Some may desire a sliding scale of privacy, proportional to the price. Articulating and negotiating these trade-offs requires more sophisticated privacy models.

Model Selection: Identifying models with which to project decision outcomes is an open research question. Sally's dashboard recommends a k-nearest neighbor model. One possible approach is to classify decision queries to models using machine learning. Similarly to [16], the framework starts by executing several models per query. It collects feedback from the decision maker to learn from examples of successful model mappings, refining its choices iteratively. There may be other approaches to this question.

Data Source Selection: A third outstanding issue for this framework is identifying relevant PD3M data sources. In its most general formulation, this is a web search problem. It has the added challenge of personalization; its data must closely resemble the decision's context. A first approach might search using keywords from the query text.

Provenance: If query output is not clear, the user "debugs" it using what we call soft provenance. This provenance conveys how the query created its predictions without miring the user in the details of the model. It also preserves the privacy policies of relevant data; hence Sally can only see the aggregate of her k-nearest neighbors, not the individual salaries. This will be a significant shift from prior work, which focused on skilled users having access to complete source data.

Query Composition: Ideally, the user poses her query using natural language. [13] identified several ways for non-programmers to compose scientific database queries, including a natural language, keyword searches, query languages, and filling out forms. All of these approaches generalize to the PD3M framework. In particular, the database community has a rich history of query language research [5, 10, 22] and is in a strong position to contribute here. There is also room for alternative solutions, such as an asker taking a "virtual highlighter' to her natural language query, annotating its parts for the query planner. Sally would mark her context (GPA, etc.), relevant data (graduated students), and outcome (lifetime earnings). User studies will be important for determining which approach(es) are most desirable.

Outcome Visualization: Finding the most informative way to present the results of a decision query is also an open question. The query's decision model may guide how its outcomes are visualized. Sally's k-nearest neighbors projects a salary paired with a range based on the relevant examples. There are several ways to project decision outcomes; statistical machine learning and user studies will identify the most promising strategies.

4. CONCLUSIONS

This vision paper proposes a novel data management system for personal data-driven decision making (PD3M). It begins by examining the components of a decision query, comprised of its context, model, and relevant data from which the model learns. The query generates one or more projected outcomes for the user. PD3M introduces numerous research opportunities for the data management community, especially in query personalization and database usability, making it fertile ground for future work.

5. ACKNOWLEDGMENTS

The author thanks Michael Brodie, Sam Madden, Michael Stonebraker, and Eugene Wu for reviewing earlier versions of this work. Their feedback and suggestions significantly strengthened its content.

6. REFERENCES

[1] *DataHub*. http://datahub.io/.

[2] *Tableau Software*. http://www.tableausoftware.com/.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *PVLDB 2002*.

[4] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Implementing P3P using database technology. In *ICDE 2003*.

[5] A. Alashqur, S. Y. Su, and H. Lam. OQL: a query language for manipulating object-oriented databases. In *VLDB 1989*.

[6] M. Anderson, D. Antenucci, V. Bittorf, M. Burgess, M. J. Cafarella, A. Kumar, F. Niu, Y. Park, C. Ré, and C. Zhang. Brainwash: A data system for feature engineering. In *CIDR 2013*.

[7] G. Bell. A personal digital store. *CACM*, 44(1):86–91, 2001.

[8] E. Brynjolfsson, L. Hitt, and H. Kim. Strength in numbers: How does data-driven decisionmaking affect firm performance? *Available at SSRN 1819486*, 2011.

[9] C. Bucilă, J. Gehrke, D. Kifer, and W. White. Dualminer: A dual-pruning algorithm for itemsets with constraints. *Data Mining and Knowledge Discovery*, 7(3):241–272, 2003.

[10] C. J. Date and H. Darwen. *A Guide to the SQL Standard*, volume 3. Addison-Wesley, 1987.

[11] J. Gemmell, G. Bell, and R. Lueder. MyLifeBits: a personal database for everything. *CACM*, 49(1):88–95, 2006.

[12] H. Gonzalez, A. Halevy, C. S. Jensen, A. Langen, J. Madhavan, R. Shapley, and W. Shen. Google fusion tables: data management, integration and collaboration in the cloud. In *SoCC*, 2010.

[13] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu. Making database systems usable. SIGMOD '07.

[14] K. Kelly and G. Wolf. http://quantifiedself.com/.

[15] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Query-based data pricing. In *PODS*, 2012.

[16] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan. Mlbase: A distributed machine-learning system. In *CIDR*, 2013.

[17] J. A. Marsh, J. F. Pane, and L. S. Hamilton. Making sense of data-driven decision making in education. Rand, 2006.

[18] A. McAfee and E. Brynjolfsson. Big data: the management revolution. *Harvard business review*, 90(10):60–66, 2012.

[19] D. L. Sackett. Evidence-based medicine. In *Seminars in perinatology*, volume 21, pages 3–5. Elsevier, 1997.

[20] G. Wang, S. H. Huang, and J. P. Dismukes. Product-driven supply chain selection using integrated multi-criteria decision-making methodology. *Int. J. of PE*, 2004.

[21] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2003.

[22] M. M. Zloof. Query-by-example: A data base language. *IBM Systems Journal*, 16(4):324–343, 1977.