Report of the Data Science Track

In Preparation for the Seattle DB Meeting, October 9-10, 2018

EXECUTIVE SUMMARY

Data science has emerged as a major interdisciplinary field, and has attracted much attention. In this short report we discuss what is data science, why it is important for us, and where we can contribute. We argue that data science is here to stay and will become even more important, and that we have a lot to contribute, but if we do not ramp up our efforts, we risk becoming increasingly irrelevant. In particular, we believe the time has come for us to develop a data science agenda that builds on our strengths, attracts broad participation from our community, and helps us (together with other communities) shape this emerging field.

Specifically, we have identified **the following (incomplete) list of recommendations:**

- Develop a strategic plan for data science that attracts broad participation from our community.
- Encourage sharing of best practices, challenges, and lessons learned (see an example in Appendix A).
- Identify high-impact *problems* in data science (DS) where our community can make significant contributions. Examples include data integration and wrangling, context management, knowledge base construction, building data repositories, and approximate query processing.
- Identify high-impact *solution techniques* where our community can make significant contributions. Examples include machine learning, visualization, and human-centric techniques.
- Devote more efforts to solving end-to-end problems, building industrial-strength systems, and connecting with real users and real data. For academic researchers, we recommend paying more attention to solving DS problems in domain sciences.
- At universities, DS is causing transformative changes in education and relationship with other communities. Identify what we want to, and should do, in these aspects.

We have also identify **the following (incomplete) list of discussion points** for the Seattle DB meeting:

- What is the magnitude of changes caused by DS? Is it just a new set of techniques, like machine learning? Is it a new emerging field? Are we just one of the many "components" of this field? Or a key player? To what extent is DS causing changes at universities and beyond?
- How do we make sure that our community engage in DS, develop a community-wide strategic plan, and regularly share best practices, challenges, and lessons learned?
- How do we identify and make significant real-world impacts on selected DS problems and techniques?
- What is the role of system building and working with real users? We have often acknowledged that these are important. But how do we go beyond acknowledgment to cause real changes in our community's attitude and work in these aspects?
- What is our strategic plan for DS at universities, to ensure that we contribute significantly and have a seat at the table?
- The statistics field is having an identity crisis. They believe that they have been working on DS for decades. But now they find other communities claiming to do the same and arguably having more real-world successes. How do we make sure that we will not run into the same kind of crisis in the future? What do we think is our "identity" in data science? If DS is an emerging interdisciplinary field, what part of this field will we "own", if any? And do we think this would still be so 30 years from now?

WHAT IS DATA SCIENCE AND WHY IS IT IMPORTANT TO US?

Currently there is no consensus definition for data science (DS). For our purposes, we will define data science as a field that develops principles, algorithms, tools, and best practices to manage data, focusing on three topics:

- analyzing raw data to infer insights,
- building data-intensive artifacts (e.g., recommender systems, knowledge bases), and
- designing data-intensive experiments to answer questions (e.g., A/B testing).

As such DS is clearly here to stay (even though the name may change), for the simple reason that everything is now data driven, and will only become even more so in the future.

The key property of data science is that it is at heart a cross-cultural discipline. It integrates methods in traditional data management with methods from fields focused on empirical science and *inferential reasoning*, including statistics, economics, AI and many of the social and hard sciences. As such, it spans mathematical foundations (logic and probability) as well as spans research methodologies and -- importantly! -- research cultures. Engaging meaningfully in data science by definition requires reaching out across areas, learning and teaching new ways of working.

Much has happened in the past ten years, across many different communities, to address and grow this new field:

- Many research communities, such as DB, AI, KDD, statistics, and more have been working on developing *data science techniques.*
- Many communities have been working on developing *data science tools,* both open source and proprietary. Examples include ecosystems of open-source DS tools in Python/R, and many companies/startups in DS.
- Many *domain sciences* are racing to develop and deploy infrastructures to capture data, techniques and tools to process data, and methodologies to do data-driven research. They also organize workshops, courses, and write textbooks to *introduce data-driven research into their communities.*
- Universities are introducing *DS courses, certificates, programs, and degrees.* The most popular programs have been MS degree in DS, DS certificates, and DS courses in domain sciences. Several universities start to introduce *undergraduate degrees in DS.*
- Universities are also working on *long-term strategic plans to address DS*. Such plans include reorganizations, courses and degrees, hiring of new faculty in DS, and introducing new organizations such as DS institutes. At universities where such plans have been lacking, many grassroot efforts have sprung up.
- *Funding agencies* have also worked on DS strategic plans (e.g., at NSF, NIH).
- There are also efforts to define and plan for the field at the highest academic level, such as recent *DS leadership summits involving many universities.*

Against this backdrop, it is clear that our community needs to engage, and engage much more than we have done so far. As the world becomes data driven, many communities have taken actions. As this report makes clear, we do have a lot to contribute, but if we do not ramp up our efforts, we risk becoming increasingly irrelevant. In particular, we believe the time has come for us to develop a data science agenda that builds on our strengths, attracts broad participation from our community, and helps us shape this emerging field (together with other communities).

COMPONENTS OF DATA SCIENCE

In what follows we discuss data science in terms of tasks, techniques, life cycle stages, and types of people. While we discuss these aspects separately, it is important to note that virtually everything in data science (including the processes, tasks, life cycle stages, etc.) is interleaved and iterative.

Tasks and Techniques

Many DS pipelines consist of two parts: data integration and data analysis. Here, data integration (a.k.a. data wrangling) is broadly interpreted as covering all major data preparation tasks, such as data acquisition, extraction, exploration, profiling, cleaning, matching, and merging. This topic is also known as data munging, curation, unification, fusion, preparation, and more.

DS tasks often use many techniques, such as machine learning, database methods, visualization, big data scaling, human-in-the-loop, crowdsourcing, approximate query processing, etc.

Life Cycle

We can broadly distinguish three phases in the life cycle of a DS project, from exploration to production:

1. In the first exploratory phase, single data scientists or small groups work (often with sample data and desktop tools) to assess and wrangle datasets, and often to build (and test) statistical models that can be used to describe the data and/or perform predictions. Some people would say that data science stops here, but that's a very narrow view.

2. Second, regular reports and predictive models are often then scaled up by "data engineering" teams who are responsible for ongoing testing of the pipelines that generate the outputs. Managing these recurrent pipelines of data generation and testing is a significant "big data" challenge enhanced with aspects of context/change management and other operational concerns. The term "DataOps" is being used by some industry analysts to talk about these issues. I view this is integral to the mission of data science.

3. Finally, high-value data science models are placed into production as live services -- e.g. recommender systems, fraud detection systems, and so on. In DBMS terms, these scenarios are more akin to transaction processing than to analytics.

It's not necessary that each DS project goes through these three phases. Some may stop at 1 or at 2 if the exploration turns out some insights there or if the end result that is desired is not necessarily something that looks like a service.

Types of People

There are multiple "personas" at play here, each of which leads to different top-down user-centric challenges for our community. The first phase consists of (at least) two classes of desktop data scientists. One class have deep domain knowledge and context on the data (e.g. they are "business users" who traditionally worked with complex spreadsheets and BI tools). The second class has more or less deep technical knowledge (e.g. they are "technical users" who work with SAS or R/Python and have some background in statistical modeling).

Phase 2 is typically managed by DevOps people, and the tools that they use come out of the DB, Big Data and Ops worlds: databases, queue management, workflow systems, metadata management, orchestration and containerization tools.

Phase 3 includes some of those same operational users, but the software that makes use of Phase 3 is often user-facing services like web and mobile apps written by traditional software developers calling into new “data-science-driven” APIs. There are research problems and commercial products targeted at each of these personas, and the problems and solutions vary widely across personas. When doing research on “data science” tools, it is critical to understand the users and use cases that are being addressed.

The Iterative Nature of DS

As mentioned earlier, data science is an inherently iterative process, with loopbacks at multiple granularities. Desktop data scientists frequently move through cycles of data discovery, profiling, querying, visualizing, transforming, and modeling, and from any one phase may find that they need to go back to one of the other phases --- for example, visualization or modeling may suggest that profiling was inadequate and more transformation was needed.

At a larger granularity, models in production may start becoming ineffective, which motivates ongoing loopbacks to the second phase of testing-at-scale on logs of real user input, and more occasional loopbacks to redesign of models in the first phase. Because everything is iterative, it’s challenging to keep context across phases and time -- metadata management (versions, logs, etc) becomes critical.

WHERE CAN WE CONTRIBUTE?

We can contribute to many aspects of DS. In what follows, we discuss contributions to a number of topics. Clearly, this is not a complete list, and more should be discussed at the Seattle meeting.

Data Integration and Wrangling

Data scientists repeatedly say that data wrangling is 80% of their challenge, that “once you wrangle the data, the rest is often obvious”. As such, data integration (DI) and wrangling is clearly a critical part of the DS process.

For these problems, we can contribute a lot. First of all, since we have been working on these problems for decades, we can bring to the table a good understanding of what the core problems in this area are, as well as what the challenges are. Second, while there has been significant recent progress in building data integration and wrangling tools, the current state of the art is still far from satisfactory. For example, there are still very few powerful integration and wrangling tools out there that are widely used.

To contribute, we should consider several important aspects. First, so far we have focused much of our effort on solving narrow "point problems". We should devote more effort toward solving end-to-end problems, i.e., those that go from raw data to an outcome desired by an end user.

Second, while we should continue to build on traditional data integration and metadata management techniques, many new topics have also come to the fore that have aspects of far-flung fields including predictive interaction and mixed-initiative interfaces, program synthesis, statistical data profiling and cleaning, second-order/nested query processing and search. Machine learning, which has received some

attention since the 1990s, has now made rapid progress and demanded renewed attention. These need narrow work, but even more we need to think about the synthesis of these ideas for solving end-to-end problems over time.

Third, we should devote more effort toward helping real users solve their data integration and wrangling problems. This requires us to go beyond the traditional convention of mostly developing algorithmic solutions, toward developing full-fledged systems, as well as using these systems and tools to help real users. A promising solution for academic researchers to work with real users and real data is to talk with domain scientists at the same university. Twenty years ago this might have been difficult, because most of the data was still at companies. But the situation has dramatically changed. At virtually any university now, often within a walking distance from the CS department, there are many domain science groups that are awash in data, with many pressing DI problems to solve.

Fourth, as we work with real users and real data, a natural need will come up to develop detailed guidance on how they can use tools as well as manual effort to solve their problems end to end. Such guidance in turn will help us identify true pain points in the data integration/wrangling process, to which we can devote more effort to develop algorithmic solutions.

Fifth, we need to devote more effort to developing data integration/wrangling systems and tools. Here, an interesting question is what forms would these systems take? Is there a single "uber" system that can solve most of integration/wrangling problems? Or would it be an ecosystem of interoperable tools? Should some of these tools be on-prem, while some others be on cloud? And how do we make this decision? What is the best way to develop and maintain such tools (e.g., in academia)?

Finally, before we can wrangle data, we need to find good datasets. There has been a lot of work on this recently, e.g., on open government data, datasets tagged with schema.org, HTML tables on the Web, and datasets inside enterprises. But a lot more work needs to be done here.

Context Management

At every phase of the data science lifecycle, it is key to understand the context of the “input” data, the processes working on the data, and the output data “products”. The *ABCs* of data context include *Application* metadata about what is being represented (including both traditional DBMS-like schematic models as well as statistical models of the process that generated the values), *Behavioral* metadata (including the lineage and usage of data and software), and *Change* over time (including versions of data, software, and output artifacts like models or charts). Tracking, integrating and analyzing this contextual metadata is a data science problem well suited to the data management community: it is largely ignored by both the algorithmicist/modelers and by the system builders, but it spans those areas as well as traditional database challenges.

Building Knowledge Bases and Data Repositories

Building knowledge bases continues to remain important in data science, as they capture domain knowledge and can be used in virtually all steps of the data science process. At the same time, there is a growing movement of building data repositories for domain sciences. In the past decade, domain scientists have created many data-centric communities, each of which maintains one or more data repositories, where scientists can submit, curate, and consume data.

For example, as a particular domain science X becomes increasingly data-driven, a set of enterprising scientists in X will band together to set up a data repository, where scientists worldwide can submit their data, use the data already in the repository, and help curate the data. These enterprising scientists often run annual workshops to educate fellow scientists in X on how to use and contribute to the data repository. Examples include the Environmental Data Initiative at environmentaldatainitiative.org for the environmental science community, the UMETRICS initiative at btaa.org/research/umetrics for the science policy community, and many more. Among the domain sciences, perhaps the biomedicine community has made the most progress. They have been advocating and implementing the notion of data commons (see https://commonfund.nih.gov/bd2k/commons), which are communities of users, datasets, tools, and infrastructures, all working together to generate, curate, disseminate, and consume data.

As such, data repositories are different from knowledge bases. They are more closely resembling "data lakes" at enterprises, but they are "data lakes" for domain sciences. Our community should devote more attention to working with domain sciences to help them build such data repositories. First, such data repositories can revolutionize domain sciences. Second, they provide real data and real users, and the data is often public. Third, there are often large-scale funding opportunities for working on these repositories. Finally, building them raises numerous data science challenges, including virtually all those that are discussed in this report.

Machine Learning and AI

From data science perspective, ML/AI is "just" a technique (inasmuch as many DS problems require inference from data), albeit a very important one. Thus, we should discuss “what problems need to get solved” as opposed to “how does ML affect this area”? Then in addressing the problems we can ask what the role is of, say, user input vs automation, and then within automation ask questions about how to do inference, optimization, etc. As one concrete example, program synthesis is an important aspect of data wrangling. Program synthesis is a search problem; sometimes it is solved brute force, sometimes with theorem provers, sometimes with AI methods. Here the choice of search heuristic is often less interesting than the formulation of the problem.

Another important issue to consider is that we often cannot just use ML/AI techniques in isolation to solve DS problems. Instead, we need to combine them with many other techniques, such as data validation, efficient user interaction, visualization, etc. How to effectively combine such techniques requires far more work. We also often need to scale up ML/AI techniques to large amount of data, and make them much more usable (e.g., with support for provenance, explanation, easy deployment, label debugging, etc.).

Natural language processing techniques are also getting increasing attention, given the growing amount of textual data in many data science settings. While our community has worked on some of these problems (e.g., natural language query interfaces for structured databases and for building data science pipelines), more effort is necessary here.

In the reverse direction, many DS problem settings require ML, often carried out over a large amount of data. As a result, we have many opportunities here to explore how we can develop effective data management techniques to help ML. Examples include data validation for ML systems, systems for ML, ease of programming for ML, real-time / online processing for ML and AI, accelerating neural network training and inference via data system techniques, end-to-end management of ML pipelines,

causality/explainability/debugging especially with respect to software spanning logic and statistics, managing model serving, and more.

Human-Centric Challenges

There is an emerging consensus that many data management problems require the user to be in the loop. In the past few years, our community has made significant progress on this topic, under the umbrella "human-in-the-loop data analytics" (HILDA) and elsewhere.

While individual problems and techniques should continued to be investigated, it is important that we also perform a systematic study of this topic, to obtain a broader view of the landscape, and to develop strategic plans. For example, this study can examine the types of data science problems that can benefit from human in the loop, the types of human users who can be involved (including the levels of technical sophistication and the roles in the data management process), the types of interfaces (e.g., traditional keyboard and mouse, touch screen, gesture, etc.), the types of techniques (e.g., visualization, learning, crowdsourcing), the types of actions that users can do (e.g., labeling, visual manipulation, providing textual feedback, etc.), and the types of data science systems to be built (e.g., on-prem, cloud-based).

Several issues deserve more attention. First, many real-world users have little computing knowledge. For them, developing *self-service data science tools* is critically important.

Second, the vast majority of current work still examines only HILDA issues related to *a single data tool*, e.g., how to design a data tool such that it can effectively solicit user feedback. In practice, to solve a problem end to end, users often must use *multiple tools*, as well as *manually* perform certain data processing steps. If we want to help users solve such problems end to end, we need to provide guidance to them on how to perform each step, which one can be performed semi-automatically using which tool, and which one needs to be be performed manually.

Third, and a related issue, we need to build systems and work closely with real users and data, to help them solve their problems and to evaluate our systems. While this point applies to all data management challenges, we believe it is even more important for human-in-the-loop ones. As mentioned earlier, domain scientists at the same university provide a promising initial set of real users with real data to work with.

Finally, to work effectively with humans (in the loop), we need to understand them thoroughly. What is easy/difficult for them in terms of data processing? What are their biases, weaknesses, strengths, preferences, etc. regarding data? There are many “rules of thumbs” and observations that have been mentioned in the literature. We should codify them, examine, and develop a coherent “theory of human data interaction”. Our community has done some recent pioneering work on this topic, but far more is necessary.

System Building and System Architecture, Open Source versus Closed Source

As discussed earlier in this report, we should devote far more effort to building realistic systems and using them to process real data for real users. Important questions include how to encourage more system building effort in our community, how to build systems in academia, and how to find real users and real data. We need to discuss these questions as well as encourage sharing experience and lessons learned about them.

Another important set of questions relate to what kinds of systems to build. Should we build monolithic stand-alone data science systems, or should we encourage building new ecosystems of interoperable data science tools, or building into existing ecosystems of such tools? Or a combination of all of these? Should we build on-premise systems? Cloud-based systems? Self-service systems? What class of systems interest us? Are we interested in targeting individual desktop data scientists or mostly interested in “Big Data” or multi-user problems? Or is the goal to bridge from one to the other? What are the pros and cons of each of these system types? How do they all relate to one another? If we are to build realistic systems and to work with real users, we need to thoroughly understand the types of systems we want to build, and why.

Even for existing ecosystems of data science tools, such as the Python/R ecosystems, we can discuss whether it’s better to address it via evolution (e.g. scaling up Pandas) or revolution (replace Pandas with a cleaner and more easily scaled algebra that integrated nicely into Python), and how we can get engaged.

Yet another important topic is open-source versus closed-source systems. Both types of systems have existed for some time for data scientists, and it is not yet clear what is the relationship between the two. As of now, it is clear that there are ecosystems of open-source data science tools that are growing rapidly and that have many users, such as the Python/R ecosystems. An argument can be made that since there are many users who use these tools, we should seek to help them too. Even in that case, we can still build either open-source or closed-source tools, as long as the tools can interoperate with other tools in the ecosystems. It is true that for academic researchers, building into open-source ecosystem is much easier, since they can quickly leverage and *customize* many open-source tools to build their own tools. But it is also true that there are many outstanding closed-source tools and that established companies may want to exploit both. As of now, the jury is still out on how critical the "open source" requirement is, for system development in our community.

Usability

A major concern here is how to make it very easy for end users to use our data science tools, where end users are not DBAs or programmers, but "common" people. This includes how to make it very easy for such users to quickly deploy data science tools, to use them, and to know when to use what tools.

A related concern is that there is too much fragmented data and too many tools. This concern is not new. It has been consistently raised in the past two decades. There has been an effort to consolidate the tools, to help reduce the complexity for end users. While we should continue that effort, we should also accept that in many data science settings, the user will have to use many tools (and even do certain steps by hand). The focus should then be to guide the users on when to use what tools, and to develop methods to help choose and configure the tools. Change management across the stack is another major challenge.

On usability issues, we need to think "outside the box" and "much closer to the user". There are many serious usability pain points that our community has not addressed. Take for example scalability. Suppose a user runs a data science tool on a single machine and it runs too slowly. At this point, what should the user do so that the tool can run faster the next time? Tuning the tool parameters? Getting a machine with more RAM? More cores? Try looking for a version of the tool that runs on a cluster of machines? Or what else? These are real usability questions that real users very commonly have, to which we have not paid sufficient attention.

Education

There is a great demand for learning DS at both undergraduate and graduate levels. Unfortunately currently we only have isolated course offerings, of varying contents. If we can (more or less) agree on a standard RDBMS curriculum (at least for undergraduates), we ought to be able to do the same for DS. In particular, current DS courses do not emphasize enough data wrangling (the stage where raw data is acquired, extracted, cleaned, transformed, and integrated), even though this step takes up to 80% of analysts' time. This is where our community (especially researchers in data cleaning / transformation / integration) can really contribute.

In general, we have done a good job training our students/workforce for RDBMSs. Can we do the same for DS? A huge number of non-CS students have also been flocking to CS departments wanting to learn DS. What and how should we teach them? Is it even our job? Many academic departments are also looking into setting up their own DS curriculum. What can we do to help them, and how?

One may argue that the above view is too DB-centric, and that DS is much wider than that and DS education is already happening (one more reason why we need to ramp up our effort, otherwise we risk becoming irrelevant). At UC-Berkeley, for example, there are 600+ undergrads signing up for data science major in Spring 2018, and they are teaching nearly 2000 per semester in the data science courses. It's not up to the DB community to make this happen … it is already happening, and will necessarily be much broader than our field. However we do need to help out or these programs will be missing key components: notably query languages, data wrangling, metadata and data modeling problems. And we'll keep seeing the poorly-reinvented stuff (Pandas, etc) replicate at the undergrad level. What is the best way for us to help out? What are the experience and lessons learned from places where this is happening?

Relationships with Other Communities

Many academic disciplines have growing DS needs, and have been looking to us for help. How can we help them with their research? With education? Should we explore providing DS services to other academic departments (e.g., consulting, DS tools, cloud-based DS services)? Should we play a leadership role in setting up Data Science Institutes at universities, and if so, what should these institutes do?

Our experience in the past few years suggests that while these academic disciplines are still looking to us for support, they have also been actively doing DS on their own: educating themselves in DS, setting up DS training programs, building domain-specific DS tools, etc. Likewise, while we can still contribute much to the various ecosystems of DS tools (e.g., in R/Python), they have also been growing rapidly without us. As a result, if we do not ramp up our efforts, and think strategically about what we want to do, we risk becoming increasingly irrelevant in this fast growing interdisciplinary field.

It is also important to note that the engagement with other communities can vary significantly depending on the university. At UW-Madison, database and machine learning people (both in CS and beyond) have been consulted widely in pushing DS initiatives (together with people in optimization, statistics, and some domain sciences). At UC-Berkeley, other groups have taken far more active roles, but the DB group is helping define curriculum, stay connected to practice outside academia, etc. Thus, in engaging with other communities, it is important to understand what we can contribute, and how. One thing we do offer here is an appreciation of problems in large and diversely-skilled enterprises, which differ from problems in the

sciences and from problems at high-tech companies. See discussion of personas above -- our community is traditionally connected to different personas than other branches of CS.

Other aspects that we can contribute in an university setting include (a) data quality (including data integration/wrangling; we are probably the community that is most active, and with most expertise, on this topic), (b) scaling methods (when there is a need to scale up querying, learning, visualization, etc., we often get consulted), (c) building data systems and tools, (d) providing courses and training materials on the above topics, and (e) supplying students well-trained in DS techniques to domain science teams.

Ethics

Data science is where the ethical questions come to the fore. We need to have that discussion here as a matter for education as well as tooling.

FURTHER POINTERS

- Data Science Leadership Summit Summary Report, by Jeannette M. Wing, Vandana P. Janeja, Tyler Kloefkorn, and Lucy C. Erickson. A summary of a DS workshop by 65 participants from 29 universities and 3 funding organizations in the US. Available at http://pages.cs.wisc.edu/~anhai/ds-report.pdf (please do not distribute).

- Recent twitter thread from DJ Patil on ways that data scientists could help with the storms in the Carolinas. Could be an intro or early subsection on "how data scientists think". A concrete example of the role, the need for agility, the need for diverse and raw datasets at various levels of aggregation, different modalities of data, the need to cooperate with domain experts and decision makers. One thing missing is that this is a one-shot scenario; we should also emphasize ongoing data pipeline scenarios.

- Realizing the potential of data science, by Francine Berman, Rob Rutenbar, Brent Hailpern, Henrik Christensen, Susan Davidson, Deborah Estrin, Michael Franklin, Margaret Martonosi, Padma Raghavan, Victoria Stodden, Alexander S. Szalay

- Joe Hellerstein has two public health case studies: one on the CDC using data science for diagnosing HIV/AIDS outbreaks in rural Indiana, and another at Kaiser Permanente using data science to predict flu outbreaks. We could borrow from these as well for motivation.

- 50 Years of Data Science, David Donoho. 2017.

APPENDIX A

To illustrate many points discussed in the above report, we present a case study of DS at the University of Wisconsin-Madison, and how some of us in the database group have been addressing it. It would be highly desirable to have more case studies discussed at the Seattle DB meeting.

Research

Starting around 2014, we recognized the importance of the emerging field of DS, and discussed what we could do to contribute.

- We quickly recognized that based on our background, we could contribute to the data integration, cleaning, and wrangling part of DS pipelines. Since then we have focused a significant amount of our research effort on these problems.

- In the past three years, we have also identified and started working on the challenges of scaling up machine learning pipelines, making them more useable, and integrating them with effective user-centric techniques. This is because we need to solve these challenges when trying to solve problems in data integration, cleaning, and wrangling.
- Finally, we also focus on system building and the challenges in building DS systems and tools.

At UW-Madison many people work on DS research challenges. Examples include theoretical ML problems, optimization, statistical techniques, low-level system issues, Big Data systems, etc. These people are also spread over many departments, including CS, ECE, statistics, iSchool, bioinformatics, etc. By working on the above problems, we aim to set ourselves apart, and carve out a clear space where we can make unique and complementary contributions.

Working with Real Users and Real Data

We made a conscious effort to work with domain sciences at UW-Madison. It turns out that many domain sciences need help in data integration/cleaning/wrangling. But we cannot just develop point solutions and hope that eventually they will use it. Instead, we attempt to help them solve problems end to end. To do so, we need a wide range of DS tools. Currently, we rely heavily on the Python ecosystem of open-source data science tools (PyData), and we build our tools into this ecosystem as well. We also develop guidance to use these tools to help domain scientists solve their problems end to end.

By working with domain scientists, we have learned a lot about data integration challenges, and have had many opportunities to evaluate our tools. We have also positioned ourselves well to apply for internal DS grants by UW-Madison, raising more than $750K in the past three years.

A major challenge is that our current model of working with domain scientists do not scale beyond working with 3-4 domain science teams. To scale further, we need to hire programmers, or look for a different model.

We have also been working closely on data integration/cleaning challenges with a number of companies, in Madison and elsewhere.

Education

UW-Madison is in the process of creating multiple MS degrees in DS (e.g., in CS, statistics, and ECE) and a BS degree in DS (joint between CS, statistics, and the iSchool). For the MS degrees, we are working to provide a set of courses. These include a course in relational data management and a course in data integration/wrangling. Both courses are very popular and have been incorporated into many MS programs in DS. For example, the MS program in DS in statistics requires both courses.

For the planned BS degree in DS, we plan to offer a simpler version of our undergraduate course on relational data management. We may also offer the data integration course as an elective.

A major component of the data integration course is to solve a data integration/cleaning problem end to end, using machine learning, PyData tool, and entity matching tools that we have developed. This project component has been hugely popular.

Contributing to Data Science Efforts at the University

At UW-Madison, we are known as people working on data quality and system challenges, as well as people who have been working well with many domain science teams. This helps us gain a seat at the table, as the university debates what to do in DS. Specifically:

- We contribute to grant proposals where data quality, scaling, and system challenges come up.
- UW-Madison is debating setting up a campus-wide DS institute. We can position ourselves as contributing expertise in data quality/curation, scaling, systems, and in working with domain sciences, so that we can contribute meaningfully and influence the directions of this new institute.
- In the meantime, many "mini" institutes in DS are being created, and we are involved in these, due to the above unique angles.
- The fact that we are contributing courses in data quality/scaling, which form an important part of many DS programs, also helps us gain a seat at the table.
- Many colleges and research groups are looking into building data lakes, data repositories, and into developing a comprehensive data storage/processing plan, and are asking us to help in these aspects.
- Within the CS department, working with colleagues in AI/ML/systems/optimization, we are pushing to grow a bigger group in data science.

In short, our experience has been that if we possess a unique set of expertise (in research, education, etc.) and are willing to offer it to others, we can meaningfully participate in campus-wide DS efforts and gain a seat at the table.