Advanced Data Management (CSCI 490/680)

Machine Learning in Databases

Dr. David Koop
Checking Computational Results in Systems

Figure 4: Process by which the study was performed.

[Collberg and Proebsting, 2015]
Reasonable e

the papers either
to build the code, but it may have required extra e
and, within
dby code we either got a negative response to our email requests, or no response within two months.

Table 2, Figure 11, and Appendix B show the results of the study. Table 4 lists the abbreviations

4 Results

green numbers papers that are weakly repeatable, red numbers papers that are non-weakly repeat-

10.

Notes:

(a)

st sure it was not there.

[Collberg and Proebsting, 2015]
Excuses for not sharing

- Versioning
- Available Soon
- No Intention to Share
- Personnel Issues
- Lost Code
- Academic Tradeoffs
- Industrial Lab Tradeoffs
- Obsolete HW/SW
- Controlled Usage
- Privacy/Security
- Design Issues

[Collberg and Proebsting, 2015]
Reproducibility in Computer Science

- Repeat the experiment in reproducibility!
- Differences from original
- Shows issues with trying to classify experiments

---

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purported Not Building; Disputed; Not Checked</td>
<td>6%</td>
</tr>
<tr>
<td>Purported Building; Disputed; Not Checked</td>
<td>2%</td>
</tr>
<tr>
<td>Conflicting Checks!</td>
<td>0%</td>
</tr>
<tr>
<td>Misclassified</td>
<td>1%</td>
</tr>
<tr>
<td>Purported Not Building But Found Building</td>
<td>14%</td>
</tr>
<tr>
<td>Purported Building But Found Not Building</td>
<td>0%</td>
</tr>
<tr>
<td>Purported Not Building; Confirmed</td>
<td>0%</td>
</tr>
<tr>
<td>Purported Building; Confirmed</td>
<td>0%</td>
</tr>
<tr>
<td>All Others Purported Not</td>
<td>27%</td>
</tr>
</tbody>
</table>
Reproducible Research

• Science is verified by replicating work independently

• Replication Issues:
  - Requires many resources to replicate (Sloan Digital Sky Survey)
  - Requires significant computing power (Climate Model Simulation)
  - Requires too much time or very specific circumstances (Environment Epidemiology)

• Reproducibility
  - Replication of the analysis based on the collected data (not replicating the data collection itself)
  - Better if we have the actual code or available executables
Reproducibility Spectrum

---

**Reproducibility Spectrum**

- **Publication only**
  - Code
- **Publication +**
  - Code and data
- **Linked and executable code and data**
- **Full replication**
- **Gold standard**
- **Not reproducible**

---

[R. D. Peng]
10 Rules for Reproducible Computational Research

• Rule 1: For Every Result, Keep Track of How It Was Produced
• Rule 2: Avoid Manual Data Manipulation Steps
• Rule 3: Archive the Exact Versions of All External Programs Used
• Rule 4: Version Control All Custom Scripts
• Rule 5: Record All Intermediate Results, When Possible in Standardized Formats

[Sandve et al., 2013]
10 Rules for Reproducible Computational Research

• Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
• Rule 7: Always Store Raw Data behind Plots
• Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
• Rule 9: Connect Textual Statements to Underlying Results
• Rule 10: Provide Public Access to Scripts, Runs, and Results

[Sandve et al., 2013]
(Database) Reproducibility Research Topics

- Design and Management of Experiment Repositories
- Querying and Searching Experiments
- Mining Experiments
Notebook Reproducibility

- Use notebooks from Github (~1 million)
  - Unambiguous cell order? 81.99%
- Study notebook dependencies
  - Dependencies Available? 13.72%
  - Dependencies Install? 5.03%
- Study notebook executability
  - Execute: 24.11% of unambiguous cell order
  - Matched results: 4.03%
# Dataflow Notebooks: Resolve Notebook Ambiguities

In [d51f8eab]:
```python
import pandas as pd
df = pd.read_csv('guardian-top100-female-2019.csv')
df:
<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age on 1 Dec 2019</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam Kerr</td>
<td>1</td>
<td>Forward</td>
<td>26</td>
<td>Australia</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ludmila</td>
<td>100</td>
<td>Forward</td>
<td>25</td>
<td>Brazil</td>
</tr>
</tbody>
</table>
```

100 rows x 5 columns

In [full]:
```python
df = df.rename(columns={'Age on 1 Dec 2019': 'Age'})
df:
<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam Kerr</td>
<td>1</td>
<td>Forward</td>
<td>26</td>
<td>Australia</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ludmila</td>
<td>100</td>
<td>Forward</td>
<td>25</td>
<td>Brazil</td>
</tr>
</tbody>
</table>
```

100 rows x 5 columns

In [over30]:
```python
df = df[df.Age >= 31]
df:
<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Megan Rapinoe</td>
<td>3</td>
<td>Midfield</td>
<td>34</td>
<td>USA</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Cláudia Neto</td>
<td>97</td>
<td>Midfield</td>
<td>31</td>
<td>Portugal</td>
</tr>
</tbody>
</table>
```

19 rows x 5 columns

In [under25]:
```python
df = df[df.Age <= 24]
df:
<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada Hegerberg</td>
<td>4</td>
<td>Forward</td>
<td>24</td>
<td>Norway</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Lena Oberdorf</td>
<td>99</td>
<td>Midfield</td>
<td>17</td>
<td>Germany</td>
</tr>
</tbody>
</table>
```

25 rows x 5 columns
Assignment 5

• Four parts
  - Loading Data
  - Spatial Analysis
  - Graph Analysis
  - Temporal Analysis
• Due at the end of the semester (April 22, 2021)
• Questions?
Final Exam

- Monday, April 26, 4:00-5:50pm, Online (Blackboard)
- Similar format
- More comprehensive (questions from topics covered in Test 1 & 2)
- Will also have questions from temporal data, provenance, reproducibility, machine learning
- Bring questions on Wednesday
Improving Databases
LEARNED AND SELF-DESIGNING DATA STRUCTURES

Stratos Idreos & Tim Kraska
Algorithms rely on the order of data
Data systems rely on algorithms
Data structures define performance

As time goes by, data structures become ever more critical for data-driven applications. Data structures define performance.

Jim Gray, Turing Award 1998

register = this room

memory = nearby city

caches = this city

disk = Pluto

[S. Idreos, 2019]
Database Questions

How do I make my data system run x times as fast?  
(sql,nosql,bigdata, ...)  

How do I minimize my bill in the cloud?  

How do I extend the lifetime of my hardware?  

How to accelerate statistics computation for data science/ML?  

How do I train my neural network x times faster?  

[S. Idreos, 2019]
Every data structure design is simply a point in the design space of possible solutions. There is no perfect design. Every design balances the fundamental tradeoffs of Read, Update, and Memory amplification. For example, Read amplification is defined as the excess data an algorithm needs to read on top of the data it wants to read. Typically a data structure would have some kind of metadata or navigation data that help locate the actual data, e.g., the internal nodes of a B-tree. Reading this navigation data is an excess cost, adding to read amplification. Creating a data structure without any navigation data would suffer update or even more read amplification. For example, we could choose to not have any structure in the data at all. Then every query would have to touch all the data. The other extreme would be to sort all data which effectively provides an implicit structure. But then updates get expensive. Overall, there is no perfect design.

Tradeoffs in each structure

[S. Idreos, 2019]
New Applications Demand Change

NEW APPLICATIONS

existing systems need to change too

WORKLOAD HARDWARE

ADAPT

IMPROVE WITHIN A BUDGET

WHAT WILL BREAK MY SYSTEM?

[Idreos, 2019]
Many efforts in the field have been motivated by the vision of generating tailored systems for a specific scenario. In fact, even traditional databases are architected with this vision in mind. A generic database system can optimize a plan on the fly to match the query needs, it can choose from different storage and indexing options, etc. This is how generic database systems can be used in a wealth of applications! And then recent research has tried to push the boundaries of tailored designs by rethinking parts of the stack of a database system.

"Traditional" Database Research

[S. Idreos, 2019]
Self-designing systems

As a first step in this direction, we built an engine, which we call the Data Calculator and which takes as input the hardware, workload and layout of a data structure. It then computes automatically the algorithms that this data structure design needs to optimally process the workload on this hardware and it also computes the performance. That is, the response time that an actual implementation of this design would need to run this workload on this hardware. However, all this happens without the user having to implement anything and without even needing access to the actual hardware. Given this engine we show that we can start thinking about game-changing paradigms for system designs such as interactive design, self-designing systems, and fully automatic design for instance optimal systems.
SageDB: a learned database system

Learned Data Structures and Algorithms
Discussion

• Is this the future?
• What about comparison baselines?
• Lots of work being done in this area
Reminders

• Assignment 5 Due Thursday
• Final Exam Review Wednesday (come with questions!)
• Final Exam on Monday, April 26 from 4-5:50pm (Online)