Capturing the Laws of (Data) Nature

Hannes Mühleisen, Martin Kersten & Stefan Manegold

CIDR 2015
Statistical Model Fitting & DB?
I am storing some data.

User gave me a model, let’s see.

I need some of the observations to fit the model.

This other guy is reading some of my data.

Cool, the model seems to fit the data well!

Let’s get some more data to validate the fit…

This other guy is reading some more of my data.

Amazing, model fit is validated.

Beer!

I am storing some data.
The point?

- Everyone has models, **they encode our understanding of the world**
- Everyone has data to train/fit and validate a model
- So far, data management community has **ignored** these models
  - But they hold precious domain knowledge!
LOFAR Example
<table>
<thead>
<tr>
<th>source</th>
<th>ngz</th>
<th>flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.15595</td>
<td>0.23159</td>
</tr>
<tr>
<td>1</td>
<td>0.12392</td>
<td>0.34782</td>
</tr>
<tr>
<td>1</td>
<td>0.14892</td>
<td>0.15927</td>
</tr>
<tr>
<td>1</td>
<td>0.14892</td>
<td>0.32261</td>
</tr>
<tr>
<td>1</td>
<td>0.15595</td>
<td>0.35390</td>
</tr>
<tr>
<td>1</td>
<td>0.14892</td>
<td>0.32784</td>
</tr>
<tr>
<td>1</td>
<td>0.12392</td>
<td>0.33068</td>
</tr>
<tr>
<td>1</td>
<td>0.18486</td>
<td>0.28135</td>
</tr>
<tr>
<td>1</td>
<td>0.18486</td>
<td>0.17872</td>
</tr>
<tr>
<td>1</td>
<td>0.15595</td>
<td>0.28813</td>
</tr>
</tbody>
</table>
alpha <- sapply(split(rsmddata, rsmddata$source),
  function(df) {
    tryCatch(
      fit <- nls(flux ~ p * ngz ^ alpha,
        data=df, start=list(alpha=-.7, p=0.8),
        control = list(maxiter = 500))
      coefs <- coef(fit)
      coefs[[1]]
      }, error=function(e) {return(as.numeric(NA))
    })
  }
flux \sim p \times ngz^{\alpha}

Fitted parameters
<table>
<thead>
<tr>
<th>Source</th>
<th>Wavelength $\nu$</th>
<th>Intensity $I$</th>
<th>Source</th>
<th>Spectral Index $\alpha$</th>
<th>Constant $p$</th>
<th>Residual SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1559555</td>
<td>0.2315911</td>
<td>1</td>
<td>-0.7183309</td>
<td>0.06257838</td>
<td>0.006559710</td>
</tr>
<tr>
<td>1</td>
<td>0.1239243</td>
<td>0.3478159</td>
<td>2</td>
<td>-0.8932245</td>
<td>0.07195620</td>
<td>0.008007786</td>
</tr>
<tr>
<td>1</td>
<td>0.1489243</td>
<td>0.1592717</td>
<td>3</td>
<td>-0.7880774</td>
<td>0.56190180</td>
<td>0.016778232</td>
</tr>
</tbody>
</table>

[1,452,821 more rows]

[35,681 more rows]
source=17562, alpha=-0.692, p=0.812
Exploit!
\[ \text{flux} \sim p \times \text{ngz} \wedge \alpha \]

Model to function conversion (automatic)

\[ \text{flux} \leftarrow \text{function}(p, \text{ngz}, \alpha) \{ p \times \text{ngz} \wedge \alpha \} \]

Move to DB (automatic)

```
CREATE FUNCTION flux3()
RETURNS TABLE (source integer, nu double, flux DOUBLE) LANGUAGE R {
    r <- data.frame(...) # model params
    r$flux <- r$p * r$nu ^ r$alpha
    return(r[c("source","nu","flux")])
};
```
sql> select flux from flux3() where source=17562 and nu=0.156;
+-------------------------------+
| flux                          |
|-------------------------------+
| 2.9370666853479666           |
+-------------------------------+

Approximate Answer with zero IO*
But...

- What do we do if model parameters are not specified in the query?
  - Sample data?
- Given multiple parameters, it is far from certain that all combinations of values are allowed in the model.
  - Construct filter?
“Semantic” Compression

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flux Residuals</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIG</td>
<td>11,665,408</td>
<td>0%</td>
</tr>
<tr>
<td>GZIP</td>
<td>4,331,782</td>
<td>86%</td>
</tr>
<tr>
<td>BZIP2</td>
<td>3,341,574</td>
<td>82%</td>
</tr>
<tr>
<td>XZ</td>
<td>2,887,584</td>
<td>94%</td>
</tr>
</tbody>
</table>

Drop residuals = lossy compression =
Data & Model Changes

• What should we do if the user gives us a better model?
  • Recompressing could be very expensive
  • Threshold for improvement?

• Changes in the data affect the model quality, too
  • Switch models?
  • Constant Monitoring?
Multiple, partial or grouped

• There could be many models for a table with overlapping parameters
  • Which one to pick?
• Models do not have to cover the entire table/column
  • “Patching”? 
• Models could be fitted on aggregation results
  • Keep group counts?
How do we get our hands on Models?
Oracle R Enterprise
Performance, Scalability, and Ease of Deployment Deployment

R on Spark
SparkR is an R package that provides a lightweight front-end to use Apache Spark from R.

Features
RDDs as Distributed Lists

SAP HANA R Integration Guide

Distributed R
A scalable and high-performance platform for the R language.
Integrate & Intercept

- Integrate model fitting infrastructure into data management system.
- Also: Huge performance benefits for analysts!
- Intercept model fitting and validation operations by the user and store the model for later use.
- Storage format: Model code + Parameters
\[ I \approx p \cdot \nu^\alpha ? \]

\[ R^2 = 0.92 ! \]

\[ I \approx p \cdot \nu^\alpha ? \]

\[ R^2 = 0.92 ! \]

\[ S = 42, \nu = 0.14, I =? \]

\[ I = 3.0 \pm 0.05 ! \]
“Essentially, all models are wrong, but some are useful.”

George E. P. Box

Questions?

http://hannes.muehleisen.org
@hfmuehleisen