A Survey on Domain-Specific Languages for Machine Learning in Big Data

Ivens Portugal
Agenda

- Introduction
  - Big Data and Machine Learning
- Research Problem and Goal
- Approach
- Research Progress
- Results and Contributions
Introduction: Big Data

• Definition (Tanaka, 2013)
  • Relates to datasets whose size is beyond the ability of typical database software to capture, store, manage, and analyze.

• 3V (Beyer & Laney, 2012)
  • Volume, Velocity, Variety

• Example (Chen & Zhang, 2014)
  • 267 million transactions in Walmart per day
  • 3 billion pieces of content generated on Facebook per day
  • 30 petabytes of image data generated by Large Synoptic Survey Telescope (LSST) per day
  • 60 terabytes of data generated by Large Hadron Collider (LHC) per day
Introduction: Machine Learning

• Definition (Simon, 2013)
  
  • A field of study that give computers the ability to learn without being explicitly programmed.

• Created in 1950s, popular since 1990s.

• Algorithms (Rajaraman & Ullman, 2012)
  
  • Bayesian Network, k-Means, Clustering, Logistic regression, Support vector machine, Neural network, and many more.
Research Problem and Goal

- Machine learning in Big Data
  - More data, more learning, more research, more results
  - How to make it easy to use and access?
- Goal:
  - Let’s make it easy to develop.
  - Let’s survey and analyze the languages being used
    - GPL - C, C++, Java, UML
    - DSL - SQL, Matlab, HTML
Approach

- DSL - classification (Van Deursen et al., 2000; Fowler, 2010)
  - Requirements, Programming, Modeling
  - Textual, Graphical
  - Internal, External
  - Dynamically typed, Statically typed
  - Imperative, Declarative
  - Translation (Compilation), Interpretation
  - (External) Target Platform and Execution Engine
  - (Modeling) Descriptive, Prescriptive model
Approach

- DSL
- OptiML (Sujeeth et al., 2011)
- ScalOps (Weimer et al., 2011)
- Pig Latin (Olston et al., 2008)
- SCOPE (Chaiken et al., 2008)
- Sawzall (Pike et al., 2005)
- BreukerVL (Breuker, 2014)
- Graphical Models (Heckerman, 1998)
## Research Progress

<table>
<thead>
<tr>
<th>DSL</th>
<th>Requirements/Programming/Modeling</th>
<th>Textual/Graphical</th>
<th>Internal/External</th>
<th>Dynamically/Staticaly typed</th>
<th>Imperative/Declarative</th>
<th>Translation/Interpretation</th>
<th>Target Platform/Execution Engine</th>
<th>Descriptive/Prescriptive model</th>
<th>Supports Vector (V)/Matrix (M)/Graph (G) operations</th>
<th>Supports Parallel operations</th>
<th>Supports Distributed (D)/Cloud computing (C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OptiML</td>
<td>Programming Textual Internal (Scala) Statically typed Declarative Translation - - V/M/G Yes -/-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ScalOps</td>
<td>Programming Textual Internal (Scala) Statically typed Declarative Translation - - V/M/G Yes D/C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pig Latin</td>
<td>Programming Textual External Dynamically typed Imperative Translation Pig Latin compiler / Apache Pig - V/M/- Yes D/C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BreukerVL</td>
<td>Modelling Graphical External - - - - Descriptive - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical Models</td>
<td>Modeling Graphical External - - - - Descriptive - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>DSL</td>
<td>Requirements/Programming/Modeling</td>
<td>Textual/Graphical</td>
<td>Internal/External</td>
<td>Dynamically/Statically typed</td>
<td>Imperative/Declarative</td>
<td>Translation/Interpretation</td>
<td>Target Platform/Execution Engine</td>
<td>Descriptive/Prescriptive model</td>
<td>Supports Vector (V)/Matrix (M)/Graph (G) operations</td>
<td>Supports Parallel operations</td>
<td>Supports Distributed (D)/Cloud computing (C)</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------</td>
<td>-------------------</td>
<td>------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>---------------------------------</td>
<td>--------------------------------</td>
<td>-----------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Sawzall</td>
<td>Programming</td>
<td>Textual</td>
<td>External</td>
<td>Statically Typed</td>
<td>Imperative</td>
<td>Interpretation</td>
<td>Sawzall compiler / Sawzall engine (proprietary)</td>
<td>-</td>
<td>V/M/-</td>
<td>Yes</td>
<td>D/C</td>
</tr>
<tr>
<td>BreukerVL</td>
<td>Modelling</td>
<td>Graphical</td>
<td>External</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Descriptive</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Graphical Models</td>
<td>Modeling</td>
<td>Graphical</td>
<td>External</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Descriptive</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
## Frameworks

<table>
<thead>
<tr>
<th>Framework name</th>
<th>Textual / Graphical</th>
<th>Languages</th>
<th>Supports Vector (V) / Matrix (M) / Graph (G) operations</th>
<th>Supports Parallel operations</th>
<th>Supports Distributed (D) / Cloud (C) computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infer.net (Minka et al., 2014)</td>
<td>Textual</td>
<td>.NET framework languages</td>
<td>V/M/-</td>
<td>Yes</td>
<td>-/-</td>
</tr>
<tr>
<td>Graphlab (Low et al., 2010)</td>
<td>Textual</td>
<td>C++, Python</td>
<td>V/M/G</td>
<td>Yes</td>
<td>D/C</td>
</tr>
<tr>
<td>TensorFlow (Abadi et al., 2015)</td>
<td>Textual</td>
<td>C++, Python</td>
<td>V/M/G</td>
<td>Yes</td>
<td>D/C</td>
</tr>
</tbody>
</table>
Results

- Results
  - Most of the languages identified so far in the survey are programming DSLs
  - No DSLs to describe system requirements
  - Two programming languages are based on Java
  - Most languages are compiled
  - Most languages support parallel or distributed execution, as well computations in the cloud
  - Modeling languages are all graphical and have a descriptive model.
Contributions

- Contributions
  - Better understanding of the development of systems in the domain of Machine Learning in Big Data
  - Beginners may better choose a language to start developing, modeling or gathering requirements for this domain
References


References

A Preliminary Survey on Domain-Specific Languages for Machine Learning in Big Data

Ivens Portugal