What’s Cooking in KNIME

Thomas Gabriel
Agenda

• Querying NoSQL Databases
• Database Improvements & Big Data
Querying NoSQL Databases
MongoDB & CouchDB

- Alexander Fillbrunn -
Database Improvements and Big Data

- Tobias Koetter -
Database Improvements
Database – New Nodes

• Database Create Table
  – Define constraints e.g. primary and unique keys
  – Indexing options
• Database Column Rename
• Database Pivoting
• More connectors
  – SAP Hana
  – ...

Database – Framework Improvements

• Connection handling
  – Connection pool (more speed)
  – Dedicated connections (more control)

• Improved sql editors
  – Syntax highlighting
  – Code completion
Big Data
Machine Learning on Hadoop

• Based on Spark MLlib
• Scalable machine learning library
• Runs on Hadoop
• Algorithms for
  – Classification (decision tree, naïve bayes, ...)
  – Regression (logistic regression, linear regression, ...)
  – Clustering (k-means)
  – Collaborative filtering (ALS)
  – Dimensionality reduction (SVD, PCA)
MLlib Integration

- Learn node for each algorithm
- Hive tables as input format
- MLlib model ports for model transfer
MLlib Integration

- MLlib nodes start and manage Spark jobs
MLlib Integration

Spark Jobs

Total Duration: 56 s
Scheduling Mode: FIFO
Active Jobs: 1
Completed Jobs: 0
Failed Jobs: 0

Active Jobs (1)

<table>
<thead>
<tr>
<th>Job Id</th>
<th>Description</th>
<th>Submitted</th>
<th>Duration</th>
<th>Stages: Succeeded/Total</th>
<th>Tasks (for all stages): Succeeded/Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>takeSample at KMeans.scala:277</td>
<td>2015/02/19 14:23:48</td>
<td>13 s</td>
<td>0/1</td>
<td>5/6</td>
</tr>
</tbody>
</table>

Completed Jobs (0)
MLlib Integration

- One predictor for all MLlib models
- Usage model and dialogs similar to existing KNIME mining nodes
MLlib Integration

Spark Jobs

Total Duration: 1.6 min
Scheduling Mode: FIFO
Active Jobs: 0
Completed Jobs: 21
Failed Jobs: 0

Active Jobs (0)

Completed Jobs (21)

<table>
<thead>
<tr>
<th>Job Id</th>
<th>Description</th>
<th>Submitted</th>
<th>Duration</th>
<th>Stages: Succeeded/Total</th>
<th>Tasks (for all stages): Succeeded/Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>runJob at InsertIntoHiveTable.scala:93</td>
<td>2015/02/19 14:24:20</td>
<td>8 s</td>
<td>1/1</td>
<td>6/6</td>
</tr>
<tr>
<td>19</td>
<td>collectAsMap at KMeans.scala:207</td>
<td>2015/02/19 14:24:16</td>
<td>0.6 s</td>
<td>2/2</td>
<td>12/12</td>
</tr>
<tr>
<td>18</td>
<td>collectAsMap at</td>
<td>2015/02/19 0.3 s</td>
<td>2/2</td>
<td></td>
<td>12/12</td>
</tr>
</tbody>
</table>
MLlib Integration

- Hive tables as input/output format
- Data stays within your HDFS file system
- No unnecessary data movements
MLlib to KNIME

- Converts supported MLlib models to PMML
- Learning at scale on Hadoop
- Prediction with speed based on compiled models
  - Can be combined with the new REST API
KNIME to MLlib

- Prediction at scale on Hadoop
- Compatible with KNIME models and pre-processing steps
Mix and Match

• Use all the KNIME nodes on your big data samples
Closing the Loop

Apply model → Learn model → PMML model

Learn model → Apply model → MLlib model

Learn model

MLlib to PMML

MLlib PMML Predictor
Agenda

• Querying NoSQL Databases
• Database Improvements & Big Data

• New KNIME Server
• Wizard Execution
• Workflow Diff
New KNIME Server
WebPortal Templates & REST Interface

- Thorsten Meinl -
Why TomEE?

- Apache TomEE is based on Apache Tomcat
  - Much higher adoption than Glassfish
  - Additional libraries to support EJB
- Communication solely via HTTP
  - No more firewall problems
  - Encryption via HTTPS
- Installation and deployment considerably easier
- Better user and group management
- Simultaneous connection to multiple servers
- KNIME Server 4.0 available after the UGM
WebPortal Templates (I)
WebPortal Templates (II)

Login to mine data gold!
Username
Password
Login
WebPortal Templates (III)
WebPortal Templates (IV)

**Test jQuery existence**

*jQuery is available, node correctly loaded and configured.*

**Test jQuery non-existence**

*jQuery is not available, node correctly loaded and configured.*

**Test data not available**

*No data available. Node correctly configured.*

**Test jQuery and D3 existence**

*jQuery and D3 available, node correctly loaded and configured.*

**Test data available**

*Data available. Node correctly configured. Table contains 2500 rows.*

**Fail on purpose**

*This node is supposed to fail.*
WebPortal Templates (V)

• Layout can be configured by templates
  – Footer & header
  – Main panel
  – Login page
  – ...

• Custom stylesheet

• Custom JavaScript libraries
  – Can be re-used in JS-based views
WebPortal Templates (VI)

- Templates are part of the configuration and are not overridden by updates
REST Interface

• Main addition to KNIME Server 4.1
• REST = Representational State Transfer
  – Communication based on HTTP
  – Usually clear text
• Many possible clients
  – Web browser
  – Java applications (e.g. via JAX-RS)
  – KREST nodes :-)
• Goal: complete server interface based on REST
REST Example: List Workflows (I)

• Via browser
  – Requires user authentication
REST Example: List Workflows (II)

• Via KNIME and KREST nodes
REST Example: Execute Workflow (I)

- **Via browser**
  - **Load workflow**
    - Returns unique job ID
  - **Execute job**
    - [http://localhost:8080/com.knime.enterprise.server/rest/v4/jobs/syncExec/24a76fec-a74e-45ba-b03f-cabf528b6a69](http://localhost:8080/com.knime.enterprise.server/rest/v4/jobs/syncExec/24a76fec-a74e-45ba-b03f-cabf528b6a69)
    - Returns final status
  - **Render report**
    - Format can be specified in request
REST Example: Execute Workflow (II)

- Via KNIME and KREST nodes
REST Example: Live-Scoring on server (I)

- Get expected parameter format from workflow
- Set input parameters in input quickform nodes
- Execute workflow
- Get results from quickform output nodes

Scoring workflow, called via REST
REST Example: Live-Scoring on server (II)

- Get expected parameter format from workflow
- Set input parameters in input quickform nodes
- Execute workflow
- Get results from quickform output nodes
REST Example: Live Scoring on server (III)

- Via Call Remote Workflow node
  - Analyzes input parameters
  - Prepare input data accordingly
  - Executes job and gets back results

File Reader → Table to JSON → Row Sampling → Call Remote Workflow
New KNIME Server

- Shared Metanode Repository
- Scheduled Execution
- Remote Execution
- REST Interface
- Custom Application Access
- Wizard Execution
- Shared Data File Repository
- Workflow Diff
- KNIME Client Access
- WebPortal Templates
- User Access Rights

Copyright © 2015 KNIME.com AG
Workflow Diff – Simple Example I

A simple example using a decision tree learner and predictor to classify some shuttle data. For more background information see http://archive.ics.uci.edu/ml/datasets/Shuttle.
Workflow Diff – Simple Example II
Workflow Diff – Extended Example
Workflow Diff – Filtering
Wizard Execution I

- New Set of JavaScript-based interactive Views and QuickForm Nodes
Wizard Execution II
Wizard Execution III

Input Variables

Output Variables
Wizard Execution IV

![Diagram of various visualization options](image)

- **JavaScript Line Plot**
- **JavaScript Table View**
- **JavaScript Scatter Plot**
- **Generic JavaScript View**
Wizard Execution IV

![Diagram of JavaScript views]

1. JavaScript Line Plot
2. JavaScript Table View
3. JavaScript Scatter Plot
4. Generic JavaScript View
What’s Cooking?

12:30-13:30 It’s lunchtime