Decision models for the Digital Economy

Vijay Bandekar
InteliOps Inc.
Agenda

• Problem Statement
• Proposed Solution
• Case studies and results
• Key takeaways
• Discussions
Digital economy demands autonomous, self-learning, real-time decision making systems that **sense, comprehend** and **act**.

Decisions must adapt to changing business environments.
**Decision Models:**
Inference, Reasoning and Deductions

- What are decision models?  
  - Business Logic Templates.

- How do they work?  
  - Workflows, rules engine, ...

- Available technologies?  
  - Rete, logic programs, SQL

- Rules development and maintenance is cumbersome.

- Machine learning can generate rules/models, however, they must be manually incorporated into a business rules engines (BRE).
Reasoning and Inference

- Data driven productions - Rete
  Underutilized.

- Implementations
  Performance varies.

- Architectures
  Complex and
  Inefficient.

- Environments
  Cumbersome
  tools.
Architectures: Stateless v/s State-full

Stateless Working Memory

1. Create knowledge session – Rete
2. Re-create state
3. Insert new facts
4. Call ‘fireall’
5. Persist state to DB
6. Dispose knowledge session

Response time =
  time to create knowledge session +
  time to re-create state +
  time to insert all facts +
  time to fire all rules +
  time to persist state to the DB +
  time to dispose session

State-full Working Memory

1. Create knowledge session – Rete
2. Insert new facts
3. Call ‘fireall’
4. Go to 2

Response time =
  time to create knowledge session +
  time to insert all facts +
  time to fire all rules
Working Memory:
Agenda Control & Transparency

WM Statistics:
Collect Mean Square from source and Statistical Variable: 15092
Collect data from source and Statistical Variable: 15092
Populate Domains: 3773
Compute conversion rate: 2778
Compute cost to order ratio: 2450
Compute cost to revenue ratio: 2449

Fact Count(before): 0, Fact Count(after): 14463
ObjectInserted: 14463, ObjectUpdated: 0, ObjectRetracted: 0
“Store-first-process-later” does not work.

The traditional big-data analytics is too slow, and requires custom integration with decision engines.

Deep-learning is not the answer to all ML problems.
Solution:
Adaptive Decisions Framework

Reasoning and Inference

Data driven productions in a state-full knowledge session
- Rule engines
- Intelligent agents
- Deductions and inferences
- Data patterns and events

Continuous Learning

Incremental model building
- Statistical classification
- Decision tree
- Clusters
- Regression
- Neural networks
  - Resilient back propagation
  - Recurrent
  - Convolutional

Sense
Comprehend
Act
Incremental Bayesian Learner

- Rules manage learning categories
- Incrementally add sample records
- Real-time update of probabilities
- Learner operates in dual mode
  - Continuous learning
  - Inference/Prediction
Learned decision tree transformed into Rete
Models injected into Rete
Case Study 1: Media Purchase Decisions

### CHANNEL PERFORMANCE

<table>
<thead>
<tr>
<th></th>
<th>&lt; &lt;&lt; 2016 Week 15 &gt;&gt; &gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top 5</strong></td>
<td><strong>Bottom 5</strong></td>
</tr>
<tr>
<td>Rank</td>
<td>Channel</td>
</tr>
<tr>
<td>1</td>
<td>EMAIL</td>
</tr>
<tr>
<td>2</td>
<td>INTERNAL</td>
</tr>
<tr>
<td>3</td>
<td>NATURAL SEARCH</td>
</tr>
<tr>
<td>4</td>
<td>PAID SEARCH</td>
</tr>
<tr>
<td>5</td>
<td>REFERRING DOMAINS</td>
</tr>
</tbody>
</table>

1. Channel attribution (A)
2. Channel combination probabilities (B)
3. Channel-Brand combination (C)

A. probability of a channel producing revenue = revenue attributed / total revenue for the period
Statistics on KPIs and categories

Frequency distributions

Depending on the distribution, mapping to categorical values will utilize mean and median. *Adaptive Decision Framework* provides Statistics template to automate this normalization process.
Results: Single retailer daily process

<table>
<thead>
<tr>
<th>Daily Data Volume per Retailer</th>
<th>Omniture tracking</th>
<th>Sales</th>
<th>Rules count</th>
<th>Rules fired</th>
<th>Facts count</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Initial</td>
<td>Final</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30,184</td>
<td>80,000</td>
<td>20</td>
<td>373,668</td>
<td>115,704</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run</th>
<th>State-full</th>
<th>Stateless</th>
<th>% Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140</td>
<td>360</td>
<td>61%</td>
</tr>
<tr>
<td>2</td>
<td>166</td>
<td>360</td>
<td>54%</td>
</tr>
<tr>
<td>3</td>
<td>66</td>
<td>360</td>
<td>82%</td>
</tr>
<tr>
<td>4</td>
<td>113</td>
<td>360</td>
<td>69%</td>
</tr>
<tr>
<td>5</td>
<td>138</td>
<td>360</td>
<td>62%</td>
</tr>
<tr>
<td>6</td>
<td>79</td>
<td>360</td>
<td>78%</td>
</tr>
<tr>
<td>7</td>
<td>68</td>
<td>360</td>
<td>81%</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>360</td>
<td>82%</td>
</tr>
<tr>
<td>9</td>
<td>71</td>
<td>360</td>
<td>80%</td>
</tr>
<tr>
<td>10</td>
<td>62</td>
<td>360</td>
<td>83%</td>
</tr>
<tr>
<td></td>
<td>96.7</td>
<td>360</td>
<td>73%</td>
</tr>
</tbody>
</table>
Case Study 2: MBS Performance Prediction

• Predict individual loan performance
  – 25 M residential mortgages acquired over 10 year
  – 1.1 B monthly performance records over 10 year

• Cluster loans and form MBS pools
  – Interest rate distance
  – Origination distance

• Simulate MBS pool performance
  – Random loan pools based on clusters
  – Monte Carlo simulation of MBS yield
Performance

**Benchmark:** 2 Core 24 GB RAM VM with 100 GB hard disk.

- 45 Quarters of loan acquisition and performance data since Q1 2000:
  - Stream loan data form published web site, and
  - Insert facts into state-full working memory **4.5 hours.** *(Stateless: 16.75 hours estimated)*

- Add sample records to learning category: **1.35 minutes/Million records.** *(Stateless: 22.3 hours estimated)* **Training time:**
  - **6 hours.** *(Stateless: 22.3 hours estimated)*
Key takeaway

• State-full working memory enables *real-time decisions*.

• Combining decision engine with ML facilitates *adaptive/autonomous* behavior.
Conclusion

• Architecture using state-full working memory wins
• Continuous learning within decision engine enables adaptive behavior
• Performance of adaptive decision framework meets the needs of Digital Economy.
  ➢ Volume
  ➢ Velocity