Real-time Sensory Pattern Mining for Autonomous Agents

Pedro Sequeira and Cláudia Antunes

ADMI-10
May 11, 2010
Outline

- Motivation
- Problem
- Proposed Solution
- Experimental Results
- Conclusions and Future Work
Autonomous Agents (1/2)

- **Definitions:** [Franklin & Graesser, 1997; Maes, 1994]
  
  - situated in dynamic environments
  
  - have and actively pursue goals
  
  - satisfy their needs
  
  - respond to external events from the environment
Autonomous Agents (2/2)

Requirements: [Franklin & Graesser, 1997; Maes, 1994]

- mechanisms to distinguish perceived features
  - focus on relevant features, ignore non-important ones
- adapt to and learn new knowledge from the environment
- structures that represent the acquired knowledge
- mechanisms that update these representations overtime to reflect interaction experience
Motivation

Pattern Mining

- **Data-Mining**: [Frawley, 1992]
  - techniques to extract previously unknown and possibly useful information from data

- **Pattern-Mining**: [Agrawal, 1994]
  - technique to discover frequent patterns, associations, correlations, or causal structures among sets of items or objects
Motivation

Idea

• Main idea:
  • use pattern mining techniques within an autonomous agent
  • extend and adapt some pattern mining algorithms and structures

• Objective:
  • discover associative patterns in the agent’s perceptions
  • discovery made while interacting with the environment

• Result:
  • patterns can provide useful knowledge about the world
  • form concepts of regularities
Outline

- Motivation
- Problem
- Proposed Solution
- Experimental Results
- Conclusions and Future Work
**Problem**

**Scenario**

- **Sensors**
  - hunger
  - health
  - mouth
  - throat
  - taste
  - tactile-tongue

- **Stimulus**
  - hungry
  - weak
  - mouthing
  - swallowing
  - sweet,
  - soft,

- **Sensory Properties**
  - taste – sweet
  - shape – round
  - color – orange

- **Action**
  - eat

- **Internal Stimuli**
  - hunger
  - health
  - mouth
  - throat
  - taste
  - tactile-tongue
Problem

Pattern-Mining Analogy (1/2)

- Transactional Pattern Mining:

<table>
<thead>
<tr>
<th>TID</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{ beer, red wine, whiskey, tequila, vodka, rum }</td>
</tr>
<tr>
<td>2</td>
<td>{ beer, white wine, red wine }</td>
</tr>
<tr>
<td>3</td>
<td>{ tequila, whiskey, rum }</td>
</tr>
<tr>
<td>4</td>
<td>{ beer, whiskey, tequila, red wine }</td>
</tr>
<tr>
<td>5</td>
<td>{ whiskey, vodka }</td>
</tr>
<tr>
<td>6</td>
<td>{ tequila, beer }</td>
</tr>
<tr>
<td>7</td>
<td>{ vodka, rum, red wine, beer }</td>
</tr>
<tr>
<td>...</td>
<td>{ ... }</td>
</tr>
</tbody>
</table>
Transactional Pattern Mining: [Agrawal, 1994]

- Discover patterns in *Transactions*
- *Patterns* are maximal sets of *Items*
  - occur together very often
- number of transactions of co-occurrence > *Support*

\[
\text{Sup}(A \rightarrow B) = P(A \cap B) = \frac{|A \cap B|}{N}
\]
Problem

Pattern-Mining Analogy (2/2)

Sensors | Stimulus
--- | ---
hunger | hungry
health | weak
mouth | mouthing
throat | swallowing
taste | sweet, …
tactile-tongue | soft, …

Sensory Properties
- taste – sweet
- shape – round
- color – orange

Action: eat

Internal stimuli

Object properties
Pattern-Mining Analogy (2/2)

Problem

Sensory Properties
- taste – sweet
- shape – round
- color – orange

Sensors
- hunger
- health
- mouth
- throat
- taste
- tactile-
- tongue

Stimulus
- hungry
- weak
- mouthing
- swallowing
- sweet, ...
- soft, ...

Sensory Transaction

action eat

internal stimuli

object properties
**Problem**

**Pattern-Mining Analogy** (3/3)

- Sensory Pattern Mining:

<table>
<thead>
<tr>
<th>TID</th>
<th>Agent's Stimuli</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{ orange, hungry, sweet, round, weak, mouthing }</td>
</tr>
<tr>
<td>2</td>
<td>{ orange, white wine, hungry }</td>
</tr>
<tr>
<td>3</td>
<td>{ round, sweet, mouthing }</td>
</tr>
<tr>
<td>4</td>
<td>{ orange, sweet, round, hungry }</td>
</tr>
<tr>
<td>5</td>
<td>{ sweet, weak }</td>
</tr>
<tr>
<td>6</td>
<td>{ round, orange }</td>
</tr>
<tr>
<td>7</td>
<td>{ weak, mouthing, hungry, orange }</td>
</tr>
<tr>
<td>...</td>
<td>{ ... }</td>
</tr>
</tbody>
</table>
Problem Definition

- Find *synchronous sensory patterns*:  
  - sets of stimuli that occur frequently and simultaneously  
  - reveal some regularities of its environment

- Requirements:  
  - real-time discovery while interacting w/ environment  
  - maintain in memory the maximum number of patterns  
  - gather useful information to be used by the agent  
  - efficient algorithm
Outline

- Motivation
- Problem
- Proposed Solution
- Experimental Results
- Conclusions and Future Work
Proposed Solution

Transactional Pattern Mining (1/2)

- **Apriori**: [Agrawal, 1994]
  - iteratively generates the set of candidate *itemsets* of length $k$ from the set of frequent-patterns of length $k-1$

- **FP-Growth**: [Han, 2004]
  - builds up a compact structure (*FP-Tree*) from the data
  - requires only 2 steps:
    - frequency count
    - build *FP-Tree*
  - frequent patterns search only from the tree
Proposed Solution

Transactional Pattern Mining (2/2)

- Drawbacks:
  - requires a first scan to order items by frequency
    - impossible to realize in autonomous agent’s
  - algorithm determines all the frequent itemsets
    - sometimes difficult to determine if a specific itemset is a pattern
  - FP-Tree is a very compact structure
    - difficult to discover patterns if in real-time
Problem

Problem Definition

- Find *synchronous sensory patterns*: 
  - sets of stimuli that occur frequently and simultaneously
  - reveal some regularities of its environment

- Requirements:
  - real-time discovery while interacting with the environment
  - maintain in memory the maximum number of patterns
  - gather useful information to be used by the agent
  - efficient algorithm
Proposed Solution

Real-Time Pattern Mining

- Modify *FP-Growth* for real-time:
  - discard first scan over the DB to perform sorting
  - use alphabetic order of the symbols (stimulus)

- Change transaction insertion into *FP-Tree*
  - each node represents a specific *itemset*
  - node count represents *itemset* frequency

- Pruning heuristic
  - delete nodes that do not seem promising
  - prune if *support* is below predefined threshold
Example

- Insert transaction \{a, b, c\}:

```
  a
 / \
 b  c
 /   \
 c    c
```

“abc”
Dependency Tree
Problem Definition

- Find *synchronous sensory patterns*:
  - sets of stimuli that occur frequently and simultaneously
  - reveal some regularities of its environment

- Requirements:
  - real-time discovery while interacting w/ environment
  - maintain in memory the maximum number of patterns
  - gather useful information to be used by the agent
  - efficient algorithm
Pattern definition problem:

- based on the statistic of the frequency of an *itemset*
- in relation to the total number of transactions recorded
- some items are correlated, but their frequency is low
- useful information is sometimes discarded
Jaccard Index: \([\text{Jaccard}, 1912]\)

- determines the correlation level between nominal variables (like the agent’s sensations)
- function of the frequency of their intersection over their union:

\[
\text{Jacc}(AB) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}
\]
Proposed Solution

Jaccard Index (3/3)

- Advantages:
  - useful heuristic to determine sensory patterns in agents’ perceptions
  - ignores the total number of perceptions made so far
  - all necessary values already in the tree:
    - divide the count value of the node
    - by the weighted sum of nodes within its Dependency Tree
Proposed Solution

Example

- Jaccard index of itemset “abc”:

\[
\text{Jacc (}abc\text{)} = \frac{|a \cap b \cap c|}{|a| + |b| + |c| - |a \cap b| - |a \cap c| - |b \cap c| + |a \cap b \cap c|}
\]
Outline

- Motivation
- Problem
- Proposed Solution
- Experimental Results
- Conclusions and Future Work
Experimental Results

Test Conditions (1/2)

- **Test environment:** [Sequeira, 2007]
  - agent has random behavior, at each decision step:
    - chooses an object from the environment to interact with
  - 5 possible actions
  - total of 8 objects to interact with
  - total of 13 sensors to perceive the environment
  - 70 possible stimuli to describe the perceptions
Experimental Results

Test Conditions (2/2)

- record file with a sensory transaction per line
- total of 10 test files with 10K transactions (length ≈ 5)
- for each test file
  - iterate minimum support threshold
  - 0 to 1, 0.1 steps
Experimental Results

Test Metrics

- time to build the tree measured in CPU time (Build-Time)
- number of nodes used to build the tree (Node-Number)
- time used to retrieve from the tree every possible pattern, measured in CPU time (Pattern-Time)
- total number of patterns found (Pattern-Number)
Experimental Results

Main Results

- **Jaccard-Index** algorithm:
  - if minimum index threshold is too low:
    - too many: nodes, patterns, time scanning the tree for patterns
    - compromise between performance and usefulness
  - more time to build tree and search for patterns than **FP-Growth**-based algorithms due to index calculation
  - however, mining performance is enhanced
    - tree maintains in memory all the nodes necessary for the index calculation (its *Dependency Tree*)
Outline

- Motivation
- Problem
- Proposed Solution
- Experimental Results
- Conclusions and Future Work
Conclusions and Future Work

Problem Definition

• Find *synchronous sensory patterns*:  
  • sets of stimuli that occur frequently and simultaneously  
  • reveal some regularities of its environment

• Requirements:  
  • real-time discovery while interacting w/ environment  
  • maintain in memory the maximum number of patterns  
  • gather useful information to be used by the agent  
  • efficient algorithm
Algorithm can be a solution to our problem:

- generates patterns of sensory data from the agent's perceptions
- based on the statistical correlation between stimuli
- can perform the task in real-time
- while the agent interacts with the environment
- reflects the agent’s experience
Future Work

- Extend the current algorithm:
  - discover *asynchronous sensory patterns*:
    - sets of stimuli that frequently occur one after the other, revealing causal relations between them
    - adapt algorithms from the sequential pattern mining area
  - test the solution within a multi-agent scenario
    - discover “social knowledge” to reflect the experiences of particular groups
THANK YOU ALL FOR LISTENING!