Toolkit to Support Intelligibility in Context-Aware Applications

Brian Y. Lim, Anind K. Dey

To help users better understand and trust context-aware applications, these applications should be intelligible; they should provide explanations about what they know, and their behavior. We have developed an Intelligibility Toolkit to support the implementation of 8 types of explanations for the 4 most popular decision models used in context-aware applications. For this demonstration, we present 4 applications showing how the Intelligibility Toolkit can be used to generate and provide explanations across decision models and application domains.

### Toolkit Requirements

- **R1** Lower barrier to providing explanations
- **R2** Flexibility of using explanations
- **R3** Facilitate appropriate explanations automatically
- **R4** Support combining explanations
- **R5** Extensible across
  - Explanation types
  - Application (decision) models
  - Provision styles

### Auto-Light Living Room

**Rules**

This takes two factors (Presence and Brightness) to determine whether to turn the light on in a living room. The light would be off if Presence = 0 (i.e., no one in the room) or the detected brightness measure is less than 100 (out of 255).

- QueryPanel to receive user interactive queries
- RulesPresenter to generate explanations from rules
- StringPresenter to generate text output for the UI

### IM Response Prediction

**Decision Tree**

This predicts when a buddy will respond to a message. It is trained on an existing dataset from [Avarhali et al. 2006] to build a decision tree. It takes desktop-based sensor inputs and makes response predictions (within/after 1 min).

- QueryPanel to receive user interactive queries
- DTTreeExplorer to generate user input text as queries
- Reducer to simplify explanations
- Conjunction - to distinguish among output values
- MotionPanelPresenter to represent the phone UI

### Mobile Phone Accelerometry

**Naïve Bayes**

This is a physical activity recognizer that uses the accelerometer on a Google Android mobile phone to infer whether the user is sitting, standing, or walking.

- QueryPanel to receive user interactive queries
- NaiveBayesExplorer to generate explanations from the naive Bayes classifier
- No Reducer
- MotionPanelPresenter to represent the phone UI

### Home Activity Recognition

**Hidden Markov Model**

This uses the dataset from [Kasteren et al. 2008] about domestic activity, and train a HMM. The application takes 14 binary input sensors and infers which activity (out of 7) the user is performing. Explanations are presented by sensors and by time.

- QueryPanel to receive user interactive queries
- HMMExplorer to generate explanation from the HMM
- TimeReducer to aggregate evidence across time
- MotionPanelPresenter to represent time-step evidence of each sensor
- FloorplanPresenter to present evidence as a bubble in a floorplan

---

**Explanation Struct in Disjunctive Normal Form**

|------------|-----------------------------|-----------------------------|----------------------------------|

**Explainers**

Explainers support the automatic generation of 8 explanation types from supported decision models. 4 explanation types are model-dependent, while 4 are not. Explainers take queries and produce Explanation Structs.

<table>
<thead>
<tr>
<th>Explanation Types</th>
<th>Input</th>
<th>Output</th>
<th>What</th>
<th>Why</th>
<th>Why Not</th>
<th>How To</th>
<th>Certainty</th>
<th>What If</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-Independent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model-Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Queries**

Queries encapsulate different question types to put into Explainers. Some queries can take parameters to specify questions (e.g., asking Why Not about a target value).

queries: Inputs, Outputs, What, Why, Certainty

**Reducers**

Reducers simplify explanations so that they are easier for users to assimilate. There are two types: disjunction and conjunction, to reduce the number of reasons, and length of reasons, respectively.

<table>
<thead>
<tr>
<th>Reducers</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disjunction</td>
<td>OR-ed Evidence</td>
</tr>
<tr>
<td>Conjunction</td>
<td>AND-ed Reason</td>
</tr>
</tbody>
</table>

---

**Auto-Light Living Room UI**

- Room Enactor
- Room Viewer

**IM Response Prediction UI**

- Response Enactor
- IM Autostatus Viewer

**Mobile Phone Accelerometry UI**

- Motion Enactor
- Motion Viewer

**Home Activity Recognition UI**

- Home Enactor
- Home Viewer