A Context Based Approach for Application Personalization

Ph.D. Proposal
October 2\textsuperscript{nd} 2006
Anil Shankar

Evolutionary Computing Systems Lab (ECSL)
Dept. of Computer Science
University of Nevada, Reno
Application Personalization

Current Status

Is Clippy Annoying?

Application Personalization

- Do you pause your music player when your phone rings?
- Do you stop your music when you leave your room?
- Should your calendar remind you of an appointment if you are talking with someone in your office?
Application Personalization

• Do you pause your music player when your phone rings?
• Do you stop your music when you leave your room?
• Should your calendar remind you of an appointment if you are talking with someone in your office?

Anil Shankar
Outline

• Need for learning user preferences
• Related work in context-aware systems
• Sycophant
• Learning user preferences
• New results for 10 users
• Proposed work
Context

“Context is any information that can be used to characterize the situation of an entity”

“An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves”

- Anind K. Dey, Gregory D. Abowd, and Daniel Salber, 2001

Present day computers use an internal clock, keyboard activity, mouse movements to provide context for an application’s information processing.
Additional Context

• Advances have been made in
  – Speech and Natural Languages Processing
  – Computer Vision
  – Machine Learning
  – Human-Computer Interaction (HCI)
User-Context

“Any information regarding a user’s presence or absence in the vicinity of a computer”

Internal user-context:
keyboard activity, states of different user processes, mouse activity

External user-context: motion and speech in the user’s environment
Hypothesis

• Simple sensors
  – Web-camera instead of a retina scanner or a gaze-tracking device
  – Microphone instead of a voice-authentication system

• Simple user-context
  – motion detection instead of face recognition, speech detection instead of speech recognition, keyboard and mouse usage, activity of different user processes

User-context helps applications to better personalize themselves to individual users
Related Work

• Sensor devices to localize a user
  –Kulkarni’s *ReBa*, 2002

• Sensor-based statistical models to predict the state of interruptability of a user
  –Fogarty, 2005

• Quantitatively evaluate the effect of interruptions based on internal application events
  –Bailey, 2006
Our Work

• Use real sensors to gather user-context
• Model the desktop PC as a stationary robot with effectors (application action) and actuators (user model)
• In addition to predicting the state of interruptability of a user, we predict a user-preferred application action
Sycophant’s User Interface

• “Sycophant: A servile self-seeker who attempts to win favor by flattering influential people”
  – (source: http://dictionary.com)

• Framework that supports a calendaring application which learns to generate a user-preferred type of reminder for appointments and tasks
Sycophant’s Architecture

- Sensors collect internal and external user-context
- Machine Learning algorithms for building a user model
- User-feedback for reminder type
- Sensors operate in a binary mode
Sycophant’s Reminder Types

- No-reminder (type-0)
- Visual (type-1)
- Voice (type-2)
- Both Visual and Voice (type-3)
### User-Context Data

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-identifier</td>
<td>user-3</td>
</tr>
<tr>
<td>Appointment Time</td>
<td>10.3000</td>
</tr>
<tr>
<td>Motion (Count, All5, Any5, All1, Any1, Immed)</td>
<td>10,7,1,0,3,1</td>
</tr>
<tr>
<td>Speech (Count, All5, Any5, All1, Any1, Immed)</td>
<td>20,1,1,1,1,1</td>
</tr>
<tr>
<td>Process-1(Count, All5, Any5, All1, Any1, Immed)</td>
<td>20,1,1,1,1,1</td>
</tr>
<tr>
<td>Process-2(Count, All5, Any5, All1, Any1, Immed)</td>
<td>20,1,1,1,1,1</td>
</tr>
<tr>
<td>Process-3(Count, All5, Any5, All1, Any1, Immed)</td>
<td>0, 0,0,0,0,0</td>
</tr>
<tr>
<td>Process-4(Count, All5, Any5, All1, Any1, Immed)</td>
<td>20,1,1,1,1,1</td>
</tr>
<tr>
<td>Process-5(Count, All5, Any5, All1, Any1, Immed)</td>
<td>0, 0,0,0,0,0</td>
</tr>
<tr>
<td>Keyboard(Count, All5, Any5, All1, Any1, Immed)</td>
<td>0, 0,0,0,0,0</td>
</tr>
<tr>
<td>Mouse(Count, All5, Any5, All1, Any1, Immed)</td>
<td>0, 0,0,0,0,0</td>
</tr>
</tbody>
</table>

user-3, 10.30, 10, 7, 1, 0, 3, 1, 20, 1, 1, 1, 1, 1, 20, 1, 1, 1, 1, 1, 20, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 (no-reminder)
System Overview

Context Sensors → Sycophant (Calendar) → User → Context Data → Machine Learner → User Model → Sycophant (Calendar) → Reminders + Feedback → User

New Context Data → TRAINING

Anil Shankar
Learning User Preferences

• **Machine Learning:** A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$”

  -Tom Mitchell, *Machine Learning*

• **Techniques:** Decision Trees, Learning Classifier Systems
XCS Overview

TRAINING PHASE

User-1, 10.30, 8, 1, 0, 1, 1, 1: 1

001 - 101011110 - 0100 - 1-0-1-1-1

TESTING PHASE

10110101110 010010111: ?

If <condition based on user-context features>
Then <generate reminder-type N>

Rule-base

Action(1) Reward/Punish

Action(2)
**XCS**

- Genetics Based Machine Learning (GBML) technique

- **Working**
  - Sample a user-context data exemplar
  - Create matching-set and action-set
  - Choose an action, receive feedback, update fitness
Decision Tree

- Tree structure
- Nodes, leaf
- For classifying a case, you traverse the tree until you reach a leaf
- Example decision-rule:
  - If kbdCount > 4
    - And mouseAny5 = 1
      - And process1Any5 = 1
        - And mouseAll5 = 1
          - Then classify the reminder type as 1
## Results-1

Three different users

### Two-class problem

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Zero-R</th>
<th>J48</th>
<th>XCS</th>
<th>XCS', N</th>
<th>XCS vs J48</th>
<th>XCS' vs J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contex-Data-1</td>
<td>0.5677</td>
<td>0.7428</td>
<td>0.7460</td>
<td>0.7406, 5059</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>Contex-Data-2</td>
<td>0.5988</td>
<td>0.7885</td>
<td>0.9800</td>
<td>0.9300, 9678</td>
<td>Better</td>
<td>Better</td>
</tr>
<tr>
<td>Contex-Data-3</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000, 4722</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>Contex-Data-allusers</td>
<td>0.6115</td>
<td>0.8685</td>
<td>0.8791</td>
<td>0.8788, 7699</td>
<td>Equal</td>
<td>Equal</td>
</tr>
</tbody>
</table>

- J48 and XCS achieve a minimum accuracy of 74 percent across all the user-context data sets (Except in case of ContextData-3)
- J48 and XCS outperform Zero-R (baseline)
Results-1…contd

Three different users

Four-class problem

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Zero-R</th>
<th>J48</th>
<th>XCS</th>
<th>XCS', N</th>
<th>XCS vs J48</th>
<th>XCS' vs J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context-Data-1</td>
<td>0.5677</td>
<td>0.7000</td>
<td>1.0000</td>
<td>1.0000, 16289</td>
<td>Better</td>
<td>Better</td>
</tr>
<tr>
<td>Context-Data-2</td>
<td>0.5988</td>
<td>0.7267</td>
<td>1.0000</td>
<td>1.0000, 9433</td>
<td>Better</td>
<td>Better</td>
</tr>
<tr>
<td>Context-Data-3</td>
<td>0.9913</td>
<td>1.0000</td>
<td>0.9118</td>
<td>0.9118, 5071</td>
<td>Worse</td>
<td>Worse</td>
</tr>
<tr>
<td>Context-Data-allusers</td>
<td>0.3884</td>
<td>0.8236</td>
<td>0.8761</td>
<td>0.8236, 8493</td>
<td>Better</td>
<td>Better</td>
</tr>
</tbody>
</table>

- J48 and XCS achieve a minimum accuracy of 70 percent across all the user-context data sets (Except in case of ContextData-3)
- J48 and XCS outperform Zero-R (baseline)
- XCS outperforms J48 on the four-class problem
**Two-class problem, J48**

<table>
<thead>
<tr>
<th>Data-Set</th>
<th>Performance with External Context</th>
<th>Performance without External Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Data-1</td>
<td>74.3100</td>
<td>71.75</td>
</tr>
<tr>
<td>Context-Data-2</td>
<td>81.0700</td>
<td>77.68</td>
</tr>
<tr>
<td>Context-Data-3</td>
<td>100.0000</td>
<td>100</td>
</tr>
<tr>
<td>Context-Data-all-users</td>
<td>86.4100</td>
<td>83.38</td>
</tr>
</tbody>
</table>

Removing external context degrades J48’s performance
**Results-2**

**Four-class problem, J48**

<table>
<thead>
<tr>
<th>Data-Set</th>
<th>Performance with External Context</th>
<th>Performance without External Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Data-1</td>
<td>67.4300</td>
<td>65.99</td>
</tr>
<tr>
<td>Context-Data-2</td>
<td>72.0300</td>
<td>71.18</td>
</tr>
<tr>
<td>Context-Data-3</td>
<td>99.1400</td>
<td>99.14</td>
</tr>
<tr>
<td>Context-Data-all-users</td>
<td>80.8100</td>
<td>78.15</td>
</tr>
</tbody>
</table>

Removing external context degrades J48’s performance
Results-3…contd

Two-class problem, XCS

<table>
<thead>
<tr>
<th>Data-Set</th>
<th>Performance with External Context</th>
<th>Performance without External Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Data-1</td>
<td>0.7406</td>
<td>0.2254</td>
</tr>
<tr>
<td>Context-Data-2</td>
<td>0.9300</td>
<td>0.8000</td>
</tr>
<tr>
<td>Context-Data-3</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Context-Data-all-users</td>
<td>0.8788</td>
<td>0.4536</td>
</tr>
</tbody>
</table>

Removing external context degrades XCS’s performance
## Results-3

### Four-class problem, XCS

<table>
<thead>
<tr>
<th>Data-Set</th>
<th>Performance with External Context</th>
<th>Performance without External Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Data-1</td>
<td>1.0000</td>
<td>0.7745</td>
</tr>
<tr>
<td>Context-Data-2</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Context-Data-3</td>
<td>0.9118</td>
<td>0.9557</td>
</tr>
<tr>
<td>Context-Data-all-users</td>
<td>0.8236</td>
<td>0.5460</td>
</tr>
</tbody>
</table>

Removing external context degrades XCS’s performance

Anil Shankar
New Results
User Study Overview

• 10 users

• 4 separate sessions each lasting 45 mins.

• Read short/long articles

• Answer questions after reading the articles

• Hints (reminders) generated for users while reading

• User provides feedback specifying her preferred hint-type
## New Results

### Experimental Design

<table>
<thead>
<tr>
<th>Session</th>
<th>Task</th>
<th>Reminder Type* Order</th>
<th>Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Short Article</td>
<td>0,2, 3, 1</td>
<td>Talk, No-Music</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3, 1, 2, 0</td>
<td>Music, No-Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 0, 2, 3</td>
<td>No-Music, Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2, 1, 0, 3</td>
<td>Music, Talk</td>
</tr>
<tr>
<td>2</td>
<td>Long Article</td>
<td>3, 1, 0, 2</td>
<td>Talk, No-Music</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 2, 0, 3</td>
<td>Music, No-Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 3, 2, 0</td>
<td>No-Music, Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3, 0, 2, 1</td>
<td>Music, Talk</td>
</tr>
<tr>
<td>3</td>
<td>Short Article</td>
<td>1, 3, 0, 2</td>
<td>Talk, No-Music</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2, 3, 1, 0</td>
<td>Music, No-Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2, 0, 3, 1</td>
<td>No-Music, Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3, 2, 0, 1</td>
<td>Music, Talk</td>
</tr>
<tr>
<td>4</td>
<td>Long Article</td>
<td>0,3, 2, 1</td>
<td>Talk, No-Music</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 0, 2, 3</td>
<td>Music, No-Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 0, 2, 3</td>
<td>No-Music, Talk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3, 0, 1, 2</td>
<td>Music, Talk</td>
</tr>
</tbody>
</table>

*Reminder Types: 0 = None; 1 = Visual; 2 = Voice; 3 = Both

Anil Shankar
New Results

Two-Class Problem

Better than chance results (50)
Six users (1,3,4,5,6,9) better than average performance
Four users (2,7,8,10) below average performance
New Results

Four-class Problem

Better than chance results (50)
Six users (1,3,4,6,7,9) better than average performance
Four users (2,5,8,10) below average performance

Anil Shankar
Short-term study, 10 users…contd

Prediction accuracy on the four-class problem

<table>
<thead>
<tr>
<th>Learning</th>
<th>Original</th>
<th>No User</th>
<th>No External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Data</td>
<td>Context</td>
<td>Context</td>
</tr>
<tr>
<td>One-R</td>
<td>63.23</td>
<td>48.62</td>
<td>63.23</td>
</tr>
<tr>
<td>J 48</td>
<td>62.71</td>
<td>50.00</td>
<td>62.54</td>
</tr>
<tr>
<td>XCS</td>
<td>88.35</td>
<td>3126</td>
<td>62.51</td>
</tr>
</tbody>
</table>

Removing user-context degrades the performance of One-R, J48 and XCS
Removing External user-context degrades the performance of J48 and XCS

Anil Shankar
**Short-term study, 10 users…contd**

Prediction accuracy on the two-class problem

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Learning</th>
<th>Original</th>
<th>No User</th>
<th>No External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-R</td>
<td>Data</td>
<td>85.56</td>
<td>85.56</td>
<td>85.56</td>
</tr>
<tr>
<td>One-R</td>
<td>Context</td>
<td>86.42</td>
<td>86.42</td>
<td>86.42</td>
</tr>
<tr>
<td>J 48</td>
<td>Context</td>
<td></td>
<td>86.42</td>
<td>86.59</td>
</tr>
<tr>
<td>XCS</td>
<td></td>
<td>85.91</td>
<td>71.13</td>
<td>74.39</td>
</tr>
</tbody>
</table>

Removing external user-context or user-context degrade the performance of J48 and XCS
New Results Summary

• Learn preferences for individual users
• Predictive accuracy for reminder-types better than chance
• Learning preferences (four-classes) is harder than learning to interrupt (two-classes)
• User-context important for learning user-preferences
• Below average performance for four users: user2, user7, user8, user10
• Classifier System (XCS) is a promising approach for learning user preferences

Anil Shankar
Proposed Work

- Incorporate Sunbird (a calendaring application) into the Sycophant framework
- Collect short-term usage data
- Collect long-term usage data
- Experiment with other machine learning approaches
- Enable one more desktop application (xmms) to use our Sycophant framework
SYCOPHANT’S ARCHITECTURE

Anil Shankar
User-Context Aware Calendaring

Anil Shankar
Application Programming Interface (API)

1. Set up SENSORS
ms = Sensors('Motion Sensor')
ms.setLogFile('MotionLog.txt')
ms.setTickValue(15) // log data every 15 seconds into a sensor file
ms.addSensorService('Any15')
ms.logSensorData('Any15')

2. Set up FEATURE EXTRACTOR
fe = FeatureExtractor()
sfDict = {ms: 'Any15'}
UserContextData = fe.getUserContextData(sfDict)
3. Enable a Calendar application to use the API

```java
appapi = ApplicationAPI()
appapi.setAppParamToPredict('reminderType')
String[] appParams = {'user-id', 'appointment time', 'appointment text'}
appapi.setAppUserContextFeatures(appParams)
appParamVals = appapi.getAppUserContextData()
learner = 'xcs'
currUserModel = appapi.getUserModel(learner, UserContextData, appParamVals)
predictedReminder = appapi.predictParameter(currUserModel)
appapi updateUserModel(currUserModel)
```
User-Data Collection

• Design user-studies
• Collect short-term usage data
• Long-term data collection
• Disperse the software; collect data from a diverse set of users
Milestones

- **Fall 2006**
  - Context-enable Sunbird, distribute the user-context aware software bundle, start short-term data gathering

- **Spring 2007**
  - Gather and analyze preliminary results. Submit to conferences and a journal, start long-term data gathering

- **Summer 2007**
  - Refactor and refine the framework, experiment with additional machine learning approaches, Integrate XMMS – media player with the framework

- **Fall 2007**
  - Distribute the framework through the Open-Source community, Gather short-term usage data from XMMS, start long-term data collection

- **Spring 2008**
  - Publish results, Write proposal, defend Ph.D.
Proposed Enhancements

Learning Services Layer:
• Incorporate additional machine learning algorithms - Support Vector Machines and reinforcement learners

Applications Layer:
• Test the framework with at least one more application, say xmms (open-source music player for Linux)
Intended Contributions

HCI - Application Personalization

- A novel machine learning framework to enable applications learn user preferences based on user-context gathered from internal and external sensors
- Personalization to individual users
- Personalization across groups of users
Intended Contributions

Computer Security

• learning user patterns to enhance current intrusion detection methodologies

Robotics:

• stationary robot model of a desktop-PC
• provide more insights into human-robot interactions (social robotics)
Publications


Anil Shankar
Comments/Questions?

Anil Shankar
Sycophant's Application Programming Interface (API)

SENSORS API

Description: Manages sensors, sensor services and sensor data

Class Sensor:

Sensor(String SensorName):

Description - Creates a new sensor with SensorName

Returns - boolean indicating the status of the operation

def setLogFile(File SensorLogFile):

Description: Associates a log file for a sensor

Returns: boolean indicating whether a file was successfully created

def delete():

Description: Deletes the sensor

Returns: boolean indicating the status of the operation

def setTickValue(int tickSeconds):

Description: Sets the time interval during which sensor data is repeatedly logged to the sensor file

Returns: a boolean status of the operation
def addSensorService(String SensorServiceName):
    Description: Adds a new SensorService object to the sensor
    Returns: a boolean indicating the status of the operation

def deleteSensorService(SensorService service):
    Description: deletes a sensor service type associated with the sensor
    Returns: a boolean indicating the status of the operation

def logSensorData(String serviceName):
    Description: logs the service data value specified by serviceName to the sensor log file
    Returns: a boolean indicating the status of the operation

def getSensorData(SensorService service):
    Description: fetches the sensor services data
    Returns: a column of data formatted according to the specified service
Description: Creates and manages a sensor’s service and its data

def SensorService(String serviceFormat):
    Description: Creates a sensor service with the data formatting type specified by serviceFormat
    Returns: boolean status of the operation

def getServiceData(String serviceFormat):
    Description: Accesses the sensor’s log file and formats the data specified by the serviceFormat
    Returns: a column of sensor data formatted according to the specified serviceFormat
FEATURE EXTRACTOR API

Description: extracts sensor service features for use by the Application API

def getUserContextData(HashMap SensorServicesDict):

    Description: Gets user-context data columns for the sensors and their services specified in the SensorServicesDict

    Returns: Columns of user-context data with formatted features of sensor services

Anil Shankar
APPLICATION API

Description: Provides user-context features access for a machine learner

```python
def setAppParamToPredict(String paramName):
    Description: Sets which application parameter needs to be predicted
    Returns: a boolean indicating the status of the operation
```

```python
def setAppUserContextFeatures(String[] appParamNames):
    Description: Sets which application specific features need to be logged
    Returns: a boolean indicating the status of the operation
```

```python
def getAppUserContextData():
    Description: Provides current values of the application-specific user-context features
    Returns: String containing the values of the application-specific user-context features
```
def getUserModel(String learner, HashMap SensorServicesDict, String[] appFeatures):
    Description: Executes a learner and returns a user model built using the application-specific user-context features and the user-context features chosen from sensor-service dictionary
    Returns: a user-model data structure

def updateUserModel(UserModel userModelName):
    Description: Updates an existing user model
    Returns: a boolean indicating the status of the operation

def predictParameter(UserModel um):
    Description: uses the current user model to predict the application’s appParamToPredict
    Returns: predicted value of the application parameter to be predicted
## Results from short-term study

<table>
<thead>
<tr>
<th>DataSet</th>
<th>ZeroR</th>
<th>OneR</th>
<th>OneR_NoExt</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>0.5806</td>
<td>0.6290 (motionCount5)</td>
<td>0.5645 (mouseCount5)</td>
</tr>
<tr>
<td>user2</td>
<td>0.5000</td>
<td>0.6111 (mouseCount5)</td>
<td>0.5925</td>
</tr>
<tr>
<td>user3</td>
<td>0.7121</td>
<td>0.6515 (no preferences!, 1 def)</td>
<td>0.7121</td>
</tr>
<tr>
<td>user4</td>
<td>0.6200</td>
<td>0.56 (mouseCount5)</td>
<td>0.6200</td>
</tr>
<tr>
<td>user5</td>
<td>0.5454</td>
<td>0.5303 (motionCount5)</td>
<td>0.5303</td>
</tr>
<tr>
<td>user6</td>
<td>0.7142</td>
<td>0.6824</td>
<td>0.7142 (kbdCount)</td>
</tr>
<tr>
<td>user7</td>
<td>0.7460</td>
<td>0.9523 (motionAny1)</td>
<td>0.9206</td>
</tr>
<tr>
<td>user8</td>
<td>0.5625</td>
<td>0.5414</td>
<td>0.5625</td>
</tr>
<tr>
<td>user9</td>
<td>0.8163</td>
<td>0.7751 (kbdCount5)</td>
<td>0.8163</td>
</tr>
<tr>
<td>user10</td>
<td>0.5245</td>
<td>0.5245 (motionCount5)</td>
<td>0.4262</td>
</tr>
<tr>
<td>allUsers</td>
<td>0.4862</td>
<td>0.6323 (user-id)</td>
<td>0.6323</td>
</tr>
</tbody>
</table>
# Rules learned for different users

<table>
<thead>
<tr>
<th>User</th>
<th>Reminder types predicted by OneR</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>Visual and Voice Reminder</td>
</tr>
<tr>
<td>user2</td>
<td>Visual and Voice Reminder</td>
</tr>
<tr>
<td>user3</td>
<td>Visual Reminder</td>
</tr>
<tr>
<td>user4</td>
<td>Visual and Voice Reminder</td>
</tr>
<tr>
<td>user5</td>
<td>Voice Reminder</td>
</tr>
<tr>
<td>user6</td>
<td>Visual Reminder</td>
</tr>
<tr>
<td>user7</td>
<td>Visual Reminder</td>
</tr>
<tr>
<td>user8</td>
<td>Visual and Voice Reminder</td>
</tr>
<tr>
<td>user9</td>
<td>Visual Reminder</td>
</tr>
<tr>
<td>user10</td>
<td>Visual Reminder</td>
</tr>
</tbody>
</table>
Intended Contributions - 2

Learning Cognitive User Models

- Psychological theories of behavior to learn cognitive user models
- Psychological theories and computational cognitive models can mutually inform each other
- Designing user-studies to incorporate human factors and human psychology in our computationally generated user-models