Autonomous Construction of User Interaction Models in Generic Environments

User Interaction Models

- Models describe certain aspects of user understanding
  - DeMeo, Quattrone, and Ursino (2010) built a system which models user expertise/interests in a recommender system
  - Mairesse and Walker (2010) constructed a user model which was based on psychological profiles
  - Köck and Paramythis (2011) model user comprehension levels in an e-learning system

Problems with Existing Techniques

- Domain specific
  - Only applicable in a single context or even just a single system
  - Expert knowledge required to configure the model
- Require supervision
  - Data collection requires profiling per-user
  - Once built, model requires manual interpretation to apply

My Approach

- Identify user behavior patterns autonomously
  - Based on high frequency interactions with the system
  - Identify patterns of interaction
    - Adapt the interface to make these easier
  - Model is general and applicable to interaction with almost any component of the system
  - Process requires little to no expert analysis of each user
    - Works "off-the-shelf"
- Apply the user model at the top level of the interface
  - Create interface shortcuts
    - Complex or high frequency interactions can be shortened to a single interaction
  - Works on top of all components of the interface

Implementation - User Modeling

- Two major steps
  - Identification of patterns in interaction history
  - Prediction of future interactions
- Identification of patterns
  - Templating
  - Clustering
  - Classifier

Templating

- Create templates for types of related interface actions
  - Example:
    - Series of interactions with components in one application followed by an application switch
    - Period of inactivity followed by an application launch
- Loses some autonomy
  - Patterns which fit the templates can be detected autonomously
  - Anything not described by the templates needs intervention to create a new template
- Potentially fully expressive
  - With enough templates, cover all possible interesting patterns
  - Range of interface adaptations might limit the domain enough
- Might not scale well with enough templates
- Easy to implement

Clustering

- Identify related sequences
  - Repeated sequences form clusters
  - Large clusters correspond to important/common actions being performed by the user
- Relatedness is tough to define
  - Current data mostly consists of time-series data
  - With meta-information, what aspects are important?
- Making predictions using the model is not straightforward
  - Join new sequences with the existing clusters
  - Optimizations possible to reduce search space

Classifier

- Markov Model
  - Represent interaction history as states
  - Markov Model predicts state transitions
  - Naturally incorporates sequence information
- Dynamic Bayesian Network
  - Subsequences are nodes in the graph
  - Links between nodes are learned
- Pros
  - Patterns should be emergent properties
  - Learning is largely autonomous
  - Robust pattern detection
- Cons
  - Overconfident of low-frequency interactions
  - Condensing repeated interactions into a single node

Data Format and Collection

- Interaction datasets available
  - Only contain information about mouse click coordinates and time stamps
- Using Sikuli and other APIs we can collect more data about the nature of the interaction
  - Meta information about the system state during the interaction
  - Ideally an API to get a handle on the actual component
- Intended to be a useful tool in everyday interactions
  - For testing purposes, real user data will be collected in a controlled environment

Evaluation

- We can use a hold out set to evaluate the user model
  - Benefits from a large dataset
    - Must come from a single individual
  - Generalizability between users can be tested
    - Build a model using one user's dataset
    - Make predictions using another dataset
- Real-world applicability
  - Test with both lab data and real world data

Future Work

- Construct datasets
  - Identify important attributes for learning
- Evaluate methods of autonomous pattern recognition
- Implement as a working system/application
- Extend to modeling user knowledge/skills
  - E-learning domain
  - Card game domain

Conclusion

- AND IN CONCLUSION IT IS GREAT
- YEA