Testing in the Presence of Uncertainty

Robert Griffith, Kyung Min Han, Christopher Lee

Introduction

Software systems are continually evolving to be more sophisticated, resulting in increased complexity in development.

Complexity increases as it needs to handle deep interactions between different systems, users, and environments.

An outcome of this complexity is the growing uncertainty surrounding the behavior of software.

Occasionally, a degree of uncertainty is accepted in software development.

Researchers believe uncertainty should be treated as a fundamental part of software development.

Current approaches in identifying uncertainties are constrained, leaving treatment for uncertainty in testing open.

Examples of Uncertainty in Software

Google Maps

- Combines GPS, cellular tower, and WiFi data to estimate users device location, accepting variability for optimization.
- Computed locations may vary due to external factors and data fusion quality, with developers and users accepting approximate locations.

Context-Aware Music Recommender System (CAMRS)

- Uses smartphone sensors to select music that will complement a user's music activity.
- Uses machine learning techniques for the activity classifier that is not 100% accurate.
- Assuming that users will allow less-than-perfect recommendations, developer accepts the uncertainty.

Quadcopter Stabilization

- Altitude adjustment using sensors to change thrust values during flight.
- Precision and accuracy heavily depended by sensors but uncertainty is introduced by environment and physical components.
- Users are willing to accept temporary oscillations from conservative controller configurations resulting from the uncertainty.

Uncertainty in Testing Today

Two ways to deal with uncertainty.

1. Control over unit testing.
2. Constrain the testing environment.

Controlling uncertainty is achievable with small units but increases in difficulty as units become larger.

Types of uncertainties include:

1. Aleatoric Uncertainty - Uncertainties on system dealing with physical world are inherent.
2. Epistemic Uncertainty - Increases in complexity and cost but resolvable with enough effort (e.g. over-engineering).

Generally, testing environments cannot fully control all aspects of uncertainty.

Uncertainty in Testing Today

Another method of testing uncertainty includes developing sophisticated oracles.

If oracles are not properly implemented then false positive results and masked faults could occur.

Oracles must be able to differentiate between an acceptable report and a fault.

Utility functions can improve oracles differentiation although requires knowledge of input distribution and result likelihood.

Cost and complexity are factors in developing sophisticated oracles.

Unfortunately, current testing measures are not as sufficient in controlling all aspects of uncertainty.

Future Handling of Uncertainty

Testing requirements for sufficient handling of uncertainty:

1. Richer testing frameworks for specifying input distributions and favoring uncertainty discovery.
2. Probabilistic oracles for distinguishing acceptable misbehaviors and specifying result likelihood.
3. Richer models to represent system and environment uncertainty, to test requirements and outcomes, and enable automated uncertainty quantification.

Hidden Markov Models (HMMs) offer a potential solution to infer hidden properties of systems based on observations, aiding in uncertainty quantification.

HMMs can model systems for testing by representing activities as states and observations, controlling transition probabilities by the tester, and using confusion matrices to determine emission probabilities.

Conclusion

No previous research in resolving issues testing systems with acceptable misbehaviors.

Researchers have used statistical reasoning in testing, specifically in software reliability testing, involving statistical sampling and reliability growth models.

BUT, those approaches are under conventional assumption that outputs are either acceptably correct or unacceptably incorrect.

HMMs offer one approach to handling uncertainty, with potential for algorithmic refinement and specialized statistical techniques.

Statistical methods can be essential for improving the tester's ability to reveal faults masked by uncertainty.

Reference

S. Elbaum and D. Rosenblum. Known Unknowns: Testing in the Presence of Uncertainty. November 2014.