Bayesian Reasoning for Software Testing

Position Paper

Akbar Siami Namin
Department of Computer Science
Texas Tech University, USA
akbar.namin@ttu.edu

Mohan Sridharan
Department of Computer Science
Texas Tech University, USA
Mohan.sridharan@ttu.edu

International Workshop on Future of Software Engineering (FoSER 2010)
Santa Fe, NM, USA, November 2010
Outline

- Motivation
- Bayesian reasoning and software testing
- Practical challenges
Motivation

- Many software testing challenges are NP-hard problems.
- Notkin[26]:
  - “we may need to approach testing and analysis more like theoreticians pursue NP-hard problems”
  - “in the absence of efficient, precise algorithms, the theoreticians pursue probabilistic and epsilon-approximate algorithms”

Motivation

Problems with Existing Software Testing Approaches

- Specificity
  - Always quasy-experimental studies
- Intractability
  - Infinite number of test inputs
- Inability to adopt
  - Unable to account for the uncertainties
Motivation

Software Testing and Machine Learning

- Software engineering is a fertile ground and many software engineering problems can be formulated as learning problem using machine learning techniques\[35\]

- Machine learning techniques
  - Offline learning
    - The rigid models are developed
      - E.g. Decision trees, SVMs
  - Online learning
    - Adaptive algorithms
      - E.g. Bayesian reasoning, MDP

\[35\] D. Zhang and J.J.P. Tsai. Machine Learning and software engineering.
Motivation

Software Testing and Bayesian Reasoning

- Software testing is among the most challenging domains for machine learning over the next ten years [11]
- Most of software testing problems are a clean application for machine learning and Bayesian reasoning
- Though
  - Offline learning has been used extensively
  - Online learning has not been utilized enough

Bayesian Reasoning & Software Testing

Probabilistic Representations

- Probabilistic representation for modeling uncertainties
- Tracking multiple hypotheses about the state of system
  - A higher probability
    - A higher likelihood that a hypothesis is true
- Bayesian reasoning
  - Incrementally updates the believes
Bayesian Reasoning & Software Testing

Basic Form of Bayes Rule

\[ p(a \mid b) = \frac{p(b \mid a)p(a)}{p(b)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{normalizer}} \]

- Computing the posterior (conditional) probability of event \( a \) given \( b \)
  - Based on:
    - Likelihood \( p(b \mid a) \)
    - Prior probability \( p(a) \)
    - Probability \( p(b) \), the normalized
Bayesian Reasoning & Software Testing

**Bayes Rule for Multi-Class Classification**

\[
p(C_i \mid z) = \frac{p(z \mid C_i) \cdot p(C_i)}{\sum_{j=1}^{N} p(z \mid C_j) \cdot p(C_j)}
\]

- Classification with classes $C$’s
  - $N$: the number of classes
- $P(C \mid z)$: Incrementally updates the probability of each $C$ given observation $z$
- $P(z \mid C)$: The prior likelihood
- $P(C)$: The prior probability of this class
Goal: estimate the probabilistic belief of system state $x$ at time $t$:

$$t : bel(x_t) = p(x_t \mid x_0, u_1, z_1, ..., x_{t-1}, u_t, z_t)$$

The Markov assumption:

$$p(x_t \mid x_0, u_1, z_1, ..., x_{t-1}, u_t, z_t) = p(x_t \mid x_{t-1}, z_t, u_t)$$

The state at time $t$ can be estimated conditionally independent of all prior states, actions and observations:

- Observed through a series of observations $z$
- The observations $z$ obtained through a set of actions:

$$u : \{u_1, z_1, ..., x_{t-1}, u_t, z_t \}$$
Bayesian Reasoning & Software Testing

Monte Carlo Sampling

- Applicable to domains with multiple hypotheses
- Each sample is an instance of a hypothesis
  - Associated with a probabilistic value representing the likelihood the hypothesis is true
- The general procedure:
  - A small set of samples are selected initially
  - Each hypothesis is modified to account for any change
  - The probability of each hypothesis is updated
  - A larger number of samples are selected for hypotheses with larger probability values
Bayesian Reasoning & Software Testing

Application

- Monte Carlo importance sampling
  - Mutation testing
    - Already studied by authors [28]
  - Applicable to
    - Statistical fault localization
    - Adaptive random testing
    - Static analysis
    - Probabilistic model checking
    - Etc.

[28] M. Sridharan, A. Siami Namin, Prioritizing Mutation Operators based on Probabilistic Sampling
Practical Challenges

- Generalization issues
  - Remember “External Threats” at the end of most papers
  - Probabilistic representations are robust to such issues
- Sensitivity to priors
  - In addition, estimating the likelihood function
  - The performance of Bayesian reasoning is robust to such issues, i.e. Convergence takes longer
- Steep learning curve
  - Difficulty in learning and using statistics and probability
Thank You