Supplementary Material for "Localizing Bugs in Program Executions with Graphical Models"

Laura Dietz
Max-Planck Institute for Computer Science
Saarbruecken, Germany
dietz@mpi-inf.mpg.de

Valentin Dallmeier
Saarland University
Saarbruecken, Germany
dallmeier@cs.uni-saarland.de

Andreas Zeller
Saarland University
Saarbruecken, Germany
zeller@cs.uni-saarland.de

Tobias Scheffer
Potsdam University
Potsdam, Germany
scheffer@cs.uni-potsdam.de

Derivation of the Predictive Distribution

Given a collection $\mathcal{G}_m$ of previously seen execution graphs for method m and a new execution $G_m = (V_m, E_m, L_m)$, Bayesian inference determines the likelihood $p((u,v) \in E_m | V_m, \mathcal{G}_m, \alpha_\psi, \beta_\psi)$ of each of the edges (u,v), thus indicating unlikely transitions in the new execution of m represented by execution graph G_m. Since we employ independent models for all methods m, inference can be carried out for each method separately.

In order to infer the probability of an edge, Equation 1 integrates over the model space.

$$p((u,v) \in E_m | V_m, \mathcal{G}_m, \alpha_\psi, \beta_\psi) = \int p((u,v) \in E_m | V_m, \Psi) p(\Psi | \mathcal{G}_m, \alpha_\psi, \beta_\psi) \, d\Psi \tag{1}$$

According to the Bernoulli graph model, the likelihood of the existence of an edge (u,v) given the parameter vector Ψ is a Bernoulli distribution. The distribution is conditioned on existence of the start vertex u and yields zero probability if the labels do not overlap appropriately:

$$p((u,v) \in E_m | V_m, \Psi) = \begin{cases} \psi_{m,s_1 \ldots s_n} & \text{if } u \in V_m,\, L_m(u) = s_1 \ldots s_{n-1}, \text{ and } L_m(v) = s_2 \ldots s_n \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Corresponding to Figure 3a), the random variable $b_{G,u,s} =$ true iff an edge $(u,v) \in G$ exists in the graph such that $L(u) = s_1 \ldots s_{n-1}$ and $L(v) = s_2 \ldots s_{n-1} s$. The likelihood of a graph is proportional to the likelihood that all edges in E_G are generated ($b_{G,u,s} =$ true) and all edges in the complementary set $\overline{E}_G$ are not (Equation 3). If the start vertex u is not contained, random variables b are false with probability 1, thus we can omit them from the product over $\mathcal{G}$ in Equation 4. This product with shared parameter ψ yields a Binomial distribution. Since the product of Binomial and Beta distributions yields a reparametrized Beta distribution, we arrive at Equation 5 using counts of successful and failed trials. Shorthand $\#^{\mathcal{G}}_{(u,s_n)}$ abbreviates the number of graphs $G \in \mathcal{G}$ with $b_{G,u,s_n} =$ true which is the case if an edge (u,v) between vertices labeled $L(u) = s_1 \ldots s_{n-1}$ and $L(v) = s_2 \ldots s_n$ exists; and $\#^{\mathcal{G}}_u$ refers to the number of graphs $G \in \mathcal{G}$ that have a vertex u labeled $s_1 \ldots s_{n-1}$, in which case a draw from ψ is issued.

$$
\begin{aligned}
& p(\Psi|\mathcal{G}_m, \alpha_\psi, \beta_\psi) \\
& \quad \propto \quad p(\Psi|\alpha_\psi, \beta_\psi) \prod_{G \in \mathcal{G}_m} \underbrace{p(V_G)}_{\text{const}} p(E_G|V_G, \Psi) \left(1 - p(\overline{E}_G|V_G, \Psi)\right) && (3) \\
& \quad \propto \quad \prod_{s_1 \ldots s_{n-1}} \prod_{s \in S} \Bigg(p_\beta(\psi_{m, s_1 \ldots s_{n-1} s}|\alpha_\psi, \beta_\psi) \prod_{\substack{G \in \mathcal{G}_m | u \in V_G \\ L(u) = s_1 \ldots s_{n-1}}} p(b_{G,u,s}|\psi_{m, s_1 \ldots s_{n-1} s}) \Bigg) && (4) \\
& \quad \propto \quad \prod_{s_1 \ldots s_n \in (S_m)^n} p_\beta \left(\psi_{m, s_1 \ldots s_{n-1} s_n} | \#^{\mathcal{G}}_{(u, s_n)} + \alpha_\psi, \#^{\mathcal{G}}_u - \#^{\mathcal{G}}_{(u, s_n)} + \beta_\psi \right) && (5)
\end{aligned}
$$

The predictive distribution in Equation 1 using a Beta-distributed posterior has an analytic solution:

$$
p((u, v) \in E_m | V_m, \mathcal{G}_m, \alpha_\psi, \beta_\psi) = \frac{\#^{\mathcal{G}}_{(u, s_n)} + \alpha_\psi}{\#^{\mathcal{G}}_u + \alpha_\psi + \beta_\psi}. \qquad (6)
$$

By definition, an execution graph G for an execution contains a vertex if its label is a substring of the execution's trace t. Likewise, an edge is contained if an aggregation of the vertex labels is a substring of t. It follows that $\#^{\mathcal{G}}_u = \#\{t \in T | s_1 \ldots s_{n-1} \in t\}$ and $\#^{\mathcal{G}}_{(u, s_n)} = \#\{t \in T | s_1 \ldots s_n \in t\}$. Equation 6 can be reformulated as in Equation 7 to predict the probability of seeing the code position $\tilde{s} = s_n$ after a fragment of preceding statements $\tilde{f} = s_1 \ldots s_{n-1}$ using the trace representation of an execution. Thus, it is not neccessary to represent execution graphs G explicitly.

$$
p(\tilde{s}|\tilde{f}, T, \alpha_\psi, \beta_\psi) = \frac{\#\{t \in T | \tilde{f}\tilde{s} \in t\} + \alpha_\psi}{\#\{t \in T | \tilde{f} \in t\} + \alpha_\psi + \beta_\psi} \qquad (7)
$$