Transfer Learning for Improving Model Predictions in Highly Configurable Software

Pooyan Jamshidi, Miguel Velez, Christian Kästner Norbert Siegmund, Prasad Kawthekar

Carnegie Mellon University
Performance Influence of Configuration Parameters

drpcc.port: 3772
drpcc.worker.threads: 64
drpcc.max_buffer_size: 1048576
drpcc.queue.size: 128
drpcc.invocations.port: 3773
drpcc.invocations.threads: 64
drpcc.request.timeout.secs: 600
drpcc.childopts: "-Xmx768m"
drpcc.http.port: 3774
drpcc.https.port: -1
drpcc.https.keystore.password: ""
drpcc.https.keystore.type: "JKS"
drpcc.authorizer.acl.filename: "drpc-auth-acl.yaml"
drpcc.authorizer.acl.strict: false
transactional.zookeeper.root: "/transactional"
transactional.zookeeper.servers: null
transactional.zookeeper.port: null

## blobstore configs
supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
supervisor.blobstore.download.thread.count: 5
supervisor.blobstore.download.max_retries: 3
supervisor.localizer.cache.target.size.mb: 10240
supervisor.localizer.cleanup.interval.ms: 600000
Classic Sensitivity Analysis
Classic Sensitivity Analysis

Measure \rightarrow \text{Data} \rightarrow \text{Learn} \rightarrow 50 - 3*\text{C}_1 + 20*\text{C}_2 - 3*\text{C}_1*\text{C}_3
Classic Sensitivity Analysis

Measure → Data → Learn

50 - 3*C1 + 20*C2 - 3*C1*C3

Reasoning + Adaptation + Debugging + Optimization
Classic Sensitivity Analysis

High measurement costs

Measure → Data

Learn

50 - 3*C1 + 20*C2 - 3*C1*C3

Reasoning + Adaptation + Debugging + Optimization
Idea: Transfer Learning

Measure

Learn w/ Transfer Learning

50 - 3*C1 + 20*C2 - 3*C1*C3

Reuse

Reasoning + Adaptation + Debugging + Optimization
Exploiting Similarity

(a) source samples
+ target samples

(b) source samples
+ target samples
- \( \hat{f}(x) \)
- \( f(x) \)
- \( \sigma(x) \)
GP for modeling black box response function

\[ y = f(x) \sim GP(\mu(x), k(x, x')) \],

\[
\mu_t(x) = \mu(x) + k(x)^T(K + \sigma^2I)^{-1}(y - \mu)
\]

\[
\sigma_t^2(x) = k(x, x) + \sigma^2I - k(x)^T(K + \sigma^2I)^{-1}k(x)
\]

Motivations:
1- mean estimates + variance
2- all computations are linear algebra
3- good estimations when few data

\[ K := \begin{bmatrix}
k(x_1, x_1) & \cdots & k(x_1, x_t) \\
\vdots & \ddots & \vdots \\
k(x_t, x_1) & \cdots & k(x_t, x_t)
\end{bmatrix}
\]

\[ k(f, g, x, x') = k_t(f, g) \times k_{xx}(x, x') \],
Scenarios and Assumptions

Environment change  (configuration option vs. environment change)
  Different benchmark/workload
  Different program version
  Different hardware

Shape of old and new model similar
Some Results ("it works")
First Feasibility Demonstration
[SEAMS 2017]

Case study & controlled experiments
Can we improve prediction accuracy?
Tradeoffs among #source and #target samples?
Fast enough?

Subject sys.: Cobot, Apache Storm, Cassandra
Example: Performance prediction for CoBot
Example: Performance prediction for CoBot
Example: Performance prediction for CoBot

Source Model

Target Model

Prediction w/ Transfer Learning

Prediction from 4 samples
Accuracy and Costs
Accuracy and Costs
Future work, insights and ideas
Selecting from Multiple Sources (Cost Model)

Source Robot \rightarrow Target Robot

Source Simulator \rightarrow Target Simulator

(1) \rightarrow (2)

(3) \rightarrow (2)

(4) \rightarrow (5)
Checking Assumptions

How similar are source and target models for real environment changes (workload/infrastructure/code changes)?

Expected similarities:
• Constant
• Constant / proportional shift
• More noise but similar trends
• Many features and interactions with similar impact
• Many important features and interactions remain important
**Goal**: find best sample points iteratively by gaining knowledge from source and target domain

**Active learning + transfer learning**

- Measure
- Learn with Transfer Learning
- 50 - 3*C1 + 20*C2 - 3*C1*C3
- Reasoning + Adaptation + Debugging + Optimization

Reuse

Data

Mathematical expression:

- $50 - 3C_1 + 20C_2 - 3C_1C_3$
Transfer Learning for Improving Model Predictions in Highly Configurable Software

Improves the model accuracy up to several orders of magnitude

Is able to trade-off between different number of samples from source and target enabling a cost-aware model

Exploring similarity across environment changes and active learning