The Which

Zachery A. Milton

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Tim Menzies, Ph.D., Chair
Arun Ross, Ph.D.
Katerina Goseva-Popstojanova, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2008

Keywords: Data Mining, Contrast Set Learning, Software Defect Detetection, (???)

Abstract

Evaluation Bias in Effort Estimation

Zachery A. Milton

Abstract

Dedication

To My Family

Acknowledgments

Acknowledgments

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mention things like motivation and state the theis

X is a problem since B. previously, R said to use D but we tried D and we saw E. We hypothesize F. We checked it using U and saw H. Therefore, in the future, we recommendP for X.

1.1 Contribution of This Thesis

Theoretical:

- stated in the abstract and intro of tse paper draft 9
- papers in workshops, etc.
- to current evaluation methodology

Practical:

- helps in JPL's tool

1.2 Structure of This Document

Outline the remainder of the chapters.

Chapter 2

Related Work

give some background about why I started this work.

2.1 Data Mining

2.1.1 Contrast Set Learning

2.2 Manual Selection

2.2.1 Experts

2.3 Automated Selection

2.3.1 Previous Experiments

Chapter 3

Which

3.1 Idea Behind Which

Which is a stochastic best first search. The idea behind this is a method of searching that relies on previous knowledge that has already been discovered. Essentially, Which uses what it has previously discovered via expansion to attempt to take great leaps through the search space to approach a solution quicker. (NEEDS TALK OF HOW WHICH SPAWNED FROM A SIMPLER TAR3)

3.1.1 Best First Search Implementation

The basic idea of Which is to use a stochastic best first search[1]. As described in [1] a best first search is a method of searching a tree by exploring the nodes that score well based on a heuristic function first. This expansion continues down the tree until a desired solution is found. This function may depend on the description of the node, the description of the goal, the description of the path from the root to the current node, and any heuristic knowledge of the domain. Which itself mainly uses the path as the description of the node as well as the description of the goal to score its nodes. However, as is explained in section 4.2.4, this scoring heuristic is completely arbitrary

to the core Which functionality.

The idea behind using the best first search here is a little different than the definition above. Which has no direct root, but instead one root for each attribute in isolation. Which will immediately score each attribute's ranges independently of each other. After this phase Which consists of n attributes each with m nodes expanded. It is here that Which decides which attribute range to expand next. Its method of doing this is to choose two paths from two trees, combine them, and then score them to see if they score well. If this is a very promising path, there is a good chance that Which will attempt to combine this new node with a different node. This process of combining is further explained in sectiole produced has a disjunction, instead of a conjunction. That is, if two ranges of the same attribute are picked, Which will join them together as a disjunction. The good above relates to the method in which Which will pick the next to paths to combine. This is explained below in section 4.2.2. Which will continue to pick two paths, combine them, and score them until one of the possible stopping conditions listed in section 4.2.5 occurs.n 4.2.1 below. The two attribute-ranges that are picked can be from the same tree. In this case the rule produced has a disjunction, instead of a conjunction. That is, if two ranges of the same attribute are picked, Which will join them together as a disjunction. The good above relates to the method in which Which will pick the next to paths to combine. This is explained below in section 4.2.2. Which will continue to pick two paths, combine them, and score them until one of the possible stopping conditions listed in section 4.2.5 occurs.

3.1.2 Implementation

3.1.3 Sorted Linked List

3.1.4 Rule Combination

3.1.5 Probabilistic Selection

3.1.6 Scoring Functionality

3.1.7 Stopping Conditions

3.1.8 The Finite List

3.2 Advantages

3.3 Experiments with Which's Parameters

3.3.1 Changing the Maximium Selection Count

3.3.2 Changing the Check Parameters

Check Every

Improvement

3.3.3 Changing the List Size

Chapter 4

Experiments

4.1 TAR3

4.1.1 Comparison of TAR3 and Which

4.1.2 Design of Experiments

4.1.3 Which's Heuristic

Data Used in Experiments

Evaluation Criteria

4.1.4 Results

4.2 Defect Detection Data

4.2.1 Design of Experiments

4.2.2 Which's Heuristic

Balance

4.2.3 Evaluation Criteria

4.2.4 "The Koru Diagram"

4.2.6 MDP Data

Data Description

Results

4.2.7 AT&T Data

Data Description

Results

4.2.8 Turkey Data

Data Description

Results

4.3 Micro Sampling

4.3.1 What is Microsampling?

Chapter 5

Conclusion

5.1 Future Work

Look at the references in the back of the TSE paper and the references from those references.