Bayesian Methods for Data Analysis in Software Engineering

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

Akbar Siami Namin
Advanced Empirical Software Testing and Analysis Research Group (AVESTA)
Department of Computer Science
Texas Tech University
akbar.namin@ttu.edu

ABSTRACT

Software engineering researchers analyze programs by applying a range of test cases, measuring relevant statistics and reasoning about the observed phenomena. Though the traditional statistical methods provide a rigorous analysis of the data obtained during program analysis, they lack the flexibility to build a unique representation for each program. Bayesian methods for data analysis, on the other hand, allow for flexible updates of the knowledge acquired through observations. Despite their strong mathematical basis and obvious suitability to software analysis, Bayesian methods are still largely under-utilized in the software engineering community, primarily because many software engineers are unfamiliar with the use of Bayesian methods to formulate their research problems.

This tutorial will provide a broad introduction of Bayesian methods for data analysis, with a specific focus on problems of interest to software engineering researchers. In addition, the tutorial will provide an in-depth understanding of a subset of popular topics such as Bayesian inference, probabilistic prediction techniques, Markov models, information theory and sampling. The core concepts will be explained using case studies and the application of prominent statistical tools on examples drawn from software engineering research. At the end of the tutorial, the participants will acquire the necessary skills and background knowledge to formulate their research problems using Bayesian methods, and analyze their formulation using appropriate software tools.

1. GOAL AND OBJECTIVES

This tutorial aims to introduce software engineering researchers and practitioners to the field of Bayesian analysis. The primary goal is to provide a broad perspective of the core theoretical concepts of Bayesian methods. In addition, we will focus in-depth on a set of methods that can be used to formulate problems of interest to the software engineering community. These methods and the associated topics will be explained through case studies and examples drawn from software engineering research. At the end of the tutorial, the participants will have acquired the following skills:

- The ability to identify the applicability of Bayesian methods to challenges in software engineering, and hence to their own specific research problems.
- The ability to translate their own research problems into the Bayesian framework, and choose the methods most appropriate to the problem under consideration.
- The ability to differentiate between traditional and Bayesian statistical data analysis, and to understand the pros and cons of these approaches.
- The ability to use existing Bayesian data analysis tools in their research projects.
- The ability to setup and conduct rigorous experiments to evaluate the relevant hypotheses in the domain under consideration.

2. MOTIVATION

Over the last few decades, software engineering research has established strong ties with empirical and statistical methods for knowledge discovery through data analysis. Researchers evaluate their techniques on a relatively small subset of the software programs, identify important characteristics of the proposed solutions, and use the acquired knowledge to predict the effectiveness of the proposed approaches on the entire population of programs. Such attempts at knowledge discovery typically use statistical methods for data analysis. The key challenge is that there are uncertainties associated with the observed behavior of each program during data analysis. In addition, it is often infeasible to run all possible tests to completely analyze a given program. Even though some behaviors may remain constant across different types of analysis performed on a program (or even across a set of similar programs), it is unlikely that we can accurately predict the behavior on the basis of limited data analysis. Hence, there is a potential external threat to any result obtained through a traditional statistical data analysis of the programs. These statistical methods do not offer the flexibility required to build unique representations for each program. Hypotheses are typically evaluated using established statistical measures, and it is difficult to fully exploit new observations that may play an important role in the evaluation of the hypotheses. Traditional statistical techniques are hence insufficient to address the inherent stochasticity of program analysis.

Bayesian methods, on the other hand, provide the math-

ICSE'10, May 2–8, 2010, Cape Town, South Africa.

ematical basis for elegantly incorporating new observations that may influence the prior beliefs about the hypotheses under consideration [3]. Bayesian methods account for this stochasticity in program analysis by explicitly building probabilistic models of the uncertainty. Specifically, Bayesian methods have at their core, a probability measure associated with the current belief about the program and hypotheses being analyzed, and these beliefs are updated over a period of time as new evidence becomes available. These properties have resulted in the use of Bayesian methods in a wide range of problems in many fields in computer science and engineering such as machine learning, computer vision and robotics [3]. Despite their obvious suitability to software engineering problems, Bayesian methods are still largely under-utilized in the software engineering community. The major challenge to the widespread use of Bayesian methods is the unfamiliarity with the strong mathematical foundations of such methods. Many software engineers therefore find it challenging to exploit the representational power of Bayesian techniques in order to formulate their research problems. A close look at the recent papers published in the premier software engineering conferences reveals that only recently has the focus shifted towards applying advanced stochastic methods for data analysis [1, 2, 4, 6, 10]. Knowledge of Bayesian methods will therefore enable researchers to make significant contributions to the field of software engineering.

This tutorial will prepare software engineering researchers and practitioners for using Bayesian methods in their research projects. Our goal is to provide a broad perspective of the core theoretical concepts of Bayesian methods and probabilistic analysis. The tutorial will therefore begin with an overview of the related concepts in probability and statistics. Next, we will introduce Bayesian inference and some relevant concepts from the field of information theory such as entropy and bootstrapping. We will then delve into some popular Bayesian techniques that can be applied to the current challenges of interest to the software engineering community. Towards this end, we will provide an in-depth description of selected topics such as information theory-based model checking, Markov models and Markov decision processes, and the powerful concept of sampling. All the concepts described in this tutorial will be grounded using illustrative examples drawn from the topics of current interest to software engineering researchers.

3. STRUCTURE OF CONTENTS

This full-day tutorial concentrates on the use of Bayesian methods for data analysis in software engineering. Below, we describe the content of the tutorial:

- Introduction: importance of data analysis in software engineering; challenges and requirements.
- Statistical analysis: hypothesis testing; statistical significance, regression analysis; demonstrations using statistical analysis tools.
- The importance of Bayesian analysis: drawbacks of existing data analysis approaches; Bayesian inference and applicability to software engineering problems.
- Illustrative software engineering examples.
- Bayesian inference: introduction to probabilistic analysis; joint, conditional and marginal distributions; Bayes' rule; Bayesian classification and regression; information theory; univariate and multivariate analysis; demos using MATLAB [7], R [9], and WinBUGS [8].
- Model checking and information theory: bootstrap testing of hypotheses; statistical information gain and entropy reduction; Bayesian model analysis; applications to software engineering problems.
- Markov models: introduction and Markov property; Markov chains; Markov decision processes (MDPs) and Partially Observable Markov Decision Processes (POMDPs); applications to software engineering.
- Sampling methods: Gibbs sampling, MonteCarlo sampling, stochastic sampling; applications to software testing and program analysis using WinBUGS.

Given the extensive experience of one of the presenters in robotics, the tutorial will include (if time permits) a live demo of the application of Bayesian methods on a mobile robot platform.

The slides, datasets and other resources used in the tutorial will be made available to all the participants. The more mathematically inclined members of the audience can look at [3, 5] for details regarding the core principles underlying the techniques being discussed in the tutorial.

4. REFERENCES

[1] G. K. Baah, A. Gray, and M. J. Harrold. On-line Aanomaly Detection of Deployed Software: A Statistical Machine Learning Approach. In *SOQUA*, pages 70–77, 2006.

[2] G. K. Baah, A. Podgurski, and M. J. Harrold. The Probabilistic Program Dependence Graph and its Application to Fault Diagnosis. In *ISSTA*, pages 189–200, 2008.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2008.

[4] L. C. Briand. Novel Applications of Machine Learning in Software Testing. In *The Eighth International Conference on Quality Software*, pages 3–10, Washington, DC, USA, 2008. IEEE Computer Society.

[5] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and HALL/CRC, 2004.

[6] J.-J. Gras, R. Gupta, and E. Perez-Minana. Generating A Test Strategy with Bayesian Networks and Common Sense. *Academic and Industrial Conference on Practice And Research Techniques, Testing*, 0:29–40, 2006.

[7] The Mathworks Website. `http://www.mathworks.com/products/matlab/`.

[8] T. B. Project. *Bayesian Inference Using Gibbs Sampling*. MRC Biostatistics Unit, Cambridge, UK, 1997.

[9] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.

[10] D. A. Wooff, M. Goldstein, and F. P. A. Coolen. Bayesian Graphical Models for Software Testing. *IEEE Trans. Softw. Eng.*, 28(5):510–525, 2002.