Bayesian Methods for Data Analysis in Software Engineering

Tutorial Outline and Introduction to Probability

Mohan Sridharan[1] Akbar Siami Namin[2]

[1] Stochastic Estimation and Autonomous Robotics (SEAR) Lab
Department of Computer Science
Texas Tech University

[2] AdVanced Empirical Software Testing and Analysis (AVESTA) Research Group
Department of Computer Science
Texas Tech University

May 3, 2010; Cape Town

Tutorial Schedule

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Tutorial Schedule

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Tutorial Schedule

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Tutorial Schedule

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Session 1: Introduction

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Adaptive Program Analysis

- *Software engineering challenges are inherently stochastic!*
- Each program or software module displays unique characteristics.
- A program's behavior can vary in response to different test cases.
- Standard statistical approaches are insufficient to model the uncertainty.
- Reliability can be achieved *only* by explicitly modeling the stochasticity.

Adaptive Program Analysis

- *Software engineering challenges are inherently stochastic!*
- Each program or software module displays unique characteristics.
- A program's behavior can vary in response to different test cases.
- Standard statistical approaches are insufficient to model the uncertainty.
- Reliability can be achieved *only* by explicitly modeling the stochasticity.

Adaptive Program Analysis

- *Software engineering challenges are inherently stochastic!*
- Each program or software module displays unique characteristics.
- A program's behavior can vary in response to different test cases.
- Standard statistical approaches are insufficient to model the uncertainty.
- Reliability can be achieved *only* by explicitly modeling the stochasticity.

Adaptive Program Analysis

- *Software engineering challenges are inherently stochastic!*
- Each program or software module displays unique characteristics.
- A program's behavior can vary in response to different test cases.
- Standard statistical approaches are insufficient to model the uncertainty.
- Reliability can be achieved *only* by explicitly modeling the stochasticity.

Adaptive Program Analysis

- *Software engineering challenges are inherently stochastic!*
- Each program or software module displays unique characteristics.
- A program's behavior can vary in response to different test cases.
- Standard statistical approaches are insufficient to model the uncertainty.
- Reliability can be achieved *only* by explicitly modeling the stochasticity.

Why Bayesian Methods?

- Reliability of paramount importance in software engineering and testing.
- Bayesian methods provide a probabilistic representation of *system uncertainty*.
- Elegant *incremental update* of probability in response to observations.
- Many possible applications: fault localization, adaptive testing, reliability analysis.
- *More details in the subsequent sessions!*

Why Bayesian Methods?

- Reliability of paramount importance in software engineering and testing.
- Bayesian methods provide a probabilistic representation of *system uncertainty*.
- Elegant *incremental update* of probability in response to observations.
- Many possible applications: fault localization, adaptive testing, reliability analysis.
- *More details in the subsequent sessions!*

Why Bayesian Methods?

- Reliability of paramount importance in software engineering and testing.
- Bayesian methods provide a probabilistic representation of *system uncertainty*.
- Elegant *incremental update* of probability in response to observations.
- Many possible applications: fault localization, adaptive testing, reliability analysis.
- *More details in the subsequent sessions!*

Why Bayesian Methods?

- Reliability of paramount importance in software engineering and testing.
- Bayesian methods provide a probabilistic representation of *system uncertainty*.
- Elegant *incremental update* of probability in response to observations.
- Many possible applications: fault localization, adaptive testing, reliability analysis.
- *More details in the subsequent sessions!*

Why Bayesian Methods?

- Reliability of paramount importance in software engineering and testing.
- Bayesian methods provide a probabilistic representation of *system uncertainty*.
- Elegant *incremental update* of probability in response to observations.
- Many possible applications: fault localization, adaptive testing, reliability analysis.
- *More details in the subsequent sessions!*

Session 1: Hypothesis Testing

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Session 1: Basic Probability

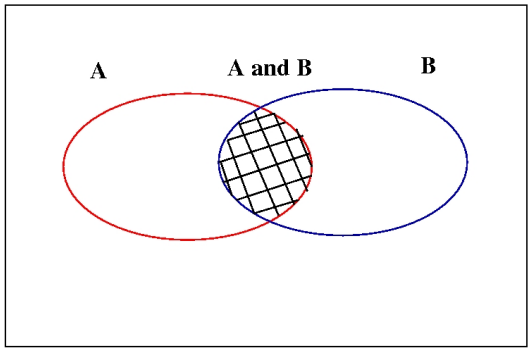
- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Basic Probability

- Explicit representation of uncertainty.
- The probability that event A occurs: $p(A)$.
- Basic axioms:

$$0 \leq p(A) \leq 1; \quad p(\text{true}) = 1; \quad p(\text{false}) = 0 \qquad (1)$$
$$p(A \vee B) = p(A) + p(B) - p(A \wedge B)$$

- Axioms can prove other relations: $p(\neg A) = 1 - p(A)$



Discrete Random Variables

- X is a R.V. i.e. a *random variable*.
- X can take a discrete number of values from the set: $\{x_1, x_2, ..., x_n\}$.
- $p(X = x_i) = p(x_i)$ is the *probability* that R.V. X takes the specific value x_i.
- $p(.)$ is the *probability mass function*.
- $p(\text{robot location}) = \langle 0.3, 0.6, 0.05, 0.05 \rangle$, over the available options $\langle \text{room}_1, \text{room}_2, \text{room}_3, \text{room}_4 \rangle$.

Discrete Random Variables

- X is a R.V. i.e. a *random variable*.
- X can take a discrete number of values from the set: $\{x_1, x_2, ..., x_n\}$.
- $p(X = x_i) = p(x_i)$ is the *probability* that R.V. X takes the specific value x_i.
- $p(.)$ is the *probability mass function*.
- $p(\text{robot location}) = \langle 0.3, 0.6, 0.05, 0.05 \rangle$, over the available options $\langle \text{room}_1, \text{room}_2, \text{room}_3, \text{room}_4 \rangle$.

Discrete Random Variables

- X is a R.V. i.e. a *random variable*.
- X can take a discrete number of values from the set: $\{x_1, x_2, ..., x_n\}$.
- $p(X = x_i) = p(x_i)$ is the *probability* that R.V. X takes the specific value x_i.
- $p(.)$ is the *probability mass function*.
- $p(\text{robot location}) = \langle 0.3, 0.6, 0.05, 0.05 \rangle$, over the available options $\langle \text{room}_1, \text{room}_2, \text{room}_3, \text{room}_4 \rangle$.

Discrete Random Variables

- X is a R.V. i.e. a *random variable*.
- X can take a discrete number of values from the set: $\{x_1, x_2, ..., x_n\}$.
- $p(X = x_i) = p(x_i)$ is the *probability* that R.V. X takes the specific value x_i.
- $p(.)$ is the *probability mass function*.
- $p(\text{robot location}) = \langle 0.3, 0.6, 0.05, 0.05 \rangle$, over the available options $\langle \text{room}_1, \text{room}_2, \text{room}_3, \text{room}_4 \rangle$.

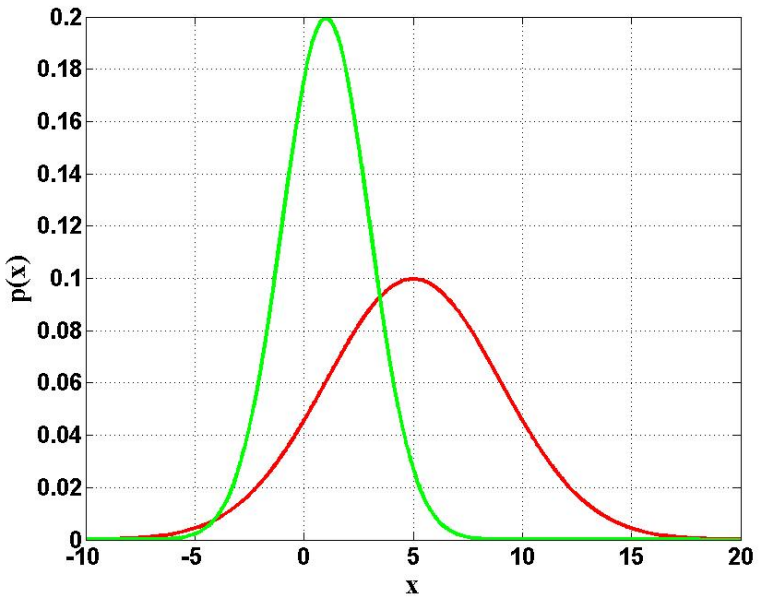
Discrete Random Variables

- X is a R.V. i.e. a *random variable*.
- X can take a discrete number of values from the set: $\{x_1, x_2, ..., x_n\}$.
- $p(X = x_i) = p(x_i)$ is the *probability* that R.V. X takes the specific value x_i.
- $p(.)$ is the *probability mass function*.
- $p(\text{robot location}) = \langle 0.3, 0.6, 0.05, 0.05 \rangle$, over the available options $\langle \text{room}_1, \text{room}_2, \text{room}_3, \text{room}_4 \rangle$.

Continuous Random Variables

- X can take values on a continuous range.
- $p(X = x) = p(x)$ is a *probability density function*.

$$p(x \in (a, b)) = \int_a^b p(x)dx$$



Joint and Conditional Probability

- $p(X = x, Y = y) = p(x, y)$.
- If X and Y are *independent* then:

$$p(x, y) = p(x) \cdot p(y)$$

- The probability of *x given y*:

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{2}$$

$$p(x, y) = p(x|y) \cdot p(y)$$

- If X and Y are *independent*: $p(x|y) = p(x)$

Marginals and Total Probability

- **Discrete RVs**:

$$\sum_x p(x) = 1 \tag{3}$$

$$p(x) = \sum_y p(x, y); \quad p(x) = \sum_y p(x|y)p(y)$$

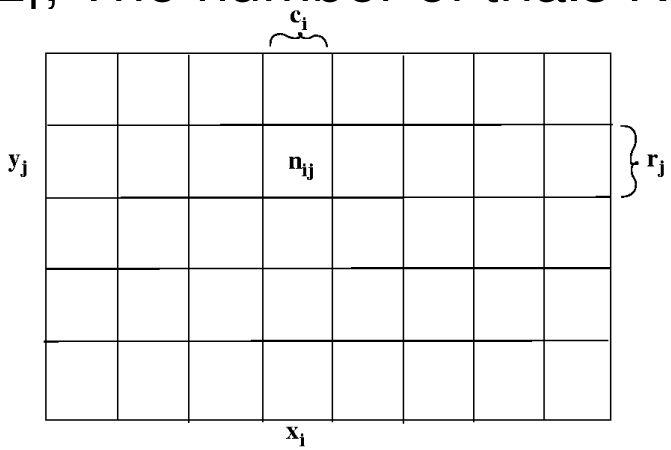
- **Continuous RVs**:

$$\int p(x)dx = 1 \tag{4}$$

$$p(x) = \int p(x, y)dy; \quad p(x) = \int p(x|y)p(y)dy$$

Illustrative Example

- $i \in [1, M], j \in [1, L]$; The number of trials $N = \sum_i \sum_j n_{ij}$.



$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}; \quad p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i} \tag{5}$$

$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j) = \frac{c_i}{N}$$

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i}\frac{c_i}{N} = p(Y = y_j | X = x_i)p(X = x_i)$$

Bayes' Rule

- Elegant way to *propagate belief*:

$$p(x, y) = p(x|y) \cdot p(y) = p(y|x) \cdot p(x) \tag{6}$$

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)} = \frac{\text{likelihood . prior}}{\text{normalizer}}$$

- Consider class labels $C_k, k \in [1, N]$ and data samples x:

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}; \quad p(x) = \sum_k p(x|C_k)p(C_k) \tag{7}$$

Bayes' Rule

- Elegant way to *propagate belief*:

$$p(x, y) = p(x|y) \cdot p(y) = p(y|x) \cdot p(x) \tag{6}$$

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)} = \frac{\text{likelihood . prior}}{\text{normalizer}}$$

- Consider class labels $C_k, k \in [1, N]$ and data samples x:

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}; \quad p(x) = \sum_k p(x|C_k)p(C_k) \tag{7}$$

Extensions

- Can include background knowledge:

$$p(x|y,z) = \frac{p(y|x,z)\,p(x|z)}{p(y|z)} \tag{8}$$

- Can generalize to *incremental* updates:

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x,z_1,\ldots,z_{n-1})p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})} \tag{9}$$

- Later: simplification based on *Markov* assumption!

Extensions

- Can include background knowledge:

$$p(x|y,z) = \frac{p(y|x,z)\,p(x|z)}{p(y|z)} \tag{8}$$

- Can generalize to *incremental* updates:

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x,z_1,\ldots,z_{n-1})p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})} \tag{9}$$

- Later: simplification based on *Markov* assumption!

Extensions

- Can include background knowledge:

$$p(x|y,z) = \frac{p(y|x,z)\,p(x|z)}{p(y|z)} \tag{8}$$

- Can generalize to *incremental* updates:

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x,z_1,\ldots,z_{n-1})p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})} \tag{9}$$

- Later: simplification based on *Markov* assumption!

Illustrative Example

- $C_1 : fault, C_2 : \neg fault, x : data.$
- $p(data|fault) = 0.6; \quad P(data|\neg fault) = 0.3$
- $p(fault) = p(\neg fault) = 0.5$

$$
\begin{aligned}
& p(fault|data) \qquad (10)\\
&= \frac{p(data|fault)p(fault)}{p(data|fault)p(fault) + p(data|\neg fault)p(\neg fault)}\\
&= \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = 0.67
\end{aligned}
$$

- Observing certain "data" incrementally increased the likelihood of a fault in the program.

Bayesian vs. Frequentist

- Classic example: coin-tossing.
- Fair-looking coin tossed thrice lands heads each time.
- Maximum likelihood estimate of probability of landing heads would be $= 1$. In other words, all future tosses would land heads!
- Bayesian estimate with a reasonable prior would provide a more conservative conclusion.

Bayesian vs. Frequentist

- Criticism: Bayesian approach sensitive to selection of prior distributions.
- Answer: non-informative priors.

- Poor choice of priors can give poor results with high confidence.
- Frequentist methods offer protection using techniques such as cross-validation.

Bayesian vs. Frequentist

- Criticism: Bayesian approach sensitive to selection of prior distributions.
- Answer: non-informative priors.

- Poor choice of priors can give poor results with high confidence.
- Frequentist methods offer protection using techniques such as cross-validation.

Summary

- Overview of hypothesis testing methods.
- Bayesian methods well-suited for adaptive program analysis.
- Introduced discrete and continuous random variables.
- Described joint and conditional probability densities.
- Bayes' rule and illustrative examples.

Summary

- Overview of hypothesis testing methods.
- Bayesian methods well-suited for adaptive program analysis.
- Introduced discrete and continuous random variables.
- Described joint and conditional probability densities.
- Bayes' rule and illustrative examples.

Summary

- Overview of hypothesis testing methods.
- Bayesian methods well-suited for adaptive program analysis.
- Introduced discrete and continuous random variables.
- Described joint and conditional probability densities.
- Bayes' rule and illustrative examples.

Summary

- Overview of hypothesis testing methods.
- Bayesian methods well-suited for adaptive program analysis.
- Introduced discrete and continuous random variables.
- Described joint and conditional probability densities.
- Bayes' rule and illustrative examples.

Summary

- Overview of hypothesis testing methods.
- Bayesian methods well-suited for adaptive program analysis.
- Introduced discrete and continuous random variables.
- Described joint and conditional probability densities.
- Bayes' rule and illustrative examples.

Further Reading


C. Bishop. *Pattern Recognition and Machine Learning*. Springer publishing house, 2007.


S. Thrun and W. Burgard and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.


G. Gigerenzer and S. Krauss and O. Vitouch. *The Null Ritual: What You Always Wanted to Know About Significance Testing but were Afraid to Ask*, The Sage Handbook of Quantitative Methodology for the Social Sciences (Editor: D. Kaplan), pp. 391–408. Sage Publications, 2004.


R. Duda and P. Hart and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.