

Qualifying Exam Preparation IV

Computational Statistics

Xi Tan (tan19@purdue.edu)

July 24, 2013

Contents

I	STAT598G Topics	4
1	Basic Computer Knowledge	5
1.1	Computer Organization, File Systems, Process Management, I/O Devices, Basic Linux Commands	5
1.2	Programming Languages, Interpreters and Compilers	5
2	C & R Programming Languages	6
2.1	Basic Variable Types and Scope	6
2.2	Control Flows: if-else, for/while Loops	6
2.3	Functions	6
2.4	I/O Access	6
2.5	Pointers and Dynamic Memory allocation in C	6
2.6	Underflow and Overflow	6
2.7	Vector, Matrix and Distributions in R	6
2.8	Graphics in R; Calling C from R	6
3	Algorithm and Data Structures	7
3.1	Big-O, Small-o, Theta Notations; Time Complexity Analysis . .	7
3.2	Recursion and Divide & Conquer	7
3.3	Dynamic Programming	7
3.4	Searching (Linear, Binary) and Sorting (Insertion Sort, Bubble Sort, Merge Sort)	7
3.5	Arrays, Linked Lists, Trees, Hash Tables and Related Operations	7
4	Statistical applications	8
4.1	Nonlinear Optimization : Gradient Descent, Coordinate Descent, Newton's Method, Line Search	8
4.2	Sampling: Generating Uniform Random Numbers, Transformation Method, Importance Sampling, Acceptance Sampling	8
4.3	Hidden Markov Models: Basic Definitions, Viterby Algorithm and Computing Forward and Backward Probabilities	8
4.4	Markov Random Fields, the Hammersley-Clifford Theorem, the Elimination Algorithm	8
4.5	Bootstrap	8

II	STAT598D Topics	9
5	Optimization	10
5.1	Unconstrained Optimization	10
5.1.1	Univariate Problems (Bisection, Newton)	10
5.2	Quasi-Newton Methods	11
5.3	Simulated annealing	11
6	The EM-type Algorithms	12
6.1	The EM algorithm	12
6.2	The ECM and ECME algorithms	12
6.3	The PX-EM algorithm	12
6.4	Computing observed Fisher information matrix	12
7	Integration	13
7.1	Numerical integration	13
7.2	Random number generation	13
7.3	Simulation	13
7.4	Monte Carlo integration	13
8	Markov chain Monte Carlo (MCMC) methods	14
8.1	The Data Augmentation algorithm	14
8.2	The Gibbs sampler	14
8.3	The Metropolis-Hastings algorithm	14
8.4	Data Fusion and Particle Filter (Sequential MCMC)	14
8.5	Reversible jump MCMC	14
8.6	Convergence Diagnostics	14

Preface

Textbook: Computational Statistics by G. H. Givens and J. A. Hoeting (GH)
TBD

Part I

STAT598G Topics

Chapter 1

Basic Computer Knowledge

- 1.1 Computer Organization, File Systems, Process Management, I/O Devices, Basic Linux Commands
- 1.2 Programming Languages, Interpreters and Compilers

Chapter 2

C & R Programming Languages

- 2.1 Basic Variable Types and Scope
- 2.2 Control Flows: if-else, for/while Loops
- 2.3 Functions
- 2.4 I/O Access
- 2.5 Pointers and Dynamic Memory allocation in C
- 2.6 Underflow and Overflow
- 2.7 Vector, Matrix and Distributions in R
- 2.8 Graphics in R; Calling C from R

Chapter 3

Algorithm and Data Structures

- 3.1 Big-O, Small-o, Theta Notations; Time Complexity Analysis
- 3.2 Recursion and Divide & Conquer
- 3.3 Dynamic Programming
- 3.4 Searching (Linear, Binary) and Sorting (Insertion Sort, Bubble Sort, Merge Sort)
- 3.5 Arrays, Linked Lists, Trees, Hash Tables and Related Operations

Chapter 4

Statistical applications

- 4.1 Nonlinear Optimization : Gradient Descent, Coordinate Descent, Newton's Method, Line Search
- 4.2 Sampling: Generating Uniform Random Numbers, Transformation Method, Importance Sampling, Acceptance Sampling
- 4.3 Hidden Markov Models: Basic Definitions, Viterby Algorithm and Computing Forward and Backward Probabilities
- 4.4 Markov Random Fields, the Hammersley-Clifford Theorem, the Elimination Algorithm
- 4.5 Bootstrap

Part II

STAT598D Topics

Chapter 5

Optimization

5.1 Unconstrained Optimization

5.1.1 Univariate Problems (Bisection, Newton)

Bisection Method

Suppose g' is continuous on $[a_0, b_0]$ and $g'(a_0)g'(b_0) \leq 0$, then the Intermediate Value Theorem implies that there exists at least one x^* for which $g'(x^*) = 0$ and hence x^* is a local optimum of g . To find this local optimum, the Bisection Method systematically halves the interval at each iteration, by checking the product of g' .

The updating equations are

$$[a_{t+1}, b_{t+1}] = \begin{cases} [a_t, x^{(t)}], & \text{if } g'(a_t)g'(x^{(t)}) \leq 0 \\ [x^{(t)}, b_t], & \text{if } g'(a_t)g'(x^{(t)}) > 0 \end{cases} \quad (5.1)$$

and $x^{t+1} = \frac{a_{t+1} + b_{t+1}}{2}$.

Newton's Method

Suppose g twice differentiable. At iteration t , Newton's method approximates $g'(x^*)$ by the linear Taylor series expansion:

$$g'(x^*) = g'(x^{(t)}) + (x^* - x^{(t)})(g''(x^{(t)})) \quad (5.2)$$

which gives us

$$x^* = x - \frac{g'(x^{(t)})}{g''(x^{(t)})} \quad (5.3)$$

5.2 Quasi-Newton Methods

5.3 Simulated annealing

Chapter 6

The EM-type Algorithms

- 6.1 The EM algorithm
- 6.2 The ECM and ECME algorithms
- 6.3 The PX-EM algorithm
- 6.4 Computing observed Fisher information matrix

Chapter 7

Integration

7.1 Numerical integration

7.2 Random number generation

7.3 Simulation

7.4 Monte Carlo integration

Chapter 8

Markov chain Monte Carlo (MCMC) methods

- 8.1 The Data Augmentation algorithm
- 8.2 The Gibbs sampler
- 8.3 The Metropolis-Hastings algorithm
- 8.4 Data Fusion and Particle Filter (Sequential MCMC)
- 8.5 Reversible jump MCMC
- 8.6 Convergence Diagnostics