STAT3011 Project 2 report You Xinyu 1155110904

1. **Root Finding**

This method basically is to find the root of a function ie. g(x)=0.

*Bisection method*

Bisection method is a root-finding method, which applies to any continuous

function to find all the zero points of the function(g(x0=0)). The method is about bisecting interval defined repeatedly, then selecting the subinterval which consist sign changes, therefore a root must be included. The steps of the method is like:

First initialize a (0) and b (0) such that f(a (0)) · f(b (0)) < 0.

Then we calculate the middle point c (0) of the interval [a (0), b(0)].

If f(c (0)) equals 0, one zero point was found and the calculation procedure is ended;

If f(c (0)) · f(a (0)) < 0, then the interval [a (0), c(0)] must contain a zero point. Therefore, in this case, let a (1) be a (0) and let b (1) be c (0);

If f(c (0)) · f(b (0)) < 0, let a (1) be c (0) and let b (1) be b (0).

Finally, In the previous cases, a new interval [a (1), b(1)] ⊂ [a (0), b(0)] was obtained, then we repeat the first step in the procedure until b (t) − a (t) is less than a specified tolerance. The result would be c(t)=

Bisection method is a very simple and quick method since the method discards half of the current interval at each repeated phase, so it brackets the root more quickly than the incremental search method. However, it only finds roots cross the x axis but not roots are tangent to the x axis. Also, the singularities in the function can fool the method as well as it cannot find complex roots in a polynomial. The performance of bisection method has slower computational time compare with Newton’s method but with high accuracy and no errors and guaranteed a root will be found.

*Newton-Raphson method*

The Newton-Raphson method (Newton's method) is also a method find the

roots of a function. But unlike bisection method, it finds a good approximation for the root. It uses a straight-line tangent to a continuous and differentiable function.

This method works on find a good approximation for the root:

Then the previous step might be repeated many times if necessarily to obtain the desired accuracy. For any x-values xn the next value is given by

Unlike bisection methods, Newton’s method can identify with repeated roots, since it doesn’t involve changes in the sign of the function explicitly. Also, it can find complex roots of polynomial as well as it has higher converges rate than bisection method. However, Newtons method is not guaranteed to find a root and sometimes the function may be difficult to differentiate. The performance of Newton’s method although have very fast convergence rate, but the accuracy may not be optimistic sometimes if there are either points of inflection or local maxima/minima around x0​ or the root.

For statistic inference, MLEs, MAP, minimizing loss functions or finding confidence interval can all be reduced to a root finding problem.

1. **Optimization- Genetic algorithm**

The genetic algorithm is a random-based classical evolutionary algorithm, which a set of random solutions was generated, in order to find a solution, random slight changes applied to the current solutions to generate new one. This algorithm reflects the process of natural selection where the fittest individuals are selected to produce offspring of the next generation.

The algorithm basically consists of five phases. Which are:

*Initial population(initialization)*

The process starts with a set of individuals (solutions) which called ***Population***. Then an individual is characterized by a set of **Genes** (parameters or variables). Then Genes joined into a string to form a ***Chromosome*** (solution). In genetic algorithm, the set of genes of an individual is represented by a string, usually binary values are used to encode the genes in a chromosome.

*Fitness function (Ranking)*

The function determines the ability of an individual to compete with others then gives a score to each individual. Based on the score, the probability of an individual will be selected for producing offspring is affected accordingly

*Selection*

The **selection** phase is to select the fittest (highest rank) individuals and let them pass their genes to the next generation. Two pairs of individuals (**parents**) are selected according to their fitness score (the higher score, the larger probability that an individual might be selected for reproduction).

*Crossover*

Crossover in the algorithm generates new generation which is similar to natural mutation: mutating old parents, new generation offspring generated with genes from both parents randomly. Sometimes the offspring carries even amounts of genes form their parents, but sometimes the percentage will change. For each two parents, crossover takes place by selecting a random point in the chromosome then exchanging genes before and after that selected point from its parents. The result is the chromosomes of their offspring.

*Mutation*

The next phase is mutation. In each offspring, some genes are selected and changed. Mutation varies based on representing the chromosomes but we can decide how to apply mutation. For the binary encoding, we can flip the bit value of one or more genes.but for more than two values binary mutation is not applicable. Without mutation, offspring might be identical to their parents.

*Termination*

The algorithm ends when the population has converged which do not produce offspring that has significantly difference with the previous generation. Then the genetic algorithm has outputed a set of solutions for the problem.

There are variety of advantages of genetic algorithm for example, it support multi-object optimization, the concept is easy to understand, it is easily parallelized and etc. However, this algorithm requires less information about the problem, but designing a objective function and operators right can be difficult. The performance of genetic algorithm may not be very efficient since it is very time consuming which we need to apply mutation every single time, and the result is objective although is gives a set of solution, but they are just for reference, no exact result was found.

For statistic inference, in non-linear constrain function, genetic algorithm helps to optimize the regression model as well as gives an objective function to the problem, which helps in interpretation from the set of solution.

1. **Numerical integration- Quadrature**

Numerical integration is a way to approximate the integral of a function over a given domain. The numerical quadrature (quadrature) is a synonym to numerical integration, especially applied to one-dimension integrals. It can generally describe as find an approximate solution to a definite integral. The integrand is evaluated at a finite set of integration points and the weighted sum of the values is used for the integral approximation. The integration points and weights depend on the specific method used and the accuracy required from the approximation.

Quadrature method is a numerical method that approximate the area under the curve (in univariate case by dividing area under curve into little pieces.)

*Mid-point rule (rectangle rule)*

The simplest method to let the interpolating function be a constant function which is a polynomial of zero degree that passes through the point () which

The generalized formula is

*Trapezoidal rule*

When the interpolating function might be straight line like a polynomial with 1 degree, passing through the points (a, f(a)) {\displaystyle \left(a,f(a)\right)}and (b,f(b)) {\displaystyle \left(b,f(b)\right)}.

Which

*Composite rule*

For each of the above rule, we can make accurate approximation by breaking uo the intervals to n sub intervals then calculating the subinterval and add up them as the final result. For example,

The advantage of quadrature is the functional data at two end points are not used but, the data points are not equispaced. For trapezoidal rule, it is simple and optimal for improper integrals. However, large amount of subintervals is needed to obtained good accuracy. The performance of the method is depended on how many subintervals was divided, the more the subinterval the higher the accuracy, but longer calculating time.

For statistic inference, it helps study the behavior of approximation error, where a method which yields a small error for a small number of evaluations is u considered as superior. Reducing

the number of evaluations of the integrand reduces the number of

arithmetic operations involved, and therefore reduces the total round-off

error.

1. **Monte Carlo integration**

Monte Carlo methods is a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The concept of the algorithm is to use randomness to solve problems that may be deterministic in principle. Monte Carlo methods are usually used in optimization, numerical integration, and generating draws from a probability distribution. With law of large number, when the probability function is parametrized we use Markov Chain Monte Carlo(MCMC) sampler.

MCMC is a method for simulating f is any method producing an irreducible, aperiodic and positive recurrent Markov chain whose stationary distribution is f. Also, MCMC provides us with ways to sample from any probability distribution. This is mostly needed when we want to sample from a posterior distribution.

*Metropolis-Hasting (MH) algorithm*

It is a MCMC method provides us with a general approach to constructing

Markov chain {X(t)} with f(x) being the limiting distribution.

For the algorithm, we first input: the pdf (or pmf) f, a starting point x (0) and

proposal distribution q(y|x).

Then initialize: t=0.

Repeatedly (t increase by 1 each time) generate yt ∼ q(·|x (t) );

Calculate ρ(x (t) , yt) = min{ ,1} ;

Accept yt as x (t+1) with probability ρ(x (t) , yt); otherwise, reject yt and let x (t+1) be x (t) Until some criteria are met end the algorithm. The output will be: Given a large number B, {x (B) , x(B+1) , …} are samples from the distribution f.

*Gibbs sampler*

When f(x) has a large number of variates, the transition distribution in the MH algorithm will be high dimensional. A state space with a high dimensionality and a poor transition distribution often lead to very slow convergence rate of Markov chain to its limiting distribution. Gibbs sampler can get around the problem by iteratively sampling from full conditional functions. For the algorithm,

First input the pdf (or pmf) f(x) and a starting point x (0) = (x 1(0) , … , x p(0) ).

Then initialize t= 0.

Repeatedly (t increase 1 each time) generate x1(t+1) ∼ f1(x1|x2(t), x3(t) , …, xp(t) );

then generate x2(t+1) ∼ f2(x2|x1(t), x3(t) , …, xp(t) );

similarly generate until xp(t+1) ∼ fp(xp|x1(t), x2(t) , …, xp-1(t) );

Until some criteria are met the algorithm ended. And the output will be: given a large number B, {x (B) , x (B+1) , . . .} are samples from the distribution f.

MCMC method is useful in Bayesian inference, and Gibbs sampling in combination with data augmentation allows inference in statistical models with many unobserved variables. The likelihood functions of these models may contain many integrals, usually makes a standard classical analysis difficult or even unfeasible. One of the advantage of MCMC is that in Bayesian approach, we only need to consider the likelihood function on the unobserved variables and in many cases it implies that Bayesian parameter estimation is faster than classical MLE.