**TBPS Week 1 - Acceptance Function Research**

What is an acceptance function?

The best example is the Metropolis–Hastings algorithm. It is a Markov chain Monte Carlo method which uses algorithms to obtain a sequence of random samples from a probability distribution. This sequence can then be used to approximate the distribution or integral of the distribution.

The *acceptance probability* ,a(X → X′), is the probability of accepting a proposed mutation from X to X′ to ensure that the distribution of samples is proportional to the function.

Diagram

Description automatically generated

T (X → X′): tentative transition function that gives the probability density of the mutation

If T is the same in both directions, simplifies to:

Text

Description automatically generated with low confidence

How to use simulation data to extract an acceptance function?

**Method**

1. Metropolis algorithm generates a sequence of samples Xi from a function
2. From the first sample X0, each following sample Xi is generated using a random mutation applied to Xi-1 to compute the proposed sample X’
3. This new value of X’ is either accepted or rejected based on the given constraints
4. X’ is then either saved as Xi or Xi takes on the value of Xi-1
5. This is then repeated for all the samples.

**Essentially to calculate Xi**

Xi -1🡪X’ via mutation/transformation

X’🡪 Xi if X’ accepted

Xi-1 🡪 Xi if X’ rejected

Repeated for 0🡪i

Need to figure out constraints (i.e. high/low bounds) which can be imposed on the samples.

<https://stephens999.github.io/fiveMinuteStats/MH_intro.html> has a walk through example of implementation in python which could be useful.

Sources

*About the algorithm:*

<https://www.sciencedirect.com/topics/computer-science/acceptance-probability>

<https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm>

*The algorithm in code:*

<https://stephens999.github.io/fiveMinuteStats/MH_intro.html>

<https://www.sciencedirect.com/topics/computer-science/metropolis-algorithm>

Text

Description automatically generated