Indian Institute of Technology, Indore

Department of Astronomy, Astrophysics and Space
Engineering (DAASE)
Astrostatistic Mini Project

MCMC Algorithm for Parameter Estimation

Prepared by: Sanjay Kumar Yadav
Roll No: 2203121008
Instructor: Dr. Suman Majumdar

Aim

The goal of MCMC is to draw samples from some probability distribution without having to know its exact height at any point..

1 Introduction

In this exercise, we explored the use of Markov chain Monte Carlo (MCMC) algorithms for parameter estimation. MCMC algorithms are widely used in statistics to generate samples from the posterior distribution of the parameters given some observed data. It is commonly used in Bayesian statistics to estimate the posterior distribution of a parameter or set of parameters. We will use the Metropolis-Hastings algorithm to estimate the parameters.

2 Basics of MCMC Algorithm

The MCMC is composed of two components - ***Monte Carlo and Markov Chain.*** MCMC is based on the concept of a **Markov chain.**

- **Markov chain** is a sequence of random variables. The Markov chain is defined by a set of states and a transition matrix that specifies the probability of transitioning from one state to another. Where the transition matrix is a square matrix in which each element represents the probability of transitioning from one state to another.

  In markov chain the key property is the Markov property, which states that the probability of transitioning to a new state depends only on the current state, and not on any previous states. This property allows us to predict the future behavior of the system based on its current state and the transition matrix. In other words, it is a mathematical model that describes the evolution of a system over time.

- **Monte carlo** is a technique for sampling from a probability distribution and using those samples to estimate desired quantity. In other words, it uses randomness to estimate some deterministic quantity of interest.
  MCMC uses Monte Carlo methods to simulate the Markov chain.

  Monte Carlo algorithms typically involve the following steps:-

  – First of all identify problem and some relevant parameters.
  – According to given instruction in problem, Use a probability distribution to generate a large number of random samples.
  – Evaluate the samples to estimate the parameter of interest. This process can involve calculating the mean, variance, or other summary statistics of the samples.
  – Repeat above steps for large number of times to improve the accuracy of the estimate.

- **Examples of MCMC algorithms:-**
  – Metropolis-Hastings algorithm (MH)
  – Gibbs sampling
  – Hamiltonian Monte Carlo (HMC)
  – Slice sampling
  – Reversible jump MCMC etc.

In our problem we will use MH-algorithm which is following.

3 MH-Algorithm

The Metropolis-Hastings algorithm is a Markov Chain Monte Carlo (MCMC) algorithm used for generating a sequence of random samples from a probability distribution. Since target probability distribution may be difficult to sample directly due to high dimensionality, intractable likelihoods, complex distribution etc. To counter that the MH-algorithm is used. It generate a sequence of samples from an proposal distribution, which is a simpler distribution that is easy to sample, and then accepting or rejecting the proposed samples based on the target distribution.

Metropolis-Hasting algorithms typically involve the following steps:-

- **It starts with an initial state or sample.**
  let us consider we want to estimate parameter θ then first of all we have to choose an initial value of θ.
  let $\theta_i = \theta_0$ is my current state or parameter.

- **Generate a proposal state or sample from a proposal distribution.**
  let say our proposal state is θ_p.

- **Calculate the acceptance probability for the proposed sample using the ratio of the target distribution and the proposal distribution.**

$$\boxed{a = min\left(1, \frac{P(\theta_p)P(D/\theta_p)}{P(\theta_0)P(D/\theta_0)}\right)}$$

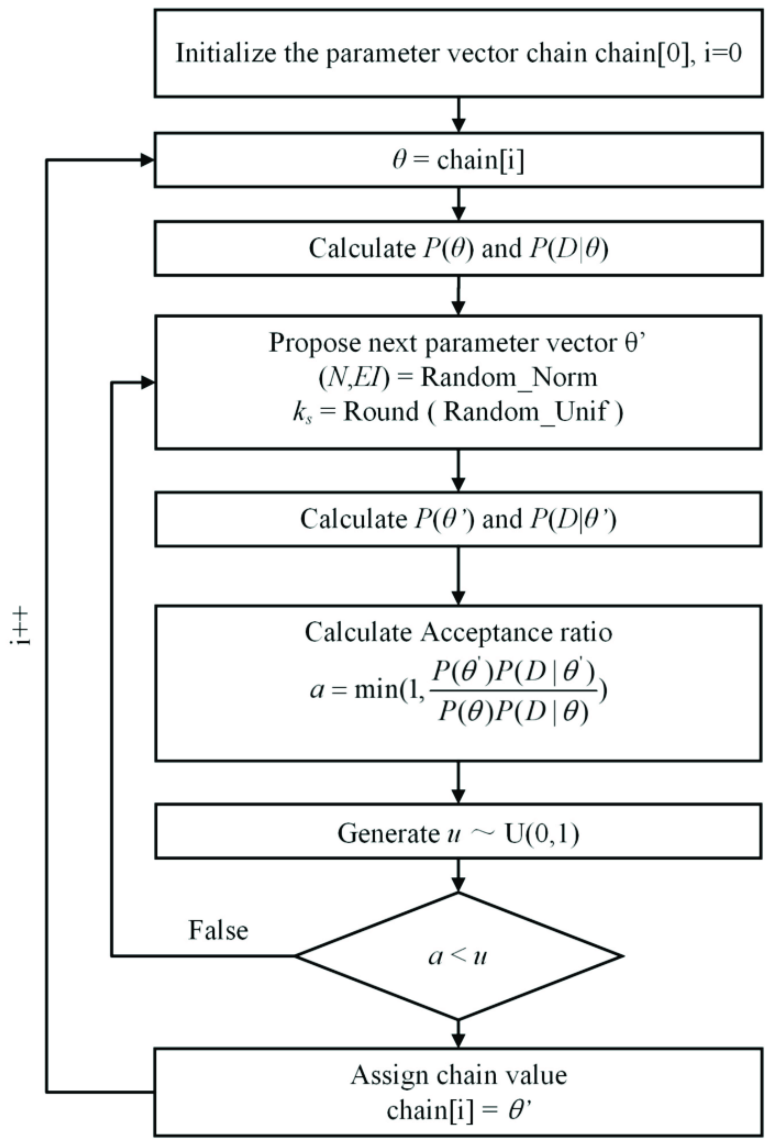
- **Generate a random number between 0 and 1.**
  let δ is random number which is $\boxed{0< \delta <1}$

- **If the random number is less than the acceptance probability, then accept the proposed sample and set it as the new state. Otherwise, reject the proposed sample and keep the current state.**

- if $\delta <$ acceptance probability(a) $\implies \boxed{\theta_{i+1} = \theta_p} \implies$ **New state**
  $Otherwise \implies \boxed{\theta_{i+1} = \theta_0} \implies$ **Currrent state**

Repeat steps 2^{nd} to 5^{th} for a specified number of times or until convergence is reached.

The following image represents all these steps in more easy way.



MH-algorithm's steps

4 Problem - Supernova Ia

The problem we are trying to deal with is the **parameter estimation of Ω_m and Hubble Constant H_0.** where $H_0 = 100hkms^{-1}Mpc^{-1}$,we are trying to estimate h and Ω_m. We see that both, Ω_m and h are dimensionless quantities
and 0 < (Ω_m, h)< 1.

Model

- We are given that distance modulus(μ) is

$$\mu = 25 - 5\log_{10} h + 5\log_{10}\left(\frac{D_L^*}{Mpc}\right)$$

and

$$D_L(z) = \frac{c}{H_0}(1+z)\left[\eta(1,\Omega_m) - \eta\left(\frac{1}{1+z},\Omega_m\right)\right]$$

where D_L is luminosity distance and D_L^* is luminosity distance calculated at h = 1.
so from above equation we can seen that μ is indirectly function of h and Ω_m. That's why we will estimate both of these parameters.

- **To estimate these parameters we will use MH-algorithm and for this we are given likelyhood.**

$$\mathcal{L} \propto \exp\left[-\frac{1}{2}\sum_{i,j=1}^{n}[\mu_i - \mu_{th}(z_i)]C_{ij}^{-1}[\mu_j - \mu_{th}(z_j)]\right]$$

Here z is Redshift, μ_{th} is theoritical value of distance modulus, μ_i is observed value of distance modulus and C^{-1} is Inverse of Covariance matrix.

- The priors on h and Ω_m can be assumed to have a uniform distribution. So the posterior distribution is only proportional to the likelihood $\mathcal{L}$.

- Given that proposal distribution is a Gaussian Distribution.

5 Results & Discussion

First of all we have to eye estimate that for which value of parameters, theoretical distance modulus(μ_{th}) matched with observed distance modulus(μ) .

5.1 Eye estimation of Parameters

For this we plot observed data vs theoretical data with different sets of parameters (Ω_m,h).

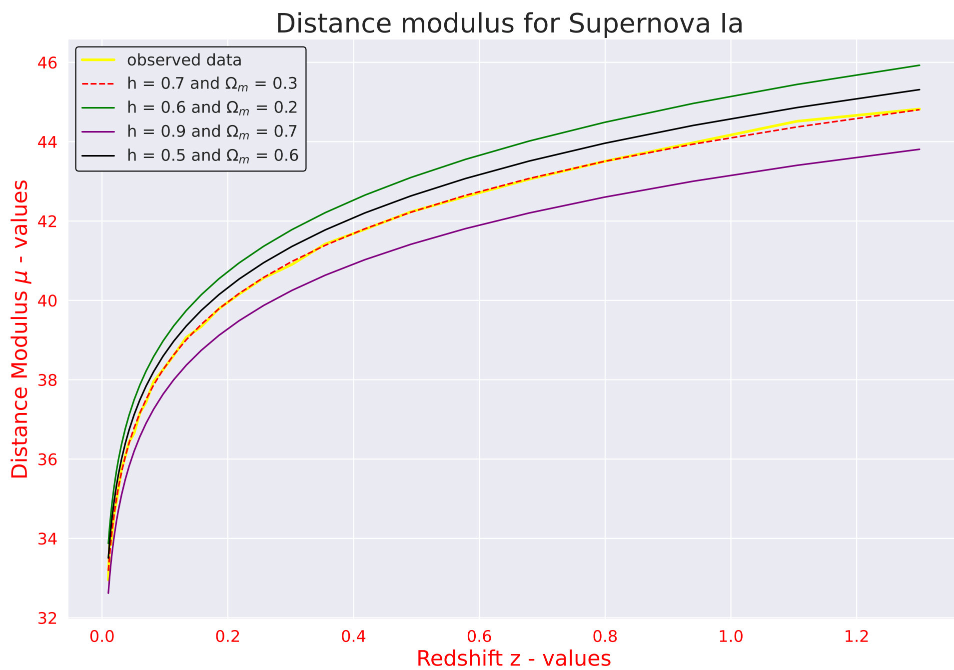


Figure 1: **Eye estimated parameter**

We can see that $h \approx 0.7$ and $\Omega_m \approx 0.3$.

5.2 Parameter estimation by MH-algorithm

After using MH-algorithm, we got values of Ω_m and h for different initial states of parameters.

Histogram plots:-

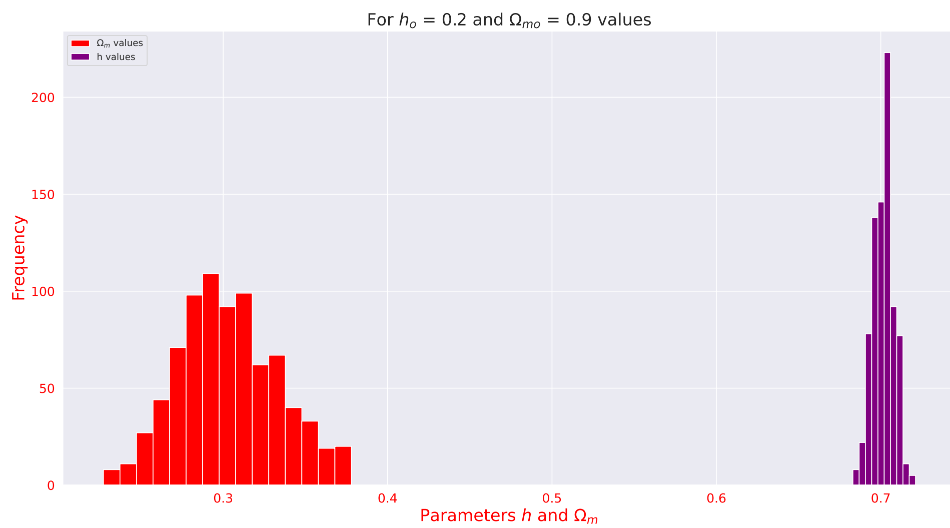
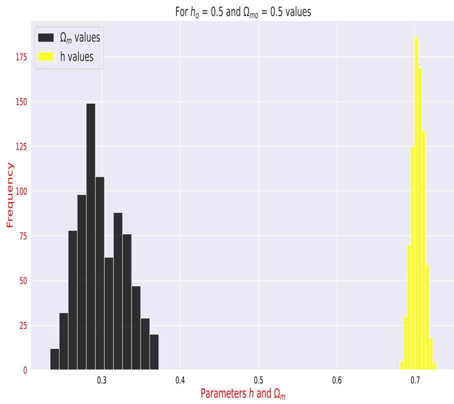
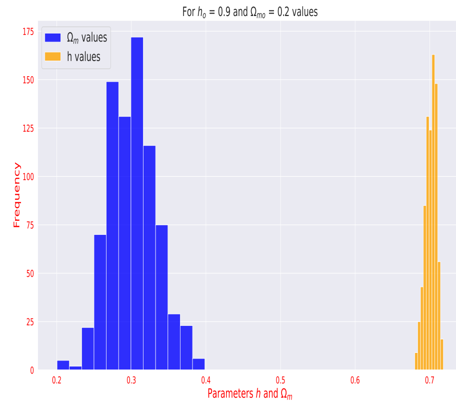


Figure 2: **Histogram for Ω_{m0} = 0.9 and h_0 = 0.2**

In above histogram we can see that value of Ω_m and h are converge in range of eye estimated values. This plot is get after burn in some starting values.
When we apply MCMC algorithm for this problem, at the end we got that values of Ω_m and h are converge in range of eye estimated values only even for different initial values it will converge. Which we can see in following plots as well. I plot all three plots in one histogram also, in this also we can seen same things.



(a) for $\Omega_{m0} = 0.5$ and $h_0 = 0.5$



(b) for $\Omega_{m0} = 0.2$ and $h_0 = 0.9$

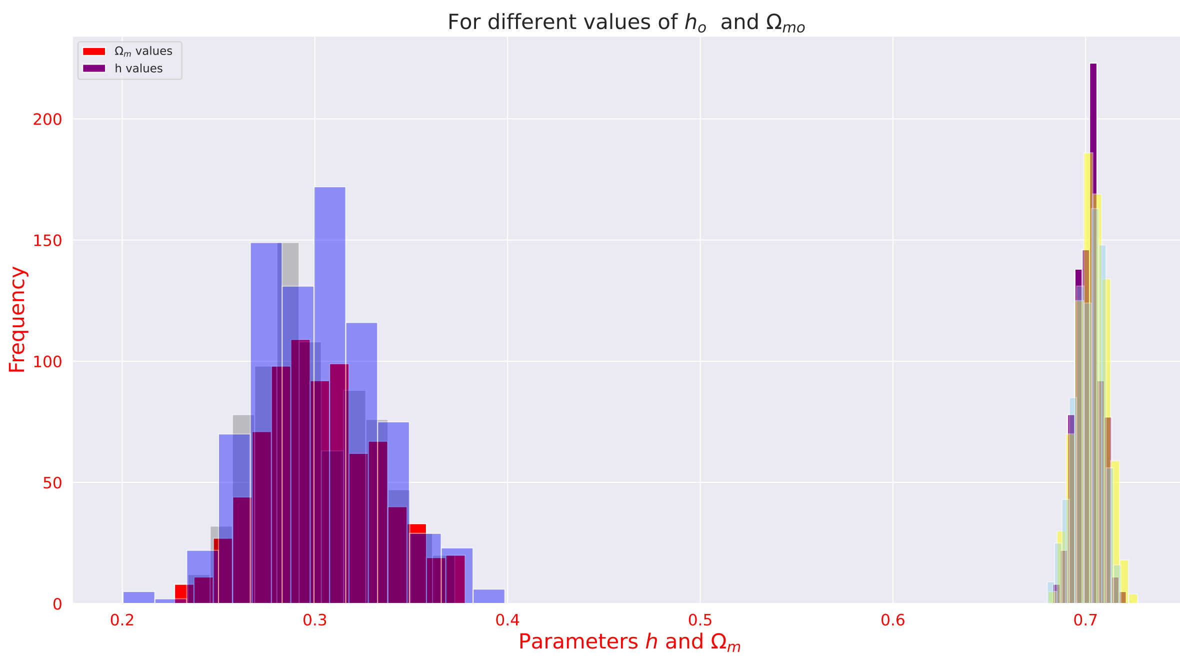


Figure 4: **Histogram for different values of Ω_{m0} and h**

Scatter Plots:-

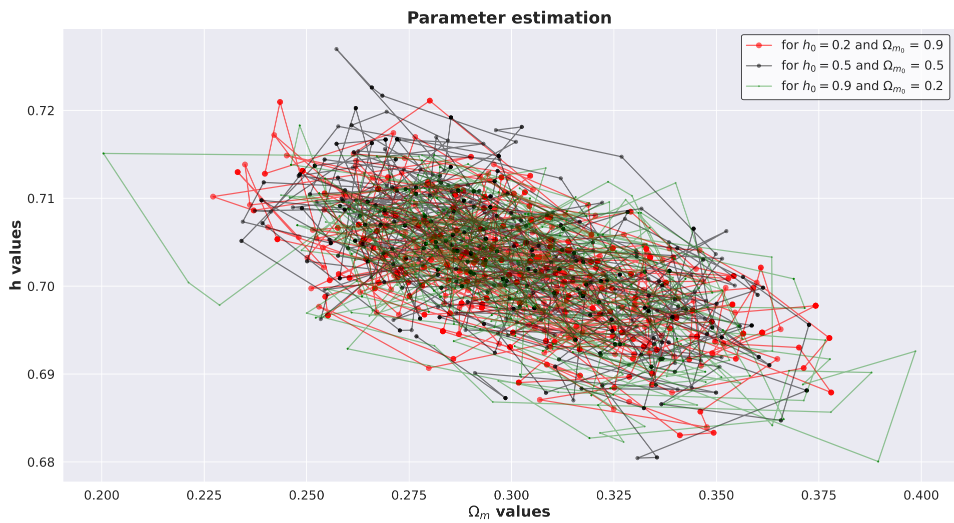
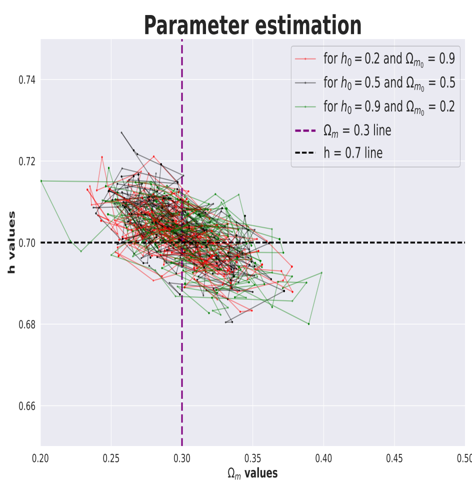
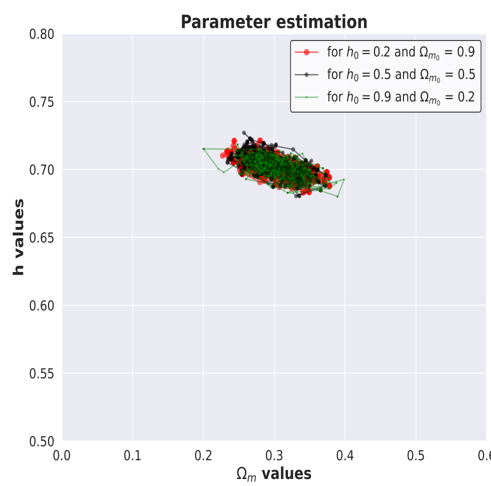


Figure 5: **Scatter plot of Ω_{m0} and h (Zoomed In)**



(a) Zoom medium



(b) Zoomed Out

6 Acceptance Probability with size of Proposal Distribution

To check variation in Acceptance probability with size of Proposal Distribution, We can check values of estimated parameters at different-different standard deviation(σ).

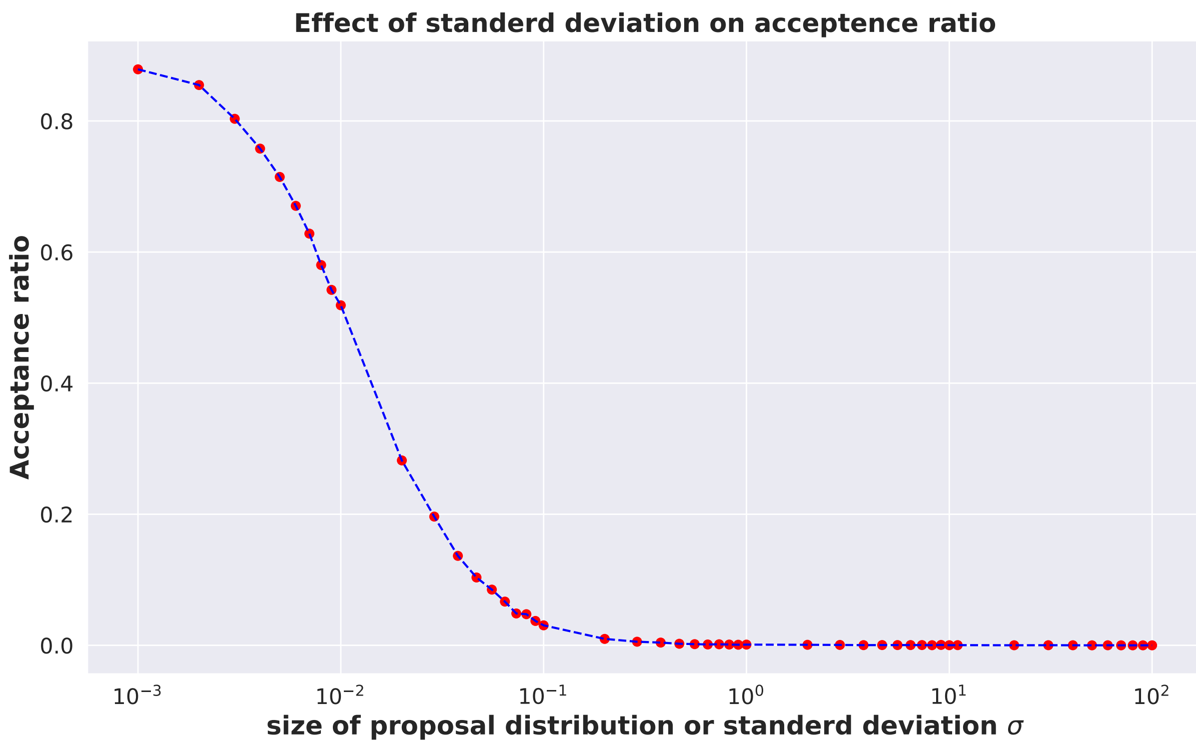


Figure 7: **Acceptance ratio vs** σ

The random walker with smaller σ never reaching the correctly estimated value previously i.e h ≈ 0.7 and $\Omega_m \approx 0.3$. On the other hand, for larger σ, the sampler doesn't sample any point at all. This result can be interpreted in terms of the Acceptance Probability(A), which is the ratio of the no. of points accepted to the total no. of steps taken by the random walker.

7 Statistics

We have selected one of the random walkers with initial h and Ω_m being 0.1 and 0.9 respectively, and calculated the statistics. The 'reasonable' choice of σ, that we have taken is $\sigma = 0.073$. After applying a 'burn-in' removing first 20% of the total steps, we then calculate the statistics of the the final arrays of h and Ω as follows:-

- Mean:
  - Mean of h = 0.7027
  - Mean of $\Omega_m = 0.2989$
- Variance:
  - Variance of h = 4.68×10^{-05}
  - Variance of $\Omega_m = 8.66 \times 10^{-04}$
- Covariance:
  - covariance of h = 4.69×10^{-05}
  - covariance of $\Omega_m = 8.67 \times 10^{-04}$

Plot of Distance Modulus using estimated parameters

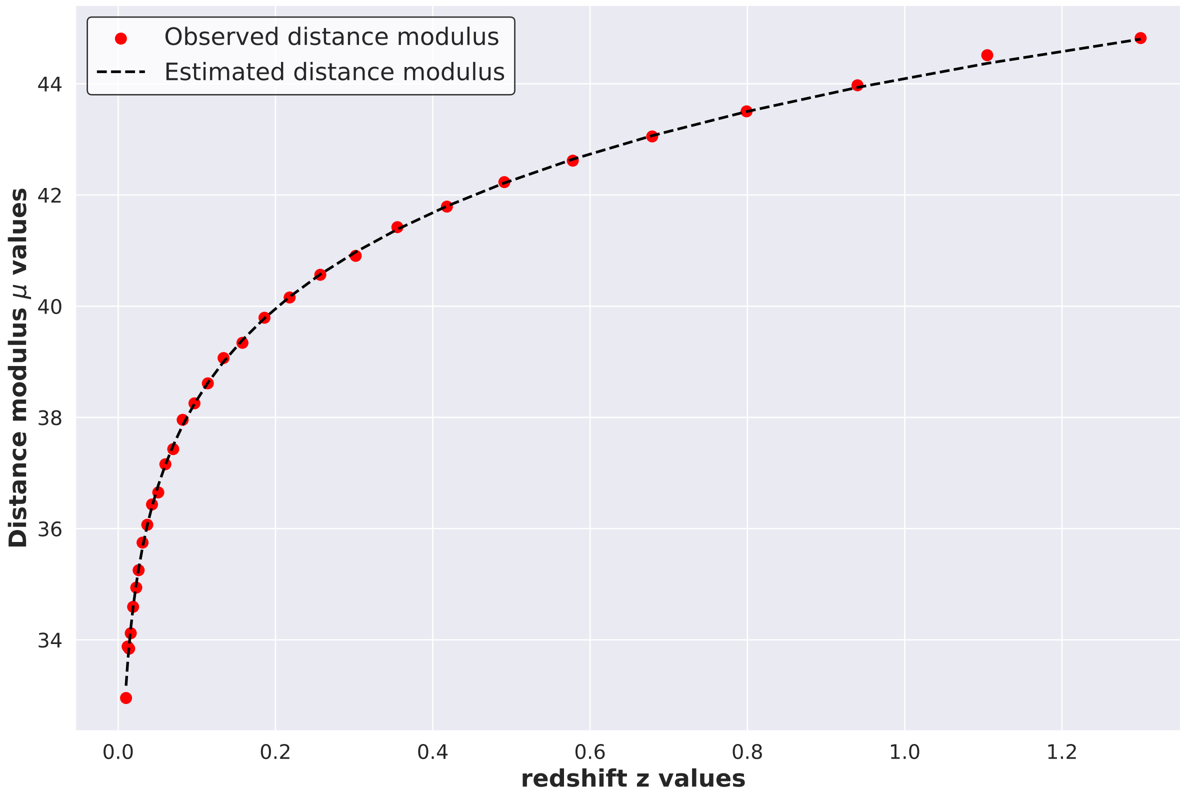


Figure 8: **Distance modulus for supernova Ia**