Estimating Motion Uncertainty in Vehicles

Aarya Gupta, Aditya Raj Jain, Aditya Kumar Singh
Networked Autonomous Systems Lab IIITD
Indraprastha Institute of Information Technology
Delhi, India
{aarya22006, aditya22037, aditya22598}@iiitd.ac.in

***Abstract*—Predictive capabilities and decision-making processes bind every Autonomous vehicle's safety.**
Probabilistic prediction varies depending on real-life factors. It can be divided into Aleatory(stochastic) and epistemic(non-stochastic).
This project aims to analyse various methods to improve the protective capabilities of an uncomplicated kinematic bicycle's steering angle prediction model.
We primarily compare and combine the KDE (Kernel Density Function) and Bootstrapping to predict the next angle while using Bayesian updation (Maximum A posteriori) to improve the model with the obtained data of previous iterations.
***Index Terms*—Autonomous vehicles, Kinematic bicycle model, Aleatory uncertainty Epistemic uncertainty, Kernel Density Estimation (KDE), Bootstrapping, Maximum A posteriori (MAP)**

I. Introduction

The system operates under the following assumptions:

1) **Discrete Time System**: The system is modelled as a discrete-time system.
2) **Uniform Time Increments**: Time increments are fixed, with $\Delta t = 1$.
3) **Underlying System Equations**: The bicycle model in our world follows the following equations:

$$\theta(t) = \theta(t - \Delta t) + \Delta\theta(t) \tag{1a}$$

$$x(t) = x(t - \Delta t) + v \cdot \cos(\theta(t)) \cdot \Delta t \tag{1b}$$

$$y(t) = y(t - \Delta t) + v \cdot \sin(\theta(t)) \cdot \Delta t \tag{1c}$$

Here,
   - θ is the R.V. describing the heading angle of the vehicle.
   - $\Delta\theta$ is the R.V. describing the change in heading angle.
   - x and y are the R.V.s depicting the position.
4) **Initial Conditions and Assumptions**:
   - Let the true underlying (unknown) distribution of $\Delta\theta(t)$ be $\hat{\mathcal{F}}$, which is same for all time instants.
   - Values of x_0, y_0, θ_0, v are definite/constants and are given.
   - For simplicity, let's assume x_0, y_0, θ_0 to be 0 while the vehicle is moving with a constant speed of 1.

$$\Rightarrow \theta_0 = 0, \quad \Delta\theta_0 = 0, \quad x_0 = 0, \quad y_0 = 0, \quad v = 1 \tag{2}$$

   - Assuming the initial set of values $\mathcal{F}_1$ are given, from which $\Delta\theta_1$ will chose a value. $\mathcal{F}_1$ represents the distribution of $\Delta\theta_1$.

	$t = 0$	$t = 1$	$t = 2$
Known RVs	$\theta_0 = 0$, $\Delta\theta_0 = 0$ $x_0 = 0$, $y_0 = 0$ $\mathcal{F}_1$ ($\Delta\theta_1$ distrib.)	$\Delta\theta_1$ (observed) θ_1, x_1, y_1	$\Delta\theta_2$ (observed) θ_2, x_2, y_2
Predicted RVs	-	$\mathcal{F}_2$ ($\Delta\theta_2$ distr.)	$\mathcal{F}_3$ ($\Delta\theta_3$ distr.)
True Distrib. (Unknown)	$\hat{\mathcal{F}}$	$\hat{\mathcal{F}}$	$\hat{\mathcal{F}}$

TABLE I
System Timeline

$\mathcal{F}_t$ is the predicted distribution at time $t-1$ from $\mathcal{F}_{t-1}$ $\forall$ $t \geq 2$. The true value of $\Delta\theta_t$ at t will be observed from the true unknown distribution $\hat{\mathcal{F}}$.

To estimate the underlying distribution of $\Delta\theta$, the following approached are followed:

A. Kernel Density Estimation (KDE)

Imagine you have a bunch of data points, and you want to understand their distribution. Kernel Density Estimation (KDE) is like a sophisticated way of creating a smooth curve that represents this distribution by placing a curve (Kernel) at all the points from the sample and adding them up to find the distribution (It is normalised such that probability adds up to 1). It's particularly useful when you don't know the underlying shape of your data's distribution.

B. Bootstrapping

Initially, given a set of real-world samples, we create bootstrap samples at each time step by randomly selecting elements from the original set, allowing replacement. The predicted distribution at each time step is the bootstrapped distribution of the original set. The true value of the change at each time step is observed from this predicted distribution. We calculate the Mean Squared Error (MSE) at each time step to understand the underlying distribution of changes over time. This MSE is measured between the predicted and initial real-world data distributions. We aim to minimize this MSE as time progresses, indicating that our predictions are becoming more accurate and approaching the unknown underlying distribution.

C. KDE combined with Bootstrapping

In this, we combine the benefits of KDE and bootstrapping. From the original samples, we bootstrap it to produce 500 samples. For each sample, we construct a KDE distribution. All the samples are assigned a different kernel bandwidth. Finally, all the distributions are combined to form a final distribution, with a bandwidth normalised from all the samples. It increases the computational complexity but provides a better estimate of the underlying distribution.

D. Bayesian Updation

This is a statistical method to update our prior distribution based on new observed data through the help of Bayes' theorem.
Over time, it has the capability to predict the underlying distribution accurately given certain constraints.

II. Introduction to Kernel Density Estimation (KDE)

A. Basic Idea

KDE places a small "bump" (called a kernel) at each data point and then adds up all these bumps to create a smooth curve. This curve estimates the probability density function (PDF) of your data.

B. Mathematical Formulation

The kernel density estimator is defined as:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{3}$$

Here, $\hat{f}_h(x)$ is our estimated density at point x, n is the number of data points, h is the bandwidth (which controls how wide our bumps are), K is the kernel function (the shape of our bumps), and X_i are our individual data points.

C. Kernel Functions

The kernel function K determines the shape of our bumps. It must integrate to 1 over its domain:

$$\int_{-\infty}^{\infty} K(u)du = 1 \tag{4}$$

Standard kernel functions include:
1. Gaussian (Normal) Kernel:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} \tag{5}$$

This is like using little bell curves as our bumps.
2. Epanechnikov Kernel:

$$K(u) = \frac{3}{4}(1 - u^2) \text{ if } |u| \leq 1, \text{ else } 0 \tag{6}$$

This uses parabolic bumps, which some consider optimal in a mathematical sense.

III. The Math Behind KDE

A. Bias-Variance Tradeoff

In statistics, we often deal with a tradeoff between bias (how far off our estimate is on average) and variance (how much our estimate varies with different samples). KDE is no exception. The mean integrated squared error (MISE) helps us quantify this tradeoff:

$$\text{MISE}(\hat{f}_h) = \int (\text{Bias}[\hat{f}_h(x)])^2 dx + \int \text{Var}[\hat{f}_h(x)]dx \tag{7}$$

Considering both bias and variance, this equation tells us how good our estimate is overall.

B. Bandwidth Selection

Choosing the proper bandwidth h is crucial. Too small, and our estimate will be too spiky; too large, and it will be too smooth, missing essential features.
The optimal bandwidth that minimizes the asymptotic MISE is:

$$h_{opt} = \left(\frac{R(K)}{n\mu_2(K)^2 R(f'')}\right)^{1/5} \tag{8}$$

where $R(K) = \int K^2(u)du$, $\mu_2(K) = \int u^2 K(u)du$, and $R(f'') = \int (f''(x))^2 dx$.
In practice, we often use data-driven methods like cross-validation to choose h.

C. Multivariate KDE

When dealing with multiple variables (like position and velocity), we use multivariate KDE:

$$\hat{f}_H(x) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - X_i) \tag{9}$$

Here, H is a matrix that determines the width and orientation of our multidimensional bumps.

IV. Applying KDE to Your Data

To use KDE on your incoming data:
1. Collect your data points (e.g., steering angles and accelerations). 2. Choose a kernel function (Gaussian is a good start). 3. Select a bandwidth (you can use cross-validation or a rule of thumb). 4. Apply the KDE formula to estimate the probability density.
This gives you a smooth estimate of the probability distribution of your data, which you can use for further analysis or prediction.

V. Bootstrapping

A. Bootstrapping the Samples

Given a set of real-world samples $\{x_1, x_2, \ldots, x_n\}$, each bootstrap sample $\mathbf{X}_i^*$ is defined as $\{x_{i1}^*, x_{i2}^*, \ldots, x_{in}^*\}$ is obtained by randomly sampling from the original set with replacement.

B. Implying the above logic in our system (Refer to Table 1)

$\mathcal{F}_t$ is the bootstrapped distribution at time $t-1$ from $\mathcal{F}_1$ $\forall$ $t \geq 1$. The true value of $\Delta\theta_t$ at t will be observed from the predicted $\mathcal{F}_t$ at t-1.

To reach the underlying distribution of $\Delta\theta_t$, we will find the **Mean Square Error** at time t defined as $\mathbf{MSE_t}$, between predicted $\mathcal{F}_{t+1}$ & $\mathcal{F}_1$; and we will try to minimise this MSE as time progresses.

The MSE_t between two distributions (lists of values) $\mathcal{F}_1$ (given real-world data) $\{x_1, x_2, \ldots, x_n\}$ and $\mathcal{F}_{t+1}$ $\{\hat{x}_{t+1,1}, \hat{x}_{t+1,2}, \ldots, \hat{x}_{t+1,n}\}$ (data predicted at time t) is defined as:

$$\text{MSE}_t = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_{t+1,i})^2 \tag{10a}$$

The aim is to minimise this MSE, and take it to below some predefined threshold value MSE_{min} so that for some t, $\mathcal{F}_{t+1}$ is the distribution nearest to the unknown underlying distribution.

More information to be added

VI. KDE combined with Bootstrapping

A. Creating KDE Distributions with Varying Bandwidths

Once we have the bootstrapped samples, we create a KDE for each sample. The bandwidth of the KDE determines the smoothness of the estimated density. We explore two methods for selecting the bandwidth:

B. Silverman's Rule of Thumb

Silverman's Rule of Thumb provides a heuristic for selecting the bandwidth in KDE:

$$h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \tag{11}$$

where n is the number of samples, and $\hat{\sigma}$ is the sample standard deviation.

C. Scott's Rule

Scott's Rule provides another approach for bandwidth selection:

$$h = n^{-\frac{1}{5}}\hat{\sigma} \tag{12}$$

where n is the number of samples, and $\hat{\sigma}$ is the sample standard deviation.

Both methods aim to minimize the mean integrated squared error (MISE) in KDE.

We find the bandwidth for each bootstrap sample using either of these methods and create its own KDE.

D. Combining KDE Distributions

Finally, we average the estimated densities to create a final KDE distribution from the individual KDEs. If $f_i(x)$ is the KDE for the i-th bootstrapped sample with bandwidth h_i, take the number of bootstrap samples to be 500, the final combined KDE $f(x)$ is:

$$f(x) = \frac{1}{500}\sum_{i=1}^{500} f_i(x)$$

Each $f_i(x)$ is given by:

$$f_i(x) = \frac{1}{nh_i}\sum_{j=1}^{n} K\left(\frac{x - x_j^*}{h_i}\right)$$

where K is the kernel function, typically a Gaussian function:

$$K(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}$$

VII. Bayesian Inference

When we have an existing distribution, and new data becomes available, we can update the existing distribution to reflect the latest information using Bayesian updating.

A. Bayesian inference

Given some observed data, this method provides a strong theoretical framework to predict the underlying probability distribution.

1) We assume a prior distribution. Since we don't want to make any assumptions, we assume a uniform distribution.
$$\theta \sim U(0, 1)$$
2) Suppose we have some experimental data from the original underlying distribution. We wish to update our current understanding of the PDF on the basis of this new information.
We first calculate the likelihood ($p(D|\theta)$) where $p(\theta)$ is the prior distribution and $D = \{x_1, x_2, \ldots, x_n\}$ is the observed data.
3) Next, we calculate $p(D)$ i.e. probability of observed data independent of θ using the formula
$$p(D) = \int_0^1 p(A|\theta)f(\theta)d\theta$$
4) Now, using the Bayes rule, we can update the priori distribution based on the new data such that
$$p(\theta|D) \propto \frac{p(D|\theta)p(\theta)}{p(\theta)}$$

From this, we can find posterior distribution, which can be used to predict future probabilities and can be updated using new original data observed at each time step.

Though this method is quite robust and easy to use, it is Parametric, i.e. it assumes a prior distribution to work with (Uniform in this case). If the initial distribution is very different from the underlying distribution, it may take a long time to reach the correct underlying distribution.

B. Updating the KDE Distribution

Suppose we have an initial KDE distribution $f_{prior}(x)$ representing our prior knowledge of the distribution. After observing new data $D = \{x_1, x_2, \ldots, x_n\}$, we can update this distribution using MAP estimation.

1) Construct the Likelihood: Assume the observed data D follows the KDE distribution $f_{prior}(x)$. The likelihood $p(D|f_{prior})$ can be constructed from the KDE's evaluation on the new data points.
2) Update the Prior: Combine the prior KDE with the likelihood to get the posterior KDE. We achieve this by adjusting the KDE to incorporate the new data points. This can be done by re-estimating the KDE with the latest data added to the original dataset or by weighting the prior and new KDEs.
3) Find the MAP Estimate: The MAP estimate in this context is the updated KDE that maximizes the posterior distribution.

VIII. REVISED METHODOLOGY

The aforementioned approaches, which consider discrete time instants, are found to be inefficient for predicting the underlying distribution. Consequently, we propose a modified approach. Assume we are given the true values of $\Delta\theta$ observed during the vehicle's motion. We aim to determine the true underlying distribution of the random variable $\Delta\theta$, once the vehicle's motion has been completed and the set of true $\Delta\theta$ values has been obtained. To achieve this, we will employ the previously discussed techniques, namely Kernel Density Estimation (KDE), Bootstrapping, and a combination of KDE with Bootstrapping, to ascertain the underlying distribution from the provided data sample.

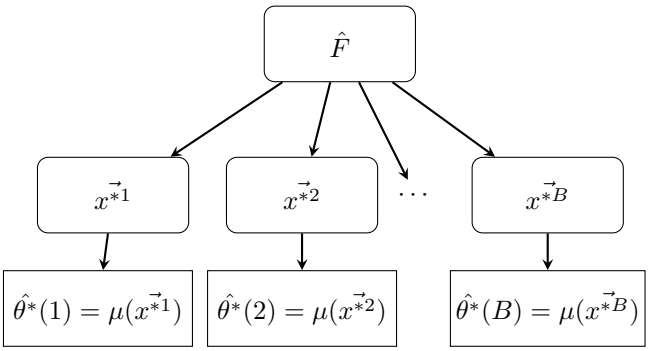
A. Initial Conditions and Assumptions

- Let the true underlying (unknown) distribution of $\Delta\theta(t)$ be denoted as $\mathcal{F}$. Let the true unknown mean be represented as θ.
- Let there be an empirical distribution $\hat{\mathcal{F}}$ representing the set of true $\Delta\theta$ values obtained once the vehicle's motion has been completed. Let the mean of this empirical distribution be $\hat{\theta}$, serving as the plug-in estimate.

IX. ESTIMATING UNCERTAINTY WITH CONFIDENCE INTERVALS

Confidence intervals provide a range of values within which we expect the true population parameter (true mean in this case) to lie, giving us a measure of uncertainty and reliability about our sample estimates. To find the most reliable confidence interval, we will try out 5 different approaches, which assume the underlying true distribution to be some distribution, and then provide a confidence interval. Finding the confidence interval, assuming the true underlying distribution to be:

A. Standard Normal



In the above chart,

- $\hat{F}$ represents the empirical distribution given to us.
- $\vec{x^{*i}}$ represents the i-th bootstrap sample $\forall i \in [1, B]$; (resampled with replacement from the given data).
- $\hat{\theta^*}(i)$ represents the bootstrap replication of $\hat{\theta}$;
- The mean of the given population (plug-in estimate) is represented as $\hat{\theta}$ s.t. $\hat{\theta} = \mu(\hat{F})$.

Now, find the standard error associated with the means of the bootstrapped samples.

$$\hat{\text{se}}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta^*}(b) - \hat{\theta^*}(.)\right)^2}$$

where, $\hat{\theta^*}(.) = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\theta^*}(b)\right)^2$

The confidence interval can be written as:

$$\hat{\theta} \pm Z^{(1-\alpha)} \cdot \hat{\text{se}}_B$$

$$\implies \theta \in \left[\hat{\theta} - Z^{(1-\alpha)} \cdot \hat{\text{se}}_B,\ \hat{\theta} - Z^{\alpha} \cdot \hat{\text{se}}_B\right]$$

where,
$\hat{Z}^{(1-\alpha)} = -\hat{Z}^{\alpha}$
$\hat{Z}^{\alpha}$ represents the $100.\alpha^{th}$ percentile point of N(0, 1) distribution.
$\hat{\theta}$ represents the plug-in estimate (the sample's mean).
θ represents the true mean of the underlying distribution.

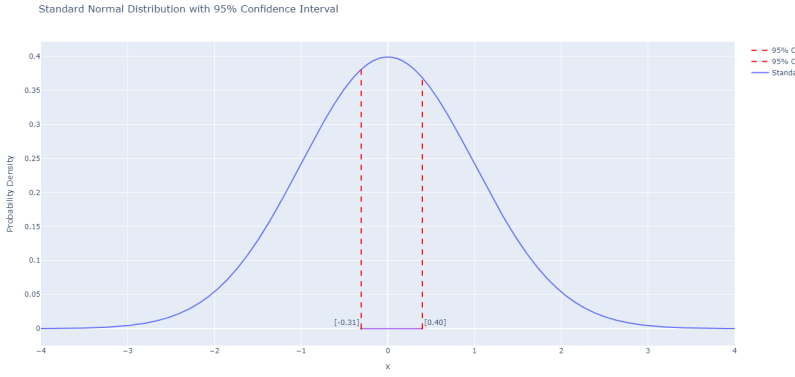


Fig. 1. Confidence Intervals (95% confidence) showing the uncertainity of the true mean.

B. Student-t

All the calculation remain same as done in Section IX-A. The assumed underlying distribution in this case will be student-t distribution. By using this, we induced the notion of known mean and unknown variance; different from the case of known mean and variance in standard normal.
The confidence interval can be written as:

$$\hat{\theta} \pm t_{n-1}^{(1-\alpha/2)} \cdot \hat{\text{se}}_B$$

$$\implies \theta \in \left[\hat{\theta} - t_{n-1}^{(1-\alpha/2)} \cdot \hat{\text{se}}_B,\ \hat{\theta} + t_{n-1}^{(1-\alpha/2)} \cdot \hat{\text{se}}_B\right]$$

where,

- n is the degrees of freedom (given by the sample size minus one).
- $t_{n-1}^{(1-\alpha/2)} = -t_{n-1}^{\alpha/2}$: $t^{\alpha/2}$ represents the $(100 \times \alpha/2)^{\text{th}}$ percentile point of the Student's t-distribution.
- $\hat{\theta}$ represents the plug-in estimate (the sample's mean).
- θ represents the true mean of the unknown underlying distribution.

More about the t-distribution:

- Mean = 0
- Degrees of Freedom (df or ν) = number of independent values or quantities that can vary in a statistical calculation.
- In our case, df is the sample size minus one, as the calculation of the mean reduces the degrees of freedom by one.

3.

$$\text{Variance} = \begin{cases} \frac{\nu}{\nu-2} & \text{for } \nu > 2 \\ \infty & \text{for } \nu < 2 \end{cases}$$

4. For large ν, variance = 1; resulting in $\mathcal{N}(0,1)$.
5. n=1 forms Standard Cauchy Distribution.

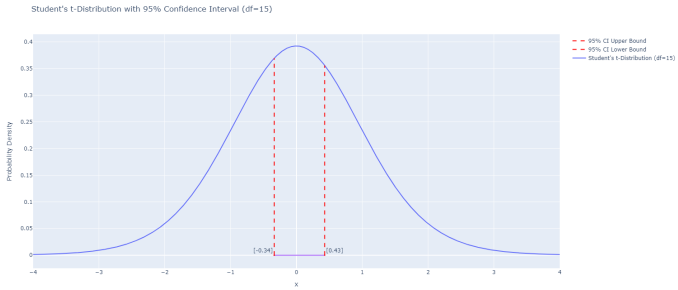


Fig. 2. Confidence Intervals (95% confidence) showing the uncertainity of the true mean {degrees of freedom = 15}

C. Bootstrap-t

Add information

D. Percentile Method

The Percentile Method is a way of empirically finding the Confidence Interval of a statistic. We start with initial data and we remove α percentile of data from both the Left and the right sides. The remaining data should ideally be a 95% confidence interval (given α = 0.05.) This method is computationally simple yet accurate when compared to the methods above. Majorly due to 2 properties:

- **Transformation Respective Property**: Consider that the underlying distribution is not standard normal. But there exists a transformation ϕ such that θ becomes normal. Now the confidence interval taken from the distribution ϕ, finding the 95% using the standard normal and using an inverse function to get the data w.r.t. to distribution θ is extremely close to stripping away α percentile from both left and right of the given sample data. We received similar results without finding ϕ and inverse of it or creating a standard normal. So, we can say that the Percentile method automatically knows the best transformation.[1]
- **Nonparametric assumption**: Percentile method doesn't make any assumptions about the underlying distribution due to this in a biased or a skewed distribution, it will still produce better results than students-t or Standard Normal, which assume a Standard Normal and underlying distribution. For better results in biased and skewed distributions, see section IX-E.

Let $X_1, X_2, \ldots, X_n$ be a sample of size n. The α percentile is denoted as P_α, where α is the significance level.
The percentile confidence interval is given by:

$$\left(X_{(\alpha/2)\times 100\%}, X_{(1-\alpha/2)\times 100\%}\right)$$

Assume a transformation ϕ such that:

$$Z = \phi(X) \sim N(\mu, \sigma^2)$$

$$\left(\phi^{-1}(Z_{\alpha/2}), \phi^{-1}(Z_{1-\alpha/2})\right)$$

$$CI = \left(X_{(\alpha/2)\times 100\%}, X_{(1-\alpha/2)\times 100\%}\right)$$

This interval does not rely on assumptions about the distribution of X.

E. Bias-corrected and accelerated (BCa)

BCa method takes the concepts of the percentile method and improves the interval by accounting for the bias z_0 and skewness (acceleration)

Steps for Calculating BCa Confidence Interval

The bias-correction term z_0 is calculated by:

$$z_0 = \Phi^{-1}\left(\frac{\#\{\theta_b^* < \hat{\theta}\}}{B}\right)$$

where Φ^{-1} is the inverse cumulative distribution function (CDF) of the standard normal distribution, and B is the number of bootstrap samples.

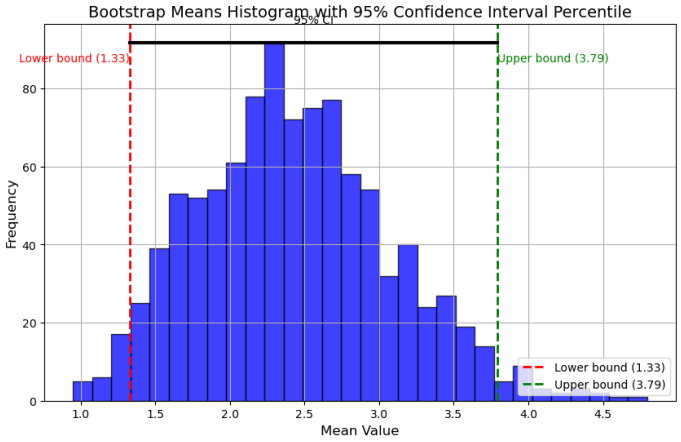


Fig. 3. Percentile Method (95% confidence) showing the Confidence Interval for a Log Normal Distribution

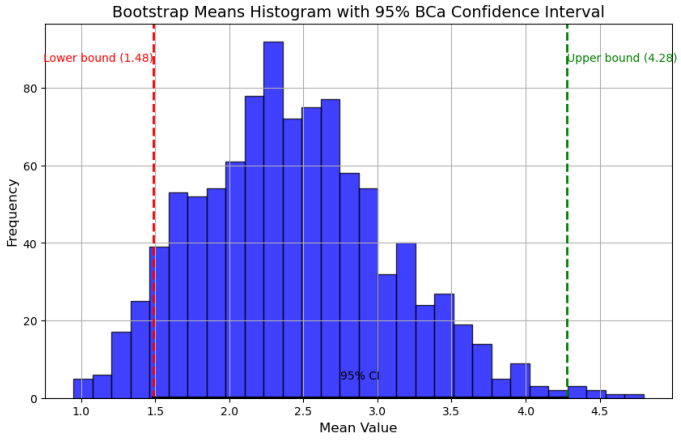


Fig. 4. BCa method (95% confidence) showing the Confidence Interval for a Log Normal Distribution

The acceleration term a is calculated using the jackknife method:

$$a = \frac{\sum_{i=1}^{n}(\bar{\theta}_{(-i)} - \bar{\theta}_{(\cdot)})^3}{6\left(\sum_{i=1}^{n}(\bar{\theta}_{(-i)} - \bar{\theta}_{(\cdot)})^2\right)^{3/2}}$$

where: - $\bar{\theta}_{(-i)}$ is the mean estimate when the i^{th} observation is removed (jackknife mean). - $\bar{\theta}_{(\cdot)}$ is the average of all jackknife means.
The adjusted percentiles for the confidence interval are given by:

$$\alpha_1 = \Phi\left(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a(z_0 + z_{\alpha/2})}\right)$$

$$\alpha_2 = \Phi\left(z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a(z_0 + z_{1-\alpha/2})}\right)$$

The BCa confidence interval is given by:

$$\left[\theta^*_{\alpha_1 \times 100\%}, \theta^*_{\alpha_2 \times 100\%}\right]$$

where $\theta^*_{\alpha_1}$ and $\theta^*_{\alpha_2}$ are the bootstrap estimates at the adjusted percentiles.

Comparison: Miss Left and Miss Right

To evaluate the performance of confidence interval methods, we can see the difference between the symmetry of intervals using the miss left and miss right percentages (for 95% interval miss left = miss right = 2.5%)

- **Miss Left:** Occurs when the true parameter value is less than the lower bound of the confidence interval.
- **Miss Right:** Occurs when the true parameter value is greater than the upper bound of the confidence interval.

BCa vs. Percentile Method

The BCa method is generally more accurate in achieving the desired coverage probability as it adjusts for bias and skewness. This improves the frequency of ”miss left” and ”miss right” cases, resulting in more balanced confidence intervals:

- The BCa interval corrects for bias, ensuring that the intervals are not shifted to one side of the distribution.
- By accounting for skewness, the BCa method reduces asymmetry in the intervals, which helps in minimizing both ”miss left” and ”miss right” percentages bringing them closer to the ideal percentages.

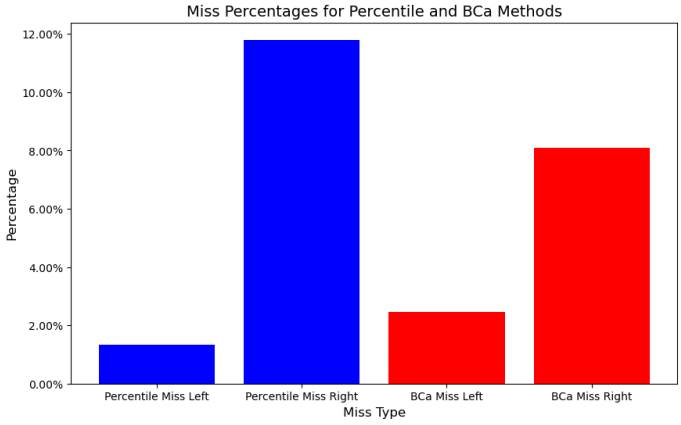


Fig. 5. The BCa method’s miss left and miss right are closer to 2.5% compared to Percentile method

Conclusion

Add conclusion
[1] [2]

References

[1] B. Efron, *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

[2] F. Edition, A. Papoulis, and S. U. Pillai, *Probability, random variables, and stochastic processes*. McGraw-Hill Europe: New York, NY, USA, 2002.