NON-PARAMETRIC TOLERANCE INTERVALS FOR SMALL SAMPLE SIZES

An empirical likelihood approach
1 Motivation
   1.1 Tolerance Intervals? | 1.2 Problems and Approach

2 Methodology
   2.1 Existing Methods | 2.2 Extension | 2.3 Performance

3 Implementation in JMP
   3.1 Scripting | 3.2 Showcase
Tolerance Intervals

- Why Tolerance Intervals?
  - Upper limit for failures
  - Statement about (almost) all future observations

- Why non-parametric?
  - Normal distribution is common but not universal

- Why small samples?
  - Generating data can be costly
  - Destructive tests are necessary
Demonstration in JMP
Problems: Summary

- Tolerance Intervals
  - Unable to calculate non-parametric tolerance Intervals for small sample sizes

- Idea: Calculate confidence intervals for quantiles.
  - Non-parametric approach from JMP: empirical likelihood
  - Problem: Confidence are not usable for extreme Quantiles and small sample sizes
Approach

▪ How does smoothed empirical likelihood Work?
  ▪ Are there different methods?

▪ How does JMP calculate the confidence Intervals?

▪ Is there a way to eliminate the problems with small sample sizes and extreme Quantiles?
**Existing Methods**

- Empirical likelihood was first introduced by Owen (1988)

- When quantiles are considered, the log likelihood is dependent on the empirical distribution function $F_n$ (Adimari, 1998):
  \[
  l(\Theta) = 2n \left[ F_n(\Theta) \log \left( \frac{F_n(\Theta)}{q} \right) + (1 - F_n(\Theta)) \log \left( \frac{1 - F_n(\Theta)}{1 - q} \right) \right]
  \]

- Confidence intervals can be calculated by using Wilks theorem Owen (1988):
  \[
  \lim_{n \to \infty} P(l(\Theta) \leq c) = P(\chi^2_1 \leq c)
  \]

- Several methods of smoothing exist:
  - Smoothing using a kernel function. (Chen, Hall, 1993) (JMP)
  - Linear smoothing of $F_n$ (Adimari, 1998)
Comparison. Median

Figure: Smoothed empirical likelihood functions ($n = 11; q = 0.5; \alpha = 0.05$)
Comparison 1%-Quantile

Figure: Smoothed empirical likelihood functions ($n = 11; q = 0.01; \alpha = 0.05$)
Coverage Rates

Figure: Estimated coverage rates based on 1000 samples
Linear smoothing

- The smoothing proposed by Adimari (1998) is achieved by using a linear smoothing $F^*$ of $F_n$:

$$F^*_n(\Theta) = \begin{cases} 
0 & \text{if } \Theta < x(1) \\
H(\Theta) & \text{if } \Theta \in [x(1), x(n)) \\
1 & \text{if } \Theta \geq x(n) 
\end{cases}$$

where

$$H(\Theta) = \begin{cases} 
\frac{2i-1}{2n} & \text{if } \Theta = x(i); i \in \{1, \ldots, n-1\} \\
(1 - \lambda) \frac{2i-1}{2n} + \lambda \frac{2i+1}{2n} & \text{if } \Theta \in (x(i), x(i+1)); \lambda = \frac{\Theta - x(i)}{x(i+1) - x(i)}; i \in \{1, \ldots, n-1\}
\end{cases}$$
Empirical distribution function

Figure: Smoothing of $F_n$ for 11 observations from a standard normal distribution
Extending the empirical distribution function

- Find an extension of the likelihood function for values outside of the observed data so that
  - finite confidence intervals are guaranteed.
  - the desired coverage rate is achieved.

1. extending \( F^* \) as follows:

\[
F_{ext}(\Theta) = \begin{cases} 
0 & \text{if } \Theta \leq x(1) - d_1 c \\
\frac{1}{2n} - \frac{1}{2n*d_1+c}(x(1) - \Theta) & \text{if } x(1) - d_1 c < \Theta < x(1) \\
H(\Theta) & \text{if } x(1) \leq \Theta \leq x(n) \\
\frac{2n-1}{2n} + \frac{1}{2n*d_2+c}(\Theta - x(n)) & \text{if } x(n) < \Theta < x(n) + d_2 c \\
1 & \text{if } \Theta \geq x(n) + d_2 c
\end{cases}
\]

Where \( c \geq 1; \ d_1 = \frac{1}{10} \sum_{i=1}^{5} (x(i+1) - x(i)) \) and

\[
d_2 = \frac{1}{10} \sum_{i=1}^{5} (x(n-i+1) - x(n-i))
\]

2. linear extension of the likelihood function.
Figure: Smoothing of $F_n$ using $F_{ext}$
Figure: Visualising the influence of the parameter $c$ on $F_{ext}$
Example

1%-Quantile

- not smoothed
- Adimari
- extension 1. step
- $\chi^2_{1,1-\alpha}$
- true quantile

Figure: First step of the extension ($c = 3$)
Figure: Second step: further linear extension ($c = 3$)
Example

**1%-Quantile**

- not smoothed
- Adimari
- extension (c=3)
- extension (c=25)

**Figure:** Fully extended likelihood function for different values of c
Assuring Coverage

- The quality of the Confidence Interval is dependent on the extension parameter c.
- The smallest value of c which results in a coverage rate of at least \((1 - \alpha)\) is desired.

- A simulation study was carried out under the following assumptions:
  - The required value for c depends on q, n, R and \(\alpha\)
  
  \[ R := \begin{cases} 
  q \times n & \text{if } q \leq 0.5 \\
  (1 - q) \times n & \text{if } q > 0.5 
  \end{cases} \]

  - The required value for c does not depend on the distribution of the data.
Simulation Study

Figure: Chosen value of $c$ for different values of $R$
Simulation Study

- Model for $c$: $\hat{c} = 12.344 - 7.082\sqrt{R} - 2.454\log(\alpha) - 75.125q - 0.004n$
- Adjusted $R^2 = 0.933$
Example

Figure: Example for the modeled value of $c$ ($n = 11, q = 0.01, \alpha = 0.05$)
Coverage Rates

Figure: coverage rates based on 1000 samples for normally distributed data
Coverage Rates

Figure: coverage rates based on 1000 samples for exponentially distributed data
**Coverage Rates**

Averages:
- Coverage: 0.941
- Width: 3.89

Figure: coverage rates based on 1000 samples for exponentially distributed data
Problems and Solutions

- The required value of $c$ is not depended on the distribution of the Data
- However, a vast improvement compared to the existing methods is achieved.
  - Drop in quality for data with very light/heavy tails

- To further improve the method, a semi parametric approach is proposed:
  - Assume a distribution of the data and develop a model for the extension parameter using that distribution
  - Example: Exponential distribution
Modelling different distributions

Figure: coverage rates based on 1000 samples for exponentially distributed data
Demonstration in JMP
Implementation: Summary

- Straight forward programming of the functions and models

- Difficulty: Finding an algorithm for the borders of the CI
  - Minimize function: Fast but unstable in some situations
  - Simple self-made algorithm: slower but stable

- Development of a simple JMP application for user friendliness.