General-Purpose Open-Source Program for Ultra Incomplete Data-Oriented Parallel Fractional Hot Deck Imputation (UP-FHDI)

Yicheng Yang, Jae-Kwang Kim, and In Ho Cho

Supported by
NSF Cyberinfrastructure for Sustained Scientific Innovation
Grant #: 1931380
Incomplete Data in Engineering and Science

Incomplete Data in Infrastructure Engineering

- Hybrid Data Set from Bridge and Transportation Sensor Data
- Shear Wall Structure Database (ACI 445-B; SERIES; BRI Wall Database)

(raw data from Dr. Phares and Dr. Sharma)

(domain-specific community database)
Naïve Remedy for Imputation

Widely Used Naïve Method in ML Community

- Naive imputation: simply use each variable’s mean to impute missing values
- **Removal of** the entire unit (instance) which has missing values

Statistical problems resulting from the naïve remedy

- Loss of substantial information
- May introduce unexpected bias
- May lead to low accuracy in machine learning/statistical predictions
- May mislead incorrect statistical inference
Multiple Imputation (MI)

- One of the most popular imputation methods
- Create $M$ completed datasets for full imputation uncertainty
- Since Rubin (1976), extensive investigations have been conducted (Rubin 1987, Schafer 1997, Little and Rubin 2002, etc.)

**Typical Multiple Imputation Steps**

- Incomplete Sci. & Eng. Data
- Imputed data sets
- Analysis results
- Result Pooling
- $M$ imputed data sets

**Final results**

Statistical Analysis on each set
Challenges of Existing Imputation Methods for Big Incomplete Data

- They often require statistical and/or distributional assumptions, which are obstacles for general researchers.
- Computational limits of them prevent general applications to large/big incomplete data in broad Eng. or Sci.

Popular Imputation Methods: MI

Difficulty in General Use of Multiple Imputation

MI requires
- “congeniality” condition (Meng 1994) and
- “self-efficient” estimation (Meng and Romero 2003)

If not, the MI variance estimator may be
- inconsistent (Nielsen 2003; Kim et al. 2006) and
- considerably biased (Beaumont et al. 2011).
Our Choice for Big Data Imputation: Fractional Hot Deck Imputation

Our group developed and shared a public, open-source R package “FHDI” (The R Journal, 2018 [1])

Strengths of “Hot Deck” Imputation

• Do not require “self-efficient” estimation condition
• Do not create artificial values, instead use the real observations
• Do not need model/distributional assumptions
• Seek to leverage and preserve the joint probability of data available.

Still, FHDI is not suitable for tackling big incomplete data
Motivations to develop UP-FHDI

- Limitations of the serial version R package FHDI regarding time and memory requirements
- Hard to deal with large/big data with immense volume and/or too many variables
- The positive impact of FHDI on learning and prediction (Cho et al. *IEEE, TKDE*, 2019 [2])
- As we enter the era of big data and powerful computing, parallel computing techniques are gradually attempted in imputations.
- Strong need for general-purpose and assumption-free big data (big-$n$ and/or big-$p$) imputation tools
Parallel Fractional Hot Deck Imputation for Ultra Data

Types of large/big incomplete datasets

- **big-\(n\) data**: \(n \gg p\)
e.g., \(n = 1M, p = 4\)
- **big-\(p\) data**: \(n \leq p\)
e.g., \(n = 1000, p = 10,000\)
- **Ultra data**: \(n\) and \(p\) are both large
e.g., \(n = 1M, p = 10,000\)

\(n\): number of instances
\(p\): number of variables
\(\eta\): missing rate

Tackled by P-FHDI ver. 1.0
(Cho et al. *IEEE, TKDE*, 2020 [3])

UP-FHDI
Key Procedures of UP-FHDI

• **Parallel Cell Construction** (denoted as Process 1)
  Categorization of imputation cells
  Donor selection in conjunction with the sure independence screening (SIS) and K-nearest neighbor (KNN) searching

• **Parallel Cell Probability Estimation** (Process 2)
  Estimate probability for each unique observed cell pattern using EM algorithm

• **Parallel Imputation** (Process 3)
  Missing values are imputed by donors

• **Parallel Variance estimation** (Process 4)
  - Jackknife method for moderately large data
  - Linearized variance estimation for ultra data
Parallel Computing Techniques

Library: MPI (Message Passing Interface)
• Suitable for distributed memory
• Specifies names, calling sequences, and results of functions/subroutines needed to communicate via message passing
• Language bindings for C/C++ and Fortran

High-performance computing (HPC) facilities:
• Condo2017 [5]: 158 servers. Each server has two 8-core Intel Haswell processors, 128 GB of memory and 2.5 TB local storage
• TACC Stampede2 [6]: 4704 servers. Each server has 192 GB of memory and no quota for local storage
How UP-FHDI Process Big Data?

- Processors can communicate via communication channel
- Store input data and temporary data in local storage provided by HPC facilities
- Slave processors only fetch required data to their memory
- Optimal Overload IO Protection System (OOOPS) adjusts intensive IO workload

Parallel file system on $Q$ processors indexed by $0, \cdots, Q - 1$.

Intensive IO may harm global distributed system of HPC

Adapted from Cho et al. *IEEE, TKDE*, 2021 [4]. Under review
Available Example Datasets for UP-FHDI

- Following example big-\(n\) or big-\(p\) datasets are available at IEEE DataPort [7]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Variable type</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic data 1</td>
<td>Continuous</td>
<td>(U(1000, 4, 0.25))</td>
</tr>
<tr>
<td>Synthetic data 2</td>
<td>Continuous</td>
<td>(U(10^6, 4, 0.25))</td>
</tr>
<tr>
<td>Air Quality Nursery</td>
<td>Hybrid</td>
<td>(U(41757, 4, 0.1))</td>
</tr>
<tr>
<td>Synthetic data 3</td>
<td>Continuous</td>
<td>(U(12960, 5, 0.3))</td>
</tr>
<tr>
<td>Synthetic data 4</td>
<td>Continuous</td>
<td>(U(15000, 12, 0.15))</td>
</tr>
<tr>
<td>Synthetic data 5</td>
<td>Continuous</td>
<td>(U(15000, 16, 0.15))</td>
</tr>
<tr>
<td>Synthetic data 6</td>
<td>Continuous</td>
<td>(U(15000, 100, 0.15))</td>
</tr>
<tr>
<td>Synthetic data 7</td>
<td>Continuous</td>
<td>(U(1000, 100, 0.3))</td>
</tr>
<tr>
<td>Synthetic data 8</td>
<td>Continuous</td>
<td>(U(1000, 10000, 0.3))</td>
</tr>
<tr>
<td>Appliance Energy</td>
<td>Continuous</td>
<td>(U(19735, 26, 0.15))</td>
</tr>
</tbody>
</table>

Adapted from Cho et al. *IEEE, TKDE*, 2020 [3].

Note that \(U(n, p, \eta)\) represents incomplete data with \(n\) rows and \(p\) columns with \(\eta\) missing rate.

- Please refer to (Cho et al. *IEEE, TKDE*, 2020 [3]) for more details
- Source codes of parallel FHDI are available and executable on local HPC or NSF Cloud Computing (e.g. NSF XSEDE).
Available Example Datasets for UP-FHDI

- Following real-world ULTRA datasets are available at IEEE Dataport [7]

<table>
<thead>
<tr>
<th>Dataset name</th>
<th># Instances</th>
<th># Variables</th>
<th>Category</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swarm</td>
<td>24016</td>
<td>2400</td>
<td>Biology</td>
<td>UCI</td>
</tr>
<tr>
<td>CT</td>
<td>53500</td>
<td>380</td>
<td>Medicine</td>
<td>UCI</td>
</tr>
<tr>
<td>P53</td>
<td>31159</td>
<td>5408</td>
<td>Genetics</td>
<td>UCI</td>
</tr>
<tr>
<td>Radar</td>
<td>325834</td>
<td>175</td>
<td>Agriculture</td>
<td>UCI</td>
</tr>
<tr>
<td>Travel</td>
<td>23772</td>
<td>50</td>
<td>Transportation</td>
<td>IEEE DataPort</td>
</tr>
<tr>
<td>Bridge</td>
<td>492641</td>
<td>31</td>
<td>Civil</td>
<td>Dr. Cho</td>
</tr>
<tr>
<td>Earthquake</td>
<td>901512</td>
<td>15</td>
<td>Civil</td>
<td>USGS</td>
</tr>
</tbody>
</table>

- Please refer to (Cho et al. IEEE, TKDE, 2021 [4]) for more details
- Source codes of UP-FHDI are available and executable on local HPC or NSF Cloud Computing (e.g. NSF XSEDE).
Strength of UP-FHDI

• The UP-FHDI inherits all strengths of the general-purpose, assumption-free FHDl
• UP-FHDI can cure incomplete synthetic data with one million instances and 10,000 variables of 80 GB (30% missing rate) in 35 hours with 240 processors
• UP-FHDI positively improves the subsequent machine learning
• The UP-FHDI is now publicly available
• Researchers in broad engineering and science can cure general, large/big data sets with ease
Reference


Thank you!

For programs, data sets, and discussion feel free to contact

icho@iastate.edu

Acknowledgement: Generous support from NSF CSSI, OAC-1931380
Supplementary Materials
Generate High-Dimensional Synthetic Data

Let $i = 0$ and repeat the following by setting $i = i + 4$ until we obtain $p$ variables:

$$Y_i = \begin{cases} 1 + e_i, & \text{if } i = 0 \text{ or } i\%8 = 0 \\ Y_{i-1} + e_i, & \text{if } i\%8 \neq 0 \end{cases}$$

$$Y_{i+1} = Y_i + 2 + \rho \times e_i + \sqrt{1 - \rho^2} e_{i+1}$$

$$Y_{i+2} = Y_{i+1} + e_{i+2}$$

$$Y_{i+3} = -1 + Y_i + 0.25Y_{i+1} + e_{i+3}$$

where $\rho = 0.5$ and $e_i, e_{i+1}, e_{i+3}$ are randomly generated by normal distribution. And $e_{i+2}$ is generated by gamma distribution.