Parallel Fractional Hot Deck Imputation for Large/Big Missing Data Curing for Improving Machine Learning and Statistical Inference

In-Ho Cho
in collaboration with Jae-Kwang Kim

Iowa State University Team of NSF Cyberinfrastructure for Sustained Scientific Innovation

TADS Seminar
October 15th, 2020
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)
Incomplete Data in Engineering and Science

Incomplete Data in Broad Science

- Phenotype Database [Genomes To Fields (G2F) Community Data]

<table>
<thead>
<tr>
<th>Name of the variables</th>
<th>Missing rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand count</td>
<td>1%</td>
</tr>
<tr>
<td>Pollen DAP</td>
<td>22%</td>
</tr>
<tr>
<td>Silk DAP</td>
<td>26%</td>
</tr>
<tr>
<td>Plant height</td>
<td>2%</td>
</tr>
<tr>
<td>Ear height</td>
<td>2%</td>
</tr>
<tr>
<td>Root loading</td>
<td>18%</td>
</tr>
<tr>
<td>Stalk loading</td>
<td>5%</td>
</tr>
<tr>
<td>Grain Moisture</td>
<td>2%</td>
</tr>
<tr>
<td>Test Weight</td>
<td>24%</td>
</tr>
<tr>
<td>Plot Weight</td>
<td>2%</td>
</tr>
<tr>
<td>Grain yield</td>
<td>2%</td>
</tr>
</tbody>
</table>

(raw data from Dr. Lawrence et al.)

- Effects of genotype and environment on the performance of a large collection of maize hybrids.

- comprises genotypic, phenotypic, and environmental data from more than 30 North American field locations across 3 years.

- These data include 14 phenotypic traits, weather measurements, locations, and image data.
Impact of Data Curing on Machine Learning and Statistical Prediction

- **FHDI**: Fractional hot deck imputation of our group
- **GAM**: Generalized Additive Model (an advanced statistical prediction model)
- **ERT**: Extremely Randomized Trees (an advanced machine learning method)
- **ANN**: Artificial Neural Network (a popular machine learning)
- **Naïve** method: missing data is cured by variable’s mean value
- **RMSE**: Root mean square error of prediction

**Phenotype data set used**
- Total instances, $n = 5,931$,
- Total variables, $p = 13$,
- Effects of genotype on maize hybrid yield by courtesy of Dr. Lawrence.

Naïve Remedy for Imputation

Widely Used Naïve Method in ML Community

• Simply use each variable’s mean to impute missing values
• Delete 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
**Popular Imputation Methods?**

**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.)

![Diagram](image)

**Typical Multiple Imputation Steps**
- Incomplete Sci. & Eng. Data
- Imputed data sets
- Analysis results
- Result Pooling

- $M$ imputed data sets
- Statistical Analysis on each set
- Final results
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).

Challenges of Existing Imputation Methods
• They often require statistical and/or distributional assumptions, which are obstacles for general researchers.
• Even with a good expertise, computational limits of them prevent general applications to large/big incomplete data in broad Eng. or Sci.
Our Choice for Big Data Imputation: Fractional Hot Deck Imputation

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.
Our Choice for Big Data Imputation: Fractional Hot Deck Imputation

**Fractional imputation for the Hot Deck Imputation**


- An efficient way of achieving hot deck imputation

- As in MI, $M$ imputed values are generated for each missing value

- But a single data set is created after imputation

- **Fractional weights** are assigned to the imputed values

- Replication methods can be directly used for variance estimation
Fractional Hot Deck Imputation

Two types of fractional hot deck imputation estimator

• Our group developed and shared a public, open-source R package “FHDI” ([*The R Journal*, 2018 [5]]) based on theories of

(1) Fully efficient fractional imputed (FEFI) estimator
• Replace the missing cells with all observed values (i.e. all donors)
• Produce zero imputation variance

(2) Fractional hot deck imputed (FHDI) estimator
• Replace the missing cells with a few randomly selected donors instead of taking all possible donors
• $M$ denotes the number of selected donors
• Have a variability due to randomly selected donors
• Seek to mimic FEFI’s performance at a cheap computational cost
Fractional Hot Deck Imputation: Theory

Basic Setup

\[ A \subset U \]

- \( A \) is an index set of a sample of size \( n \) drawn from a finite population’s index set \( U \), \( U = \{1, 2, \ldots, N\} \): Finite population
- \( y_i \in \mathbb{R}^p \) is the \( i \)-th row of the data matrix \( Y \in \mathbb{R}^{N \times p} \); \( y_i = (y_{1i}, \ldots, y_{pi}) \)

Imputation Cell Creation

- Each variable is categorized into \( k \) categories based on its CDF, i.e., \( j \)-th continuous variable \( y_{ji} \) transformed into discrete \( z = (1, \ldots, k_j) \); \( y \to z \).
- Each row having missingness is decomposed into observed and missing part \( y_i = (y_{i,obs}, y_{i,mis}) \to z_i = (z_{i,obs}, z_{i,mis}) \).
- Based on \( z \), the index set \( A \) is decomposed \( A = A_R \cup A_M \)
  - \( A_R = \{ j \in A; \ \delta_j = 1 \} \); \( A_M = \{ j \in A; \ \delta_j = 0 \} \);
  - \( \delta_j = \prod_{l=1}^{p} \delta_{lj} \); \( \delta_{lj} = 0 \) if \( y_{lj} \) is missing whereas 1 if \( y_{lj} \) is observed.
Fractional Hot Deck Imputation: Theory

- “Imputation cell” is defined by a unique pattern of $z$, e.g., if $z = (z_1 \in \{1, \ldots, G\}, z_2 \in \{1, \ldots, H\})$, in total $(G \times H + 1)$ different imputation cells exist.

- Assuming the Rubin’s missing at random (MAR, i.e., missing variables are independent of themselves, but only dependent on observed variables),

- the cell mean model holds for all imputation cells:

$$\mathbf{y}|(z_1 = q, \ldots, z_p = s) \sim ii(\boldsymbol{\mu}_{q \ldots s}, \boldsymbol{\Sigma}_{q \ldots s})$$

- $\sim ii$ denotes independently and identically distributed
- $\boldsymbol{\mu}_{q \ldots s}$ and $\boldsymbol{\Sigma}_{q \ldots s}$ are the cell mean vector and the variance-covariance matrix of $\mathbf{y}$ in the imputation cell $(q, \ldots, s)$, respectively.
Example of Fractional Hot Deck Imputation

Sample observations for the multivariate data vector

\[ y_i = (y_{1i}, y_{2i}, y_{3i}, y_{4i}), \quad i = 1, \ldots, (n = 100) \]

Generated from

- \( Y_1 = 1 + e_1 \)
- \( Y_2 = 2 + \rho e_1 + \sqrt{1 - \rho^2} e_2 \)
- \( Y_3 = Y_1 + e_3 \)
- \( Y_4 = -1 + 0.5Y_3 + e_4 \)

Settings:

- \( \rho = 0.5 \)
- \( e_1, e_2 \sim N(0, 1) \)
- \( e_3 \) from a standard exponential distribution
- \( e_4 \sim N(0, 3/2) \)
- Randomly remove variables by using the Binomial distributions
Example of Fractional Hot Deck Imputation

Serial version FHDI
install.packages("FHDI")

(Im, Cho, Kim, *The R Journal*, 2018)

```r
> library(FHDI)
> example(FHDI)
> summary(daty)

<table>
<thead>
<tr>
<th></th>
<th>y1</th>
<th></th>
<th>y2</th>
<th></th>
<th>y3</th>
<th></th>
<th>y4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>-1.6701</td>
<td></td>
<td>0.02766</td>
<td></td>
<td>-1.4818</td>
<td></td>
<td>-2.920292</td>
<td></td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.4369</td>
<td></td>
<td>1.03796</td>
<td></td>
<td>0.9339</td>
<td></td>
<td>0.781067</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>0.8550</td>
<td></td>
<td>1.79693</td>
<td></td>
<td>1.7246</td>
<td></td>
<td>0.121467</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.9821</td>
<td></td>
<td>1.93066</td>
<td></td>
<td>1.7955</td>
<td></td>
<td>0.006254</td>
<td></td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>1.6171</td>
<td></td>
<td>2.71396</td>
<td></td>
<td>2.5172</td>
<td></td>
<td>0.787863</td>
<td></td>
</tr>
<tr>
<td>Max.</td>
<td>3.1312</td>
<td></td>
<td>5.07103</td>
<td></td>
<td>5.3347</td>
<td></td>
<td>4.351372</td>
<td></td>
</tr>
<tr>
<td>NA’s</td>
<td>42</td>
<td></td>
<td>34</td>
<td></td>
<td>18</td>
<td></td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>
```

IOWA STATE UNIVERSITY

College of Engineering
Example of Fractional Hot Deck Imputation

Imputation Cell Creation

```r
> cdaty=FHDI_CellMake(daty,k=3)
> head(cdaty$data)

   ID WT       y1       y2       y3       y4
[1,] 1 1 1.47963286 2.150860   NA 1.894211796
[2,] 2 1       NA 1.141496 1.6025296 -1.036946859
[3,] 3 1 0.70870936 1.885673 1.2506894    NA
[4,] 4 1       NA 2.753840   NA 1.211049509
[5,] 5 1 0.86273572 2.425549 1.8875492 -0.539284732
[6,] 6 1 0.03460025 1.740481 0.4909525 0.007130484

> head(cdaty$cell)

   y1 y2 y3 y4
[1,] 3 2 0 3
[2,] 0 1 1 1
[3,] 2 2 2 0
[4,] 0 2 0 3
[5,] 2 3 2 2
[6,] 1 2 1 2
```

Serial version FHDI
install.packages("FHDI")

← Original data with missing values

← Categorized data with 0 for missing cell
Unique patterns in the observed and missing parts

- There are 10 unique observed patterns, i.e. $n(A_R) = 10$.
- There are 47 unique missing patterns, i.e. $n(A_M) = 47$.
- e.g., as marked below, there are many possible donors for curing $(0, 0, 0, 2)$
Fractional Hot Deck Imputation: Theory

Hot Deck Imputation Estimator

- Our goal is to estimate population’s mean under missingness

\[ Y_p = \sum_{i=1}^{N} y_{pi} \]

- Taking a form of **finite mixture model**, the conditional distribution is

\[ f(\mathbf{y}_{i,\text{mis}}|\mathbf{y}_{i,\text{obs}}) \approx \sum_{\forall \mathbf{z}_{i,\text{mis}}^*} p(\mathbf{z}_{i,\text{mis}}^*|\mathbf{z}_{i,\text{obs}})f(\mathbf{y}_{i,\text{mis}}|\mathbf{z}_{i,\text{obs}}, \mathbf{z}_{i,\text{mis}}^*) \]

- \( \mathbf{z}_{i,\text{mis}}^* \) denotes one of the possible donors (imputed values) for \( \mathbf{z}_{i,\text{mis}} \)
- \( p(\mathbf{z}_{i,\text{mis}}^*|\mathbf{z}_{i,\text{obs}}) \) is the conditional cell probability of \( \mathbf{z}_{i,\text{mis}}^* \) given \( \mathbf{z}_{i,\text{obs}} \)
Multivariate Hot Deck Imputation

• Due to the missing values, the joint cell probabilities $p(z)$ are unknown.
• We use a modified expectation maximization (EM) algorithm for $p(z)$.

**EM Algorithm [E-step]**

$$\hat{\pi}_{b^*|a}^{(t)} = \hat{\pi}^{(t)} \left(z_{i,obs}, z_{i,mis} = z_{i,mis}^{(h)} \right) / \sum_{h=1}^{M_i} \hat{\pi}^{(t)} \left(z_{i,obs}, z_{i,mis}^{(h)} \right)$$

**EM Algorithm [M-step]**

$$\hat{\pi}^{(t+1)}(z_{obs}, z_{mis}^*) = \frac{\sum_{i=1}^{n} \sum_{h=1}^{M_i} w_i \hat{\pi}_{b^*|a}^{(t)} I \left(z_{i,obs} = z_{obs}, z_{i,mis}^{(h)} = z_{mis}^* \right)}{\sum_{i=1}^{n} w_i}$$

• $(t)$ denotes iteration step; $M_i$ denotes the number of total possible donors
• $\hat{\pi}_{b^*|a}^{(t)}$ denotes the conditional probability where $b^*|a = z_{i,mis}^{(h)} | z_{i,obs}$
• $I(True) = 1; I(False) = 0$
• $w_i$ denotes the inverse of the sample selection probability of unit $i$. 
EM Algorithm for Cell Probability Estimate

- The normality is confirmed as shown below.
- In theory, there is no restriction to the total variables (i.e., big \( p \) problem) except for the computational burden.

```r
> datz=cdaty$cell
> jcp=FHDI_CellProb(datz)
> jcp$cellpr

<table>
<thead>
<tr>
<th>1111</th>
<th>1123</th>
<th>1212</th>
<th>1221</th>
<th>2223</th>
<th>2322</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.181</td>
<td>0.0547</td>
<td>0.1269</td>
<td>0.07786</td>
<td>0.17388</td>
<td>0.08263</td>
</tr>
<tr>
<td>3133</td>
<td>3232</td>
<td>3233</td>
<td>3331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0217</td>
<td>0.1036</td>
<td>0.08871</td>
<td>0.08879</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> sum(jcp$cellpr)
[1] 1

← Joint probability from a modified EM
← Normality of the joint probabilities
Fractional Hot Deck Imputation: Theory

Fully Efficient Fractional Imputation (FEFI) Estimator

- FEFI uses all possible donors for a missing cell

- \( \hat{Y}_{p,FEFI} \), the FEFI estimator of \( Y_p = \sum_{i=1}^{N} Y_{pi} \) is given by

\[
\hat{Y}_{p,FEFI} = \sum_{i \in A} w_i \left\{ \delta_{pi} Y_{pi} + (1 - \delta_{pi}) \sum_{j \in A} w_{ij,FEFI} Y_{pj} \right\}
\]

where

\[
w_{ij,FEFI}^* = \hat{\pi}_{z_{i,mis}^* | z_{i,obs}} \frac{w_j I \left\{ \left( z_{i,obs}, z_{i,mis}^* \right) = \left( z_{j,obs}^{(i)}, z_{j,mis}^{(i)} \right) \right\}}{\sum_{l \in A_R} w_l I \left\{ \left( z_{i,obs}, z_{i,mis}^* \right) = \left( z_{l,obs}^{(i)}, z_{l,mis}^{(i)} \right) \right\}}
\]

- \( w_{ij,FEFI}^* \) is the fractional weight of \( j \)-th donor for \( i \)-th recipient
- \( \left( z_{l,obs}^{(i)}, z_{l,mis}^{(i)} \right) \) denotes the vector of unit \( l \) corresponding to the observed and missing part of \( i \)-th recipient
- \( \sum_{j \in A} w_{ij,FEFI}^* = 1 \)
Fractional Hot Deck Imputation: Theory

Fractional Hot Deck Imputation (FHDl) Estimator

• FHDl seeks to approximate FEFI by selecting $M$ possible donors for a missing cell.

• To select $M$ donors, we used the probability proportional to size (PPS) sampling: i.e., a random selection proportional to $w_{ij,FEFI}^*$

• $\hat{Y}_{p,FHDI}$, the FHDl estimator of $Y_p = \sum_{i=1}^{N} y_{pi}$ is given by

$$\hat{Y}_{p,FHDI} = \sum_{i \in A} w_i \left\{ \delta_{pi} y_{pi} + (1 - \delta_{pi}) \sum_{j=1}^{M} w_{ij}^* y_{pi}^{*(j)} \right\}$$

• $w_{ij}^* = 1/M$
• $y_{pi}^{*(j)}$ denotes $j$-th donor for the $i$-th recipient $y_{pi}$
Example of Fractional Hot Deck Imputation

FEFI Imputation

```
> FEFI=FHDI_Driver(daty,s_op_imputation="FEFI",i_op_variance=1,k=3)
> dim(FEFI$fimp.data)
[1] 330  8
>
> FEFI$fimp.data[1:13,]
   ID  FID WT FWT    y1       y2      y3       y4
[1,]  1   1 1 0.5000000 1.47963286 2.150860 2.881646 1.8942118
[2,]  1   2 1 0.5000000 1.47963286 2.150860 2.493438 1.8942118
[3,]  2   1 1 0.2000000 -0.09087472 1.141496 1.602530 -1.0369469
[4,]  2   2 1 0.2000000 -1.6706193 1.141496 1.602530 -1.0369469
[5,]  2   3 1 0.2000000 -0.39302750 1.141496 1.602530 -1.0369469
[6,]  2   4 1 0.2000000  0.97612864 1.141496 1.602530 -1.0369469
[7,]  2   5 1 0.2000000  0.21467221 1.141496 1.602530 -1.0369469
[8,]  3   1 1 0.1666667  0.70870936 1.885673 1.250689 0.7770526
[9,]  3   2 1 0.1666667  0.70870936 1.885673 1.250689 1.2839115
[10,] 3   3 1 0.1666667  0.70870936 1.885673 1.250689 0.6309413
[11,] 3   4 1 0.1666667  0.70870936 1.885673 1.250689 0.3232018
[12,] 3   5 1 0.1666667  0.70870936 1.885673 1.250689 0.5848844
[13,] 3   6 1 0.1666667  0.70870936 1.885673 1.250689 1.0342970
```

← Final FEFI-imputed data set (c.f. $n=100$) with fractional weights (FWT) and all possible donors (in color)
FHDI Imputation (e.g., with donors $M = 5$)

- Substantial sample size reduction can be achieved by FHDI, which is critical for large/big data applications.

Example of Fractional Hot Deck Imputation

```r
> FHDI=FHDI_Driver(daty,s_op_imputation="FHDI",M=5,i_op_variance=1,k=
> dim(FHDI$fimp.data)
[1] 285 8
> FHDI$fimp.data[1:12,]
  ID  FID WT FWT  y1      y2  y3      y4
[1,] 1  1 1 0.500000 1.47963286 2.150860 2.881646 1.8942118
[2,] 1  2 1 0.500000 1.47963286 2.150860 2.493438 1.8942118
[3,] 2  1 1 0.200000 -0.09087472 1.141496 1.602530 -1.0369469
[4,] 2  2 1 0.200000 -1.67006193 1.141496 1.602530 -1.0369469
[5,] 2  3 1 0.200000 -0.39302750 1.141496 1.602530 -1.0369469
[6,] 2  4 1 0.200000  0.97612864 1.141496 1.602530 -1.0369469
[7,] 2  5 1 0.200000  0.21467221 1.141496 1.602530 -1.0369469
[8,] 3  1 1 0.200000  0.70870936 1.885673 1.250689  0.7770526
[9,] 3  2 1 0.200000  0.70870936 1.885673 1.250689  1.2839115
[10,] 3  3 1 0.200000  0.70870936 1.885673 1.250689  0.6309413
[11,] 3  4 1 0.200000  0.70870936 1.885673 1.250689  0.3232018
[12,] 3  5 1 0.200000  0.70870936 1.885673 1.250689  1.0342970
```

← Final FHDI-imputed data set (c.f. $n=100$) with fractional weights (FWT) and selected donors (in color)
Fractional Hot Deck Imputation: Theory

Variance Estimator for the FEFI and FHDI estimators

- We used the **Jackknife method** consisting of two steps:
- (1) delete one unit $k$ and (2) calculate $w_{ij}^{*(k)}$ ($k = 1, \ldots, n$) by

$$w_{ij}^{*(k)} = \begin{cases} 
    w_{ij}^* - w_{ij,\text{FEFI}}^* & \text{if } (j = r, i \in A_{Mg}, k \in A_{Rg}) \\
    w_{ij}^* + w_{ir,\text{FEFI}}^* \frac{w_{ij,\text{FEFI}}^*}{\sum_{j \neq r} w_{ij,\text{FEFI}}^*} & \text{if } (j \neq r, i \in A_{Mg}, k \in A_{Rg}) \\
    w_{ij}^* & \text{if } k \in A_{Mg}
\end{cases}$$

- $r$ stands for the nearest donor for $k$ (for multivariate case, we use the Mahalanobis Distance $\equiv \sqrt{(x - \mu)^T S (x - \mu)}$ as a distance measure)

**Jackknife variance estimator** is given by

$$\hat{V}(\hat{Y}_{p,\text{FHDI}}) = \sum_{k=1}^{n} c_k \left( \hat{Y}_{\text{FHDI}}^{(k)} - \hat{Y}_{\text{FHDI}} \right)^2$$

where $\hat{Y}_{p,\text{FHDI}}^{(k)} = \sum_{i \in A} w_i \left\{ \delta_{pi} y_{pi} + (1 - \delta_{pi}) \sum_{j=1}^{M} w_{ij}^{*(k)} y_{pi}^{(j)} \right\}$
Example of Fractional Hot Deck Imputation

Regression (y1~y2) coefficient estimates with standard errors (SEs)
- Superiority of FHDI and FEFI over Naïve method in terms of SE.
- FHDI approximates well FEFI

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Intercept</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve (with mean)</td>
<td>-0.074</td>
<td>0.305</td>
</tr>
<tr>
<td>FHDI</td>
<td>0.023</td>
<td>0.252</td>
</tr>
<tr>
<td>FEFI</td>
<td>0.035</td>
<td>0.251</td>
</tr>
<tr>
<td>True</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Slope</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve (with mean)</td>
<td>0.588</td>
<td>0.142</td>
</tr>
<tr>
<td>FHDI</td>
<td>0.472</td>
<td>0.095</td>
</tr>
<tr>
<td>FEFI</td>
<td>0.466</td>
<td>0.094</td>
</tr>
<tr>
<td>True</td>
<td>0.5</td>
<td>-</td>
</tr>
</tbody>
</table>
Fractional Hot Deck Imputation: Asymptotic Properties

Asymptotic Properties of the FHDI estimator (see details in [4,6])

\[
\hat{Y}_{p,FHDI} = \hat{Y}_{p,FHDI} + o_p \left( \frac{N}{\sqrt{n}} \right); \\
E(\hat{Y}_{p,FHDI} - Y_p) = 0
\]

where (\overset{\sim}{}) represents Taylor expansion term

\[
\hat{Y}_{p,FHDI} = \hat{Y}_{p,FEFI} + \sum_{g=1}^{G} \sum_{i \in A_{Mg}} w_i (\bar{y}_{pi} - \hat{\mu}_{pg}),
\]

\[
\hat{Y}_{p,FEFI} = \sum_{g=1}^{G} \sum_{i \in A_{Mg}} w_i \left( \mu_{pg} + R_g^{-1} \delta_i (y_{pi} - \mu_{pg}) \right) = \hat{Y}_{p,FEFI} + o_p (N/\sqrt{n}).
\]

- \( A_M = A_{M1} \cup \cdots \cup A_{Mg} \); \( A_{Mg} \) is a sample subgroup (c.f., \( U_g \) in population)
- \( \bar{y}_{pi} = M^{-1} \sum_{j=1}^{M} y_{pi}^{(j)} \); \( R_g = \Sigma_{i \in U_g} \delta_i / \Sigma_{i \in U_g} 1 \)
- \( \hat{\mu}_{pg} \) = the mean of \( y_{pi} \)’s that are associated with \( A_{Mg} \)
- \( \mu_{pg} \) = the mean of \( y_{pi} \)’s that are associated with \( g \)-th population subgroup
Fractional Hot Deck Imputation: Asymptotic Properties

Asymptotic Properties of the FHDI estimator (cont’d)

\[ V(\tilde{Y}_{p,FHDI}) = V(\tilde{Y}_{p,FEFI}) + E \left\{ \sum_{g=1}^{G} \sum_{i \in A_{Mg}} w_i^2 V(\bar{y}_{pi} - \hat{\mu}_{pg} | A_g) \right\} \]

where

\[ V(\tilde{Y}_{p,FEFI}) = V \left( \sum_{g=1}^{G} \sum_{i \in A_{Mg}} w_i \mu_{pg} \right) + E \left\{ \sum_{g=1}^{G} \sum_{i \in A_g} w_i^2 R_g^{-2} \delta_i (y_{pi} - \mu_{pg})^2 \right\} \]

• Thus, for large/big data FHDI can asymptotically replace FEFI
• FHDI has a strong computational efficiency compared to FEFI
• This is the starting point of the Parallel FHDI for large/big data curing
Example of Fractional Hot Deck Imputation

Validation with Practical Data

\[ E(Y_i)/E(Y_{i_{FHDI}}) \]

<table>
<thead>
<tr>
<th>Some Attribute</th>
<th>10% Missing</th>
<th>30% Missing</th>
<th>50% Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>1.0023</td>
<td>0.9993</td>
<td>1.0116</td>
</tr>
<tr>
<td>y3</td>
<td>1.0000</td>
<td>1.0002</td>
<td>1.0002</td>
</tr>
<tr>
<td>y5</td>
<td>1.0000</td>
<td>0.9998</td>
<td>0.9997</td>
</tr>
<tr>
<td>y7</td>
<td>1.0000</td>
<td>0.9998</td>
<td>1.0000</td>
</tr>
<tr>
<td>y9</td>
<td>0.9994</td>
<td>0.9964</td>
<td>0.9945</td>
</tr>
<tr>
<td>y11</td>
<td>1.0006</td>
<td>1.0022</td>
<td>1.0020</td>
</tr>
<tr>
<td>y14</td>
<td>1.0021</td>
<td>1.0015</td>
<td>1.0002</td>
</tr>
</tbody>
</table>

U.S. Appliance energy data set used

- Total instances, \( n = 19,735 \),
- Total variables, \( p = 14 \),
- Energy uses in residential population in the U.S. by courtesy of Dr. Cetin.


- \[ E(Y_i)/E(Y_{i_{FHDI}}) = 1 \] means the ideal imputation
- \( E(Y_i) \) is the mean of \( Y_i \) in the full data and
- \( E(Y_{i_{FHDI}}) \) is the mean of the cured \( Y_i \) in the randomly deleted missing data
Motivation

- 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
- Strong need for general-purpose and assumption-free big data (big-\(n\) and/or big-\(p\)) imputation tools
- The positive impact of FHDI on learning and prediction
- Asymptotic properties of FHDI support the inheritance of strengths of FEFI
Parallel Fractional Hot Deck Imputation (P-FHDI)

Types of large/big incomplete datasets

- **big-n data**: $n \gg p$
  - Tackled by P-FHDI ver. 1.0 (Cho et al. *IEEE, TKDE*, 2020 [12])

- **big-p data**: $n \leq p$

- **Ultra data**: $n$ and $p$ are both large
  - On-going research

$n$: number of instances
$p$: number of variables
$\eta$: missing rate
Key Procedures of P-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

• **Variance estimation** (Process 4)
  Jackknife method for the variance of each variable
Algorithm-Oriented Parallel Computing

Figure: (a) Internal parallelization within the unbreakable implicit loop (e.g. EM algorithm); (b) Typical divide and conquer for embarrassing parallelizable explicit loop.
Work distribution scheme

Figure: An irregular area enclosed by dashed line is the work domain. Suppose we need parallelize it to three processors.

<table>
<thead>
<tr>
<th></th>
<th>Uniform job distribution</th>
<th>Cyclic job distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantage</strong></td>
<td>Easy to implement</td>
<td>Hard to implement</td>
</tr>
<tr>
<td><strong>Disadvantage</strong></td>
<td>Workload imbalance</td>
<td>Efficient workload balance</td>
</tr>
</tbody>
</table>
P-FHDI: Time Cost Model

**Total running time** (see details in [12]):

\[ T(Q) \approx \frac{\alpha'}{Q} + \beta' Q \]

where \( Q \) is the number of processors, \( \alpha' = \alpha \psi_1 O(n^2) + \alpha O(n^3) \), and \( \beta' = \beta \psi_1 O(n) \). The iterations in cell construction and cell probability are \( \psi_1 \) and \( \psi_2 \), respectively. Computational cost and communication transfer cost per element are \( \alpha \) and \( \beta \), respectively.

**Scalability:**

\[ \frac{T(Q)}{T(c_p Q)} = c_p \times \frac{\alpha' + Q^2 \beta'}{\alpha' + c_p^2 Q^2 \beta'} \]

where \( c_p \in N^+ \) is the number of nodes.
Scalability Result of a Big-\( n \) Data Set

Summary of big-\( n \) synthetic input datasets:
- 1 million rows and 4 columns with 15\% missing values
- 1 million rows and 4 columns with 25\% missing values
- 1 million rows and 4 columns with 35\% missing values
Challenges to Donor Selection for Big-p Data

When there are too many variables (extremely high dimensionality), there are nearly infinitely many possible missing patterns.

For a given missing pattern, how can we find enough donors?

<table>
<thead>
<tr>
<th>A missing pattern</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>3</th>
</tr>
</thead>
</table>

3 categories
0=N/A

\[ p = 10000, 100K, \text{or } 1 \text{ M} \]

Searching vast space is not possible, and thus we need a "variable reduction" scheme to determine donors.
Popular Variable Reduction Methods

I. Least absolute shrinkage selection operator (LASSO) [7]
\[
\argmin_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} X_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|
\]

II. The adaptive Lasso (Ada. LASSO) [8]
\[
\argmin_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} X_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{\hat{\beta}_j}
\]

III. Ridge [9]
\[
\argmin_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} X_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2
\]

IV. Smoothly clipped absolute deviation (SCAD) [10]

Penalty: \[
\lambda \left\{ I(|\beta_j| \leq \lambda) + \frac{(a\lambda - |\beta_j|)I(|\beta_j| > \lambda)}{(a-1)\lambda} \right\}
\]

V. Sure independence screening (SIS) [11]
Sure Independence Screening (SIS)

Setup

• Consider a linear regression model

\[ y = \beta X + e \]

where \( y = (y_1, y_2, \ldots, y_n)^T \) is a vector of responses, \( X = (X_1, X_2, \ldots, X_p) \) is an \( n \times p \) random design matrix with \( p \) covariates, and \( e \) is a \( n \)-dimensional error vector.

• Let \( M^* = \{1 \leq i \leq p; \beta_i \neq 0\} \) be the true model. The covariates \( X_i \) with \( \beta_i \neq 0 \) are so-called important variables, otherwise as noise variables.

Two-step method

1) Screening step: find

\[ \hat{M} = \{1 \leq j \leq p; r_j^2 \text{ is among the top of largest ones}\} \]

where \( r_j = corr(X_j, y) \) is the sample correlation between \( X_j \) and \( y \)

2) Selection Step: Using the covariates in \( \hat{M} \), apply a penalized regression method to obtain the best model.
Sure screening property for SIS

• Under fairly general conditions [11],
  \[ P\{M^* \subset \hat{M}\} \to 1 \]
as \( n \to \infty \), with a relatively small size of \( \hat{M} \).

• The probability of the true model \( M^* \) among the built models \( M \) in
  FHDV using screening step is assumed to be
  \[ P\{M^* \subset M\} \to 1 \]
as \( n \to \infty \).
SIS for Donor Selection of P-FHDI

Schematic of Donor Selection by SIS for P-FHDI

Donor selection is done within the selected variables of

- **Intersection:** \( \{y_{12}, y_{100}, \ldots\} \)
- **Union:** \( \{y_9, y_{10}, y_{12}, \ldots\} \)
- **Global Ranking:** \( \{y_i; \text{Sorted by global correlation scores}\} \)

A missing pattern:

by correlation of \( y_3 \)

\( \{y_{10}, y_{12}, y_{100}, y_{201}, y_{310}, \ldots\} \)

by correlation of \( y_4 \)

\( \{y_{9}, y_{12}, y_{100}, y_{301}, y_{510}, \ldots\} \)
EM for the Selected Variables by SIS

Setup
For each $Y_j$ that is missing, let $M_j$ be the selected covariate set for $Y_j$. Let $X^* = \bigcup_{j=1}^{w} M_j$ (Union) or $\bigcap_{j=1}^{w} M_j$ (Intersection) or selected covariates based on (Global ranking). We assume that $X^* \subset X$. Then decompose $Y = (y_{mis}, y_{obs})$ and support $C = \{c | c_{obs} = y_{obs}\}$.

Modified EM Algorithm

$$
\pi_c(X^*) = P(Y = c | X^*)
$$

be the conditional probability of $y_{mis} = c$ given $X^*$ and $y_{obs} = c_{obs}$. Note that $\sum_{c \in C} \pi_c(X^*) = 1$.

To estimate the joint probability for $P(Y_1, \ldots, Y_p | X^*)$, the E step is

$$
p_i^{(t)}(c) = I(y_{i,obs} = c_{i,obs}) \cdot P^{(t)}(y_{i,mis} = c_{i,mis} | X_i^*; y_{i,obs} = c_{i,obs})
$$
EM for the Selected Variables by SIS (cont’d)

where \( P^{(t)}(y_{i,\text{mis}} = c_{i,\text{mis}} | \mathbf{X}_i^*, y_{i,\text{obs}} = c_{i,\text{obs}}) \) is given as

\[
P^{(t)}(y_{i,\text{mis}} = c_{i,\text{mis}} | \mathbf{X}_i^*, y_{i,\text{obs}}) = \frac{\pi_c^{(t)}(\mathbf{X}_i^*)}{\sum_{c \in C_i} \pi_c^{(t)}(\mathbf{X}_i^*)}
\]

Here, \( C_i = \{c \in C\} \) is the set of possible value given \( y_{i,\text{obs}} = c_{i,\text{obs}} \).

The M step is to update the conditional probability as

\[
\pi_c^{(t+1)}(\mathbf{X}^*) = \frac{\sum_{i=1}^n I(\mathbf{X}_i^* = \mathbf{X}^*) p_i^{(t)}(c)}{\sum_{c \in C} \sum_{i=1}^n I(\mathbf{X}_i^* = \mathbf{X}^*) p_i^{(t)}(c)}
\]
Example Scalability for a Big-$p$ data set

Figure: Speedup of the P-FHDI with an extremely high-dimensional dataset $U(1000, 10000, 0.3)$: 1,000 instances and 10,000 variables with 30% missingness. Note that we adopt 3 selected variables using SIS. In selected variables space via SIS, perform K-nearest neighbors (KNN) searching in a Euclidean sense.

(a) big-$p$ with SIS only

(b) big-$p$ with SIS&KNN
Available Example Datasets for P-FHDI

- Following example datasets are available (https://sites.google.com/site/ichoddcse2017/home/type-of-trainings/parallel-fhdi-p-fhdi-1)

<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</td>
<td>Hybrid</td>
<td>U(41757, 4, 0.1)</td>
</tr>
<tr>
<td>Nursery</td>
<td>Categorical</td>
<td>U(12960, 5, 0.3)</td>
</tr>
<tr>
<td>Synthetic data 3</td>
<td>Continuous</td>
<td>U(15000, 12, 0.15)</td>
</tr>
<tr>
<td>Synthetic data 4</td>
<td>Continuous</td>
<td>U(15000, 16, 0.15)</td>
</tr>
<tr>
<td>Synthetic data 5</td>
<td>Continuous</td>
<td>U(15000, 100, 0.15)</td>
</tr>
<tr>
<td>Synthetic data 6</td>
<td>Continuous</td>
<td>U(1000, 100, 0.3)</td>
</tr>
<tr>
<td>Synthetic data 7</td>
<td>Continuous</td>
<td>U(1000, 1000, 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>

- Please refer to (Cho et al. IEEE, TKDE, 2020 [12]) for more details
- Source codes of parallel FHDI are available and executable on local HPC or NSF Cloud Computing (e.g. NSF XSEDE).
Summary

• The P-FHDI inherits all strengths of the general-purpose, assumption-free FHDI
• The P-FHDI shows a favorable linear speedup when it is applied to big datasets with up to millions of instances
• The SIS embedded in P-FHDI holds the linear speedup still when it is applied to big datasets up to 10,000 variables
• The P-FHDI 1.0 is now publicly available
• Researchers in broad engineering and science can cure general, large/big data sets with ease

Ongoing research

• Pursuing ultra data (Big-n & Big-p), more systematic variable reduction for donor selection in P-FHDI (e.g. graphical lasso)
• Develop theories for P-FHDI with high-dimensional categorical data
Reference


**Program and data link:**
https://sites.google.com/site/ichoddccse2017/home/type-of-trainings/parallel-fhdi-p-fhdi-1
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
Note on Difficulty in General Use of Multiple Imputation

• In MI, the Rubin’s formula is based on the decomposition of

\[ V(\hat{\theta}_{MI}) = V(\hat{\theta}_n) + V(\hat{\theta}_{MI} - \hat{\theta}_I) \]

within imputation variance  
between imputation variance

\[ \hat{\theta}_n = \text{complete-sample estimator of } \theta; \]

But in general, we have

\[ V(\hat{\theta}_{MI}) = V(\hat{\theta}_n) + V(\hat{\theta}_{MI} - \hat{\theta}_n) + 2\text{Cov}(\hat{\theta}_{MI} - \hat{\theta}_n, \hat{\theta}_n) \]

• For Rubin’s variance estimator to be valid, we must have

\[ 2\text{Cov}(\hat{\theta}_{MI} - \hat{\theta}_n, \hat{\theta}_n) = 0 \]

• This condition is so-called “congeniality” of \( \hat{\theta}_n \) according to Meng (1994)
• When \( \hat{\theta}_n \) is the MLE of \( \theta \) (i.e. “self-efficient” estimator), congeniality is met.
• But, in practical applications, satisfying this condition is intractable or ignored
Notation for the iteration of operations

A new mathematical symbol ’\(\mathbb{II}\)’ denotes a loop which repeats a sequence of the same operation \(S(x)\) with discrete input augments within a fixed range.

- The following is the simplest proposal of the loop symbol of an operation \(S\):
  \[
  \mathbb{II}_{i=a}^{b} \ S(x_i) = \{S(x_a), S(x_{a+1}), ..., S(x_{b-1}), S(x_b)\}
  \]
  where \(i = a, ..., b\).

- Furthermore,
  \[
  \mathbb{II}_{i=a}^{b} \mathbb{II}_{j=c}^{d} \ S(x_i) = \begin{bmatrix}
  S(x_{ac}) & S(x_{a(c+1)}) & ... & S(x_{ad}) \\
  S(x_{(a+1)c}) & S(x_{(a+1)(c+1)}) & ... & S(x_{(a+1)d}) \\
  ... & ... & ... & ... \\
  S(x_{bc}) & S(x_{b(c+1)}) & ... & S(x_{bd})
  \end{bmatrix}
  \]
  where \(i = a, ..., b\) and \(j = c, ..., d\).
1. Parallel Cell Construction

- Generate a set of unique missing patterns at processor $q$:
  \[
  \tilde{z}_M^{(q)} = \bigcup_{i=s}^e z_{Mi} I(\|z_{Mi}\| \leq \|z_M\|)
  \]

- Generate a set of unique observed patterns at processor $q$:
  \[
  \tilde{z}_R^{(q)} = \bigcup_{i=s}^e z_{Ri} I(\|z_{Ri}\| \leq \|z_R\|)
  \]

- Generate a set of donor numbers at processor $q$:
  \[
  M^{(q)} = \bigcup_{i=s}^e \left\{ \sum_{j=1}^{\tilde{n}_R} I(\|z_{i,obs}\| = \|z_{j,obs}\|) \right\}
  \]

- Generate an actual index set of donors for all recipients at processor $q$:
  \[
  L^{(q)} = \bigcup_{i=s}^e \bigcup_{j=1}^{\tilde{n}_R} j I(\|z_{i,obs}\| = \|z_{j,obs}\|)
  \]

Note that $s$ and $e$ are boundary indices of distributed work.
2. Parallel Cell Probability Estimation

• Compute conditional probability of \( H_g \) donors in Estep:

\[
\hat{\pi}(t) = \prod_{h=1}^{H_g} \frac{\hat{p}(t)(z_{g,obs}, z_{i,mis}=z^{*(h)}_{g,mis})}{\sum_{h=1}^{H_g} \hat{p}(t)(z_{g,obs}, z_{i,mis}=z^{*(h)}_{g,mis})}
\]

Let \( A_\pi \) be local index set of conditional probability \( \hat{\pi}(t) \in \mathbb{R}^{n_\pi} \), and \( \hat{A}_\pi \) be sorted index set of \( A_\pi \). The index mapping from \( A_\pi \) to \( \hat{A}_\pi \) at processor \( q \) is

\[
\delta(q) = \prod_{i=s}^{e} \prod_{j=1}^{n_\pi} j I(\hat{A}^{(q)}_{\pi i} = A_{\pi j})
\]

The quantity \( \hat{\pi}(t) \) requires reorganization regrading sorted index mapping \( \delta \) before M step.

• Update the joint probability of \( z^* = (z_{g,obs}, z^{*(h)}_{g,mis}) \) in M step:

\[
\hat{p}(t+1)(z^*) = \left( \sum_{i=1}^{n} w_i \right)^{-1} \sum_{i=1}^{n} w_i \hat{\pi}(t) I(z_{i,obs} = z_{g,obs})
\]
3. Parallel imputation and variance estimation

- Let $\hat{A}$ be local index set of imputed values in size of $n_R + \sum_{i=1}^{nM} M_i$, and $\tilde{A}$ be sorted index set of $\hat{A}$. The index mapping from $\tilde{A}$ to $\tilde{\hat{A}}$ at processor $q$ is

$$\varphi^{(q)} = \prod_{i=s}^{e} \prod_{j=1}^{nR+\sum_{i=1}^{nM} M_i} jI(\tilde{\hat{A}}^{(q)}_i = \tilde{A}_j)$$

Parallel variance estimation

- Compute cell probability for unique patterns at processor $q$:

$$\hat{P}^{(q)}(\tilde{Z}_M) = \prod_{j=s}^{e} \prod_{i=1}^{n} w_i^{-1} \sum_{i=1}^{n} w_i \hat{\pi}_j^{(t)} I(z_{i,obs} = z_{j,obs})$$

- Compute replicate estimator at processor $q$:

$$\hat{Y}_{FHD}^{K,q} = \prod_{i=s}^{e} \prod_{i=1}^{p} \hat{Y}_{i,FHD}^{K}$$

$$= \prod_{i=s}^{e} \sum_{i=1}^{p} w_i^k \left\{ \delta_{il} y_{il} + (1 - \delta_{il}) \sum_{j=1}^{M} w_i^{*(k)} y_{ij}^{*(j)} \right\}$$
**SIS for Donor Selection of P-FHDI**

**Setup**

Suppose one selects \( \nu \) variables for a recipient. Let \( \mathbf{X} = \{X_1, \ldots, X_q\} \) be always observed and \( \mathbf{Y} = \{Y_1, \ldots, Y_w\} \) be subject to missing. Consider \( r_k = (r_k^1, r_k^2, \ldots, r_k^q) \) be a vector of sample correlations between \( X_i, i = 1, \ldots, q \), and \( Y_k \).

**Variable reduction with intersection or union of simple correlation:**

1. Compute correlation vectors \( r_k \) where \( k \in \{1, \ldots, w\} \) and sort them in descending order.
2. Define sub-covariate set \( M_k \) for imputing \( Y_k \), \( k \in \{1, \ldots, w\} \) such that
   \[
   M_k = \{1 \leq i \leq q; |r_k^i| \text{ is among the top of largest } \nu, \text{ where } r_k^i \in r_k\}.
   \]
3. Implicitly assume that
   \[
P(Y_1, \ldots, Y_w | X_1, \ldots, X_q) = P(Y_1, \ldots, Y_w | X^*)
   \]
   where \( X^* \) is the intersection/union of covariates such that \( M = \bigcap_{k=1}^w M_k \) or \( M = \bigcup_{k=1}^w M_k \).
4. If size \( |M| = \nu \), then stop. Otherwise, repeat step (2) and (3) by setting \( \nu = \nu + 1 \) until one obtain \( \nu \) selected variables.
Variable reduction with **global ranking** of simple correlation:

1. Compute correlation vectors $\mathbf{r}_k$ where $k \in \{1, \ldots, w\}$. $\mathbf{r}_k = (r_{k1}, r_{k2}, \ldots, r_{kq})$ be a vector of sample correlations between $X_i, i = 1, \ldots, q$, and $Y_k$.

2. Define $\mathbf{r}^* = (r_1^*, \ldots, r_i^*, \ldots, r_q^*)$, where $r_i^* = \max\{r_{i1}, r_{i2}, \ldots, r_{ik}\}$.

3. Define selected covariate set $\mathbf{X}^*$ for imputing $\mathbf{Y}_k, k \in \{1, 2, \ldots, w\}$ such that $\mathbf{X}^* = \{1 \leq i \leq q; |r_i^*| \text{ is among the top of largest } v, \text{ where } r_i^* \in \mathbf{r}^*\}$. 

---

**SIS for Donor Selection of P-FHDI**