

# Multivariate imputation of qualitative missing data using Bayesian networks

V. Romero, A. Salmerón

Departamento de Estadística y Matemática Aplicada  
Universidad de Almería

# Approaches to imputation

---

- Discard the records with missing values.

# Approaches to imputation

---

- Discard the records with missing values.

Missing completely at random (MCAR).

# Approaches to imputation

---

- Discard the records with missing values.  
Missing completely at random (MCAR).
- Hot deck methods.

# Approaches to imputation

---

- Discard the records with missing values.
  - Missing completely at random (MCAR).
- Hot deck methods.
- Missing at random assumption (MAR).

# Approaches to imputation

---

- Discard the records with missing values.  
Missing completely at random (MCAR).
- Hot deck methods.
- Missing at random assumption (MAR).  
Regression models.

# Approaches to imputation

---

- Discard the records with missing values.
  - Missing completely at random (MCAR).
- Hot deck methods.
- Missing at random assumption (MAR).
  - Regression models.
  - Classification trees.

# Approaches to imputation

---

- Discard the records with missing values.
  - Missing completely at random (MCAR).
- Hot deck methods.
- Missing at random assumption (MAR).
  - Regression models.
  - Classification trees.
  - Iterative algorithms (EM, Gibbs sampling).

# Imputation of missing values

---

- A sample  $S = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}\}$  of qualitative random vectors.

# Imputation of missing values

---

- A sample  $\mathbf{S} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}\}$  of qualitative random vectors.
- $\mathbf{X}^{(i)} = (\mathbf{X}_m^{(i)}, \mathbf{X}_o^{(i)})$ .

# Imputation of missing values

---

- A sample  $\mathbf{S} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}\}$  of qualitative random vectors.
- $\mathbf{X}^{(i)} = (\mathbf{X}_m^{(i)}, \mathbf{X}_o^{(i)})$ .
- An imputation procedure involves:

# Imputation of missing values

---

- A sample  $\mathbf{S} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}\}$  of qualitative random vectors.
- $\mathbf{X}^{(i)} = (\mathbf{X}_m^{(i)}, \mathbf{X}_o^{(i)})$ .
- An imputation procedure involves:  
**Learning a model for the missing values.**

# Imputation of missing values

---

- A sample  $\mathbf{S} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}\}$  of qualitative random vectors.
- $\mathbf{X}^{(i)} = (\mathbf{X}_m^{(i)}, \mathbf{X}_o^{(i)})$ .
- An imputation procedure involves:
  - Learning a model for the missing values.
  - Predicting a value for the missing cells.

# Imputation using Bayesian networks

---

# Imputation using Bayesian networks

---

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .

# Imputation using Bayesian networks

---

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$

# Imputation using Bayesian networks

---

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$

# Imputation using Bayesian networks

---

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .

# Imputation using Bayesian networks

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .
  - (b) Learn a Bayesian network  $G$  from  $\mathbf{S}_l$ .

# Imputation using Bayesian networks

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .
  - (b) Learn a Bayesian network  $G$  from  $\mathbf{S}_l$ .
  - (c) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_m$

# Imputation using Bayesian networks

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .
  - (b) Learn a Bayesian network  $G$  from  $\mathbf{S}_l$ .
  - (c) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_m$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .

# Imputation using Bayesian networks

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .
  - (b) Learn a Bayesian network  $G$  from  $\mathbf{S}_l$ .
  - (c) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_m$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .
    - ii. Replace  $\mathbf{x}_m$  in  $\mathbf{x}$ .

# Imputation using Bayesian networks

1.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains missing values}\}$  ;  
 $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
2. FOR  $i := 1$  to  $N$ 
  - (a) IF ( $i = 1$ ),  $\mathbf{S}_l := \mathbf{S}_o$  ELSE  $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$ .
  - (b) Learn a Bayesian network  $G$  from  $\mathbf{S}_l$ .
  - (c) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_m$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .
    - ii. Replace  $\mathbf{x}_m$  in  $\mathbf{x}$ .
3.  $\mathbf{S}' := \mathbf{S}_o \cup \mathbf{S}_m$ .

# Incremental imputation

---

# Incremental imputation

---

1.  $k$  := number of vars. in each item.

# Incremental imputation

---

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .

# Incremental imputation

---

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$

# Incremental imputation

---

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .
  - (d) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .
  - (d) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .
  - (d) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .
    - ii. Replace  $\mathbf{x}_m$  in  $\mathbf{x}$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .
  - (d) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .
    - ii. Replace  $\mathbf{x}_m$  in  $\mathbf{x}$ .
  - (e)  $\mathbf{S}_o := \mathbf{S}_o \cup \mathbf{S}_i$ .

# Incremental imputation

1.  $k :=$  number of vars. in each item.
2.  $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x} \text{ contains m.v.}\}$  ;  $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$ .
3. FOR  $i := 1$  to  $k$ 
  - (a)  $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x} \text{ has } i \text{ missing values}\}$ .
  - (b)  $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$ .
  - (c) Learn a Bayesian network  $G$  from  $\mathbf{S}_o$ .
  - (d) For each  $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$ 
    - i. Generate a new value  $\mathbf{x}_m$  for  $\mathbf{X}_m$  from the distribution  $p(\mathbf{x}_m \mid \mathbf{x}_o)$  computed from  $G$ .
    - ii. Replace  $\mathbf{x}_m$  in  $\mathbf{x}$ .
  - (e)  $\mathbf{S}_o := \mathbf{S}_o \cup \mathbf{S}_i$ .
4.  $\mathbf{S}' := \mathbf{S}_o$ .

# Experimental evaluation

---

Three databases:

# Experimental evaluation

---

Three databases:

- **chest**: A database with 8 variables and 2000 registers generated using forward sampling from the chest clinic network.

# Experimental evaluation

---

Three databases:

- **chest**: A database with 8 variables and 2000 registers generated using forward sampling from the chest clinic network.
- **water**: A 32-variable database with 4000 registers generated by forward sampling from the water network borrowed from the Decision Support Systems group at Aalborg University.

# Experimental evaluation

---

Three databases:

- **chest**: A database with 8 variables and 2000 registers generated using forward sampling from the chest clinic network.
- **water**: A 32-variable database with 4000 registers generated by forward sampling from the `water` network borrowed from the Decision Support Systems group at Aalborg University.
- **housing**: A database with 13 variables and 506 registers taken from the UCI Machine Learning Repository.

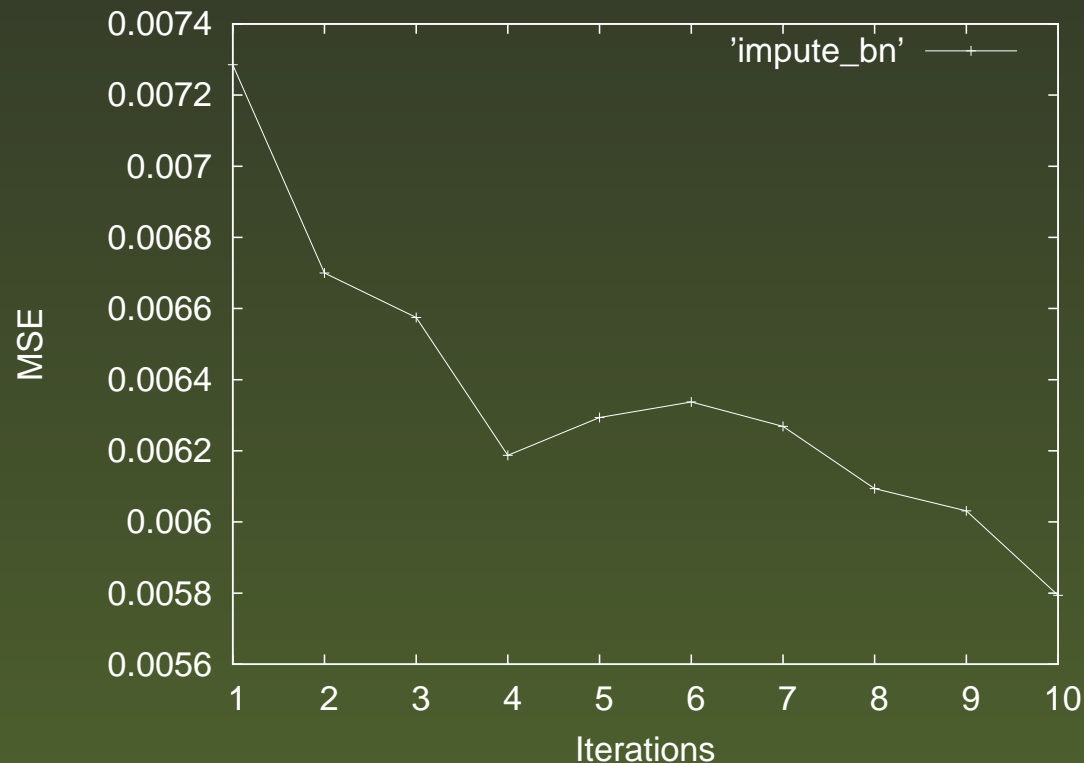
# Experimental evaluation

---

Results of algorithm **IMPUTE\_BN** for database `chest` with 10 iterations.

# Experimental evaluation

Results of algorithm **IMPUTE\_BN** for database **chest** with 10 iterations.



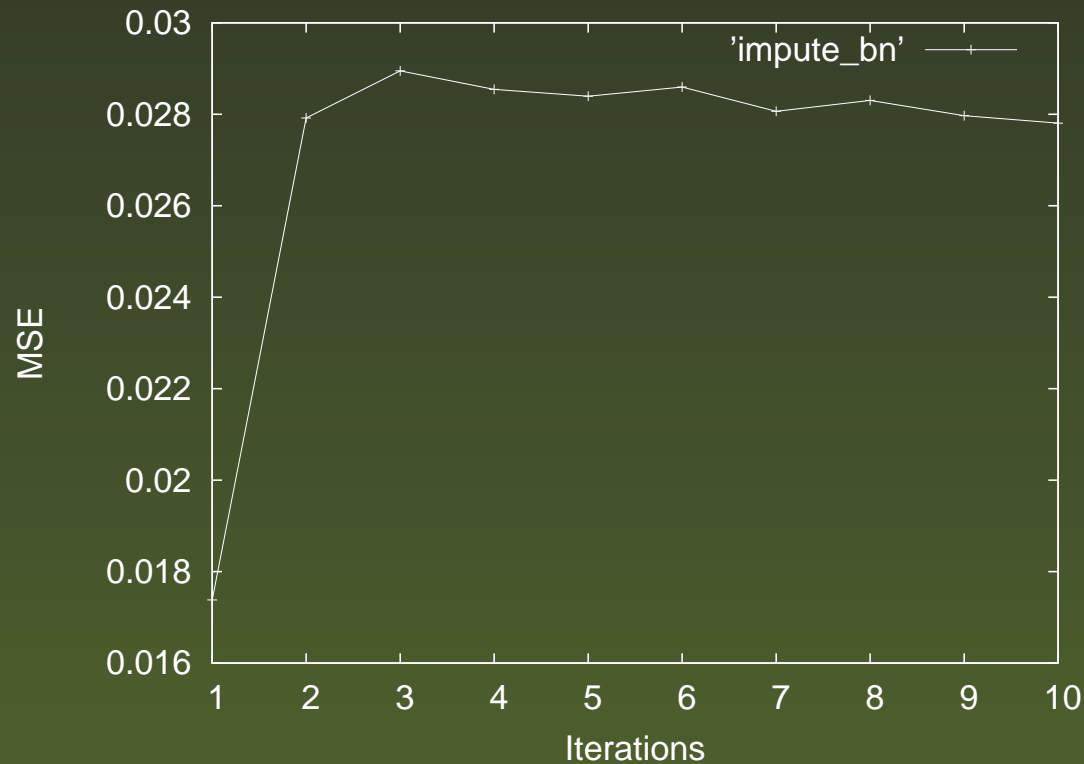
# Experimental evaluation

---

Results of algorithm **IMPUTE\_BN** for database **housing** with 10 iterations.

# Experimental evaluation

Results of algorithm **IMPUTE\_BN** for database **housing** with 10 iterations.



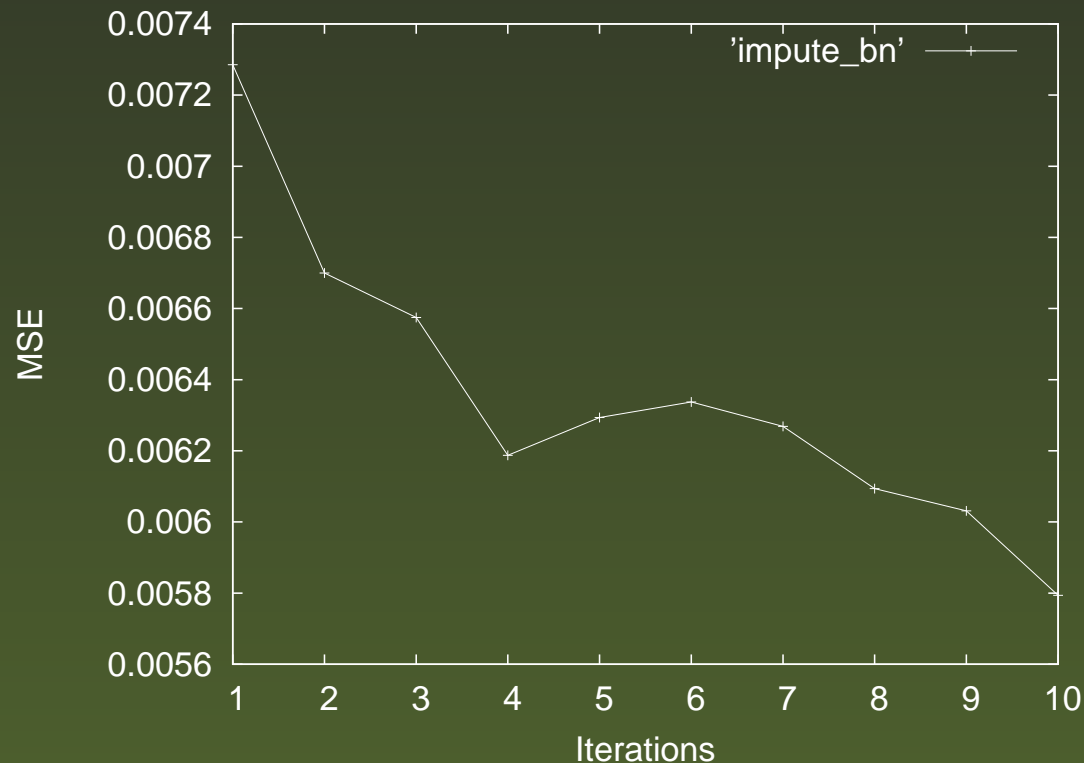
# Experimental evaluation

---

Results of algorithm **IMPUTE\_BN** for database **water** with 10 iterations.

# Experimental evaluation

Results of algorithm **IMPUTE\_BN** for database **water** with 10 iterations.



# Experimental evaluation

---

Classification trees vs. **INCR\_IMPUTE\_BN** and **IMPUTE\_BN** after 10 iterations.

# Experimental evaluation

Classification trees vs. **INCR\_IMPUTE\_BN** and **IMPUTE\_BN** after 10 iterations.

	Classification tree	INCR_IMPUTE_BN	IMPUTE_BN
chest	0.0328	0.0051	0.0058
housing	0.0371	0.0157	0.0278
water	0.0328	0.0051	0.0058