KNOWLEDGE REPRESENTATION AND REASONING HOMEWORK 2: EXPECTATION MAXIMIZATION AND EMBEDDINGS TECHNIQUES

Cătălin-Alexandru Rîpanu IA1-A
Automatic Control and Computers
National University of Science and Technology POLITEHNICA Bucharest
E-mail:catalin.ripanu@upb.ro
Laboratory Assistant: Mihai Trăscău

10 January 2025

1 Part I

1.1 The most important use of the Expectation-Maximization (EM) algorithm is in handling datasets with incomplete or missing data. In this homework, you will be applying EM algorithm for imputing missing values in a dataset.

- Consider a dataset with two variables X and Y, where some values of Y are missing. Explain how the EM algorithm can be used to estimate the mean and variance of Y and the correlation between X and Y. Why is the log-likelihood expected to increase after each iteration?

  The EM algorithm uses two alternating steps:

  Expectation Steps (E-steps): In this stage, we estimate (or "impute") the missing Y values based on our current best guess for the parameters. For each missing Y value, we calculate its conditional expectation as follows:

  Using bivariate normal distributions, the expected value of Y given X is:

  $$\mathbb{E}[Y \mid X] = \mu_Y + \rho \left(\frac{\sigma_Y}{\sigma_X} \right) (X - \mu_X).$$

  This formula demonstrates how we can make an educated approximation regarding missing Y values using the X values we know and our existing parameter estimates.

  Maximization Step (M-step): We can improve our parameter estimations after estimating all observed and imputed Y values.

  - μ_Y: Calculate the mean of all Y values (observed and imputed).
  - σ_Y^2: Find the variance of all Y values.
  - ρ: Find the correlation between X and all Y values.

The approach optimizes the "expected complete-data log-likelihood." During each iteration:

- The E-step uses Jensen's inequality to provide a lower bound on the true log-likelihood.
- The M-step maximizes the lower bound.

This approach ensures that each iteration will either increase the log probability or remain unchanged if we've reached a local maximum.

Imagine ascending a hill in the fog: The E-step indicates which direction may be uphill depending on what you can see from your current position, and the M-step directs you in that direction. Each step either moves you up or, at worst, keeps you at the same height.

A practical example would be:

Suppose we have these X values: [1, 2, 3, 4, 5].

And these Y values are: [2, ?, 5, ?, 8] (where ? signifies missing values).

We might begin with initial guesses:

- $\mu_Y = 5$ (average of observed Y values),
- $\sigma_Y^2 = 9$ (variance between observed Y values),
- $\rho = 0.8$ (an initial correlation estimate).

E-step: We would utilize these parameters to estimate the missing Y values.

M-step: We would use all values (observed and imputed) to calculate improved parameter estimates.

We'd repeat these steps until the changes in parameter estimates become very small, indicating convergence.

Jensen's inequality is a key notion in convex analysis, with significant consequences for the EM algorithm.

First, Jensen's inequality states that for a convex function f:

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

For concave functions, the inequality is reversed:

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)].$$

In the context of the EM algorithm, we are using the log-likelihood function. Let us denote:

- Our parameters as θ (in this example, μ_Y, σ_Y^2, and ρ),
- Our observed data as X,
- Z as our missing data,
- $Q(\theta \mid \theta^{(t)})$ as our estimated complete-data log likelihood.

The main idea is that we can write the log-likelihood as:

$$\log P(X \mid \theta) = \log \int P(X, Z \mid \theta) dZ.$$

Using Jensen's inequality (since log is a concave function):

$$\log \int P(X, Z \mid \theta) dZ \geq \int Q(Z \mid X, \theta^{(t)}) \log \frac{P(X, Z \mid \theta)}{Q(Z \mid X, \theta^{(t)})} dZ.$$

This imbalance is important because:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}} \left[\log P(X, Z \mid \theta) \right].$$

The EM algorithm guarantees the following:

- $L(\theta^{(t+1)}) \geq L(\theta^{(t)})$.

- The E-step uses Jensen's inequality to create a lower bound, which leads to this result.
- The M-step optimizes the lower bound.
- The new bound at $\theta^{(t+1)}$ must be at least as high as the bound at $\theta^{(t)}$.

Therefore, in each iteration:

$$Q(\theta^{(t+1)} \mid \theta^{(t)}) \geq Q(\theta^{(t)} \mid \theta^{(t)}).$$

This mathematical basis explains why the log-likelihood increases (or remains constant) with each iteration. It's like having a series of increasingly better estimates, with each step ensuring that things don't get worse.

1.2 Implement the EM algorithm to handle missing data in a dataset. Your implementation should:

1. **Accept a dataset with missing values:**
   - Generate a synthetic dataset.
   - Eliminate some values from the dataset to obtain a set with missing values.
2. **Randomly initialize the missing values:**
   - Replace missing values with random initial estimates.
3. **Iteratively perform the Expectation-Maximization (EM) steps:**
   - **E-step:** Estimate the missing values based on current parameter estimates.
   - **M-step:** Recompute the parameters (e.g., means, variances, correlations) based on the completed dataset.
4. **Output:**
   - The imputed dataset with missing values filled in.
   - The final parameter estimates, such as means, and variances:

The dataset taken into consideration is the one given at the Expectation Maximization Laboratory of KRR.

Some examples are shown below, where the 2's means missing values:

```
A B C D E F G H I J K L M N O
1 0 1 1 1 0 0 1 2 0 1 0 1 0 2
0 0 1 1 1 0 2 1 1 1 0 0 1 1 2
0 0 1 0 1 2 0 1 2 0 2 1 1 0 1
0 0 1 2 1 2 0 1 0 0 0 0 1 2 0
```

Where A, B, .., O are binary random variables, namely the nodes of the following BN structure and CPDs:

```
15 20
A ; ; 0.75
B ; ; 0.25
C ; A ; 0.75 0.6
D ; A B ; 0.6 0.1 0.6 0.15
E ; C D ; 0.15 0.9 0.45 0.25
F ; E ; 0.5 0.15
G ; B ; 0.1 0.8
H ; ; 0.8
I ; G H ; 0.6 0.9 0.45 0.1
J ; I ; 0.1 0.75
K ; F J ; 0.6 0.5 0.25 0.9
L ; ; 0.45
M ; ; 0.8
N ; K L ; 0.6 0.1 0.45 0.1
O ; L M ; 0.1 0.1 0.1 0.9
```

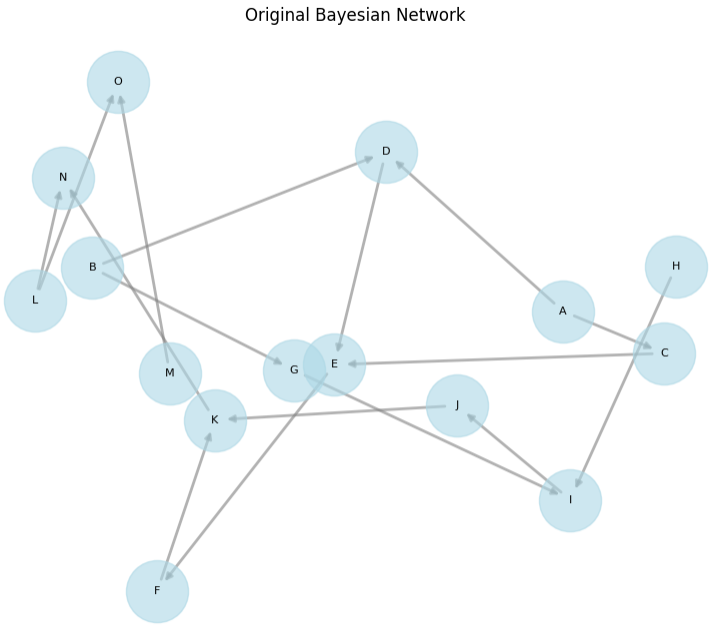

Figure 1: Initial Bayesian Network of the Synthetic Dataset

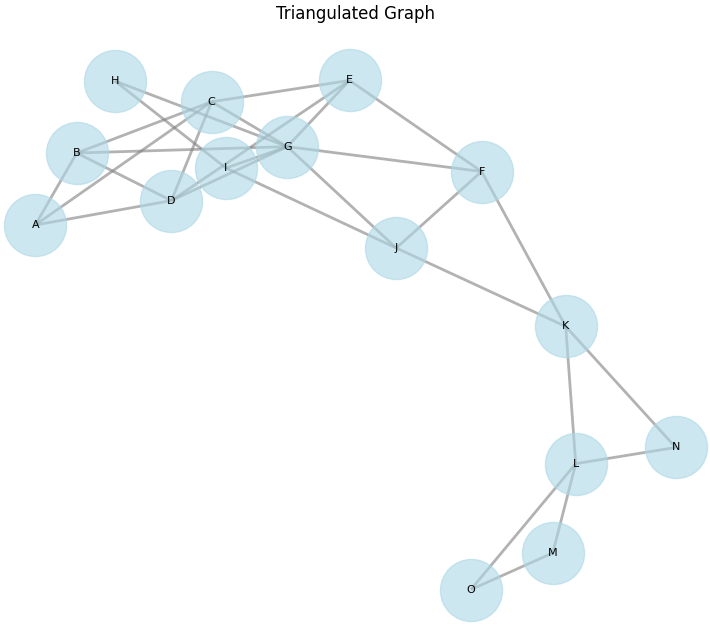

Figure 2: Triangulated Bayesian Network of the Synthetic Dataset

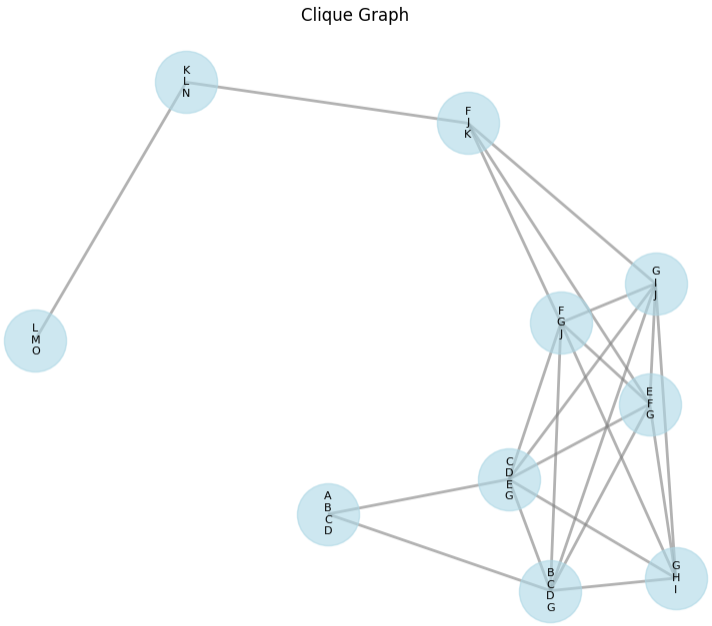

Figure 3: Clique Graph of the Synthetic Dataset

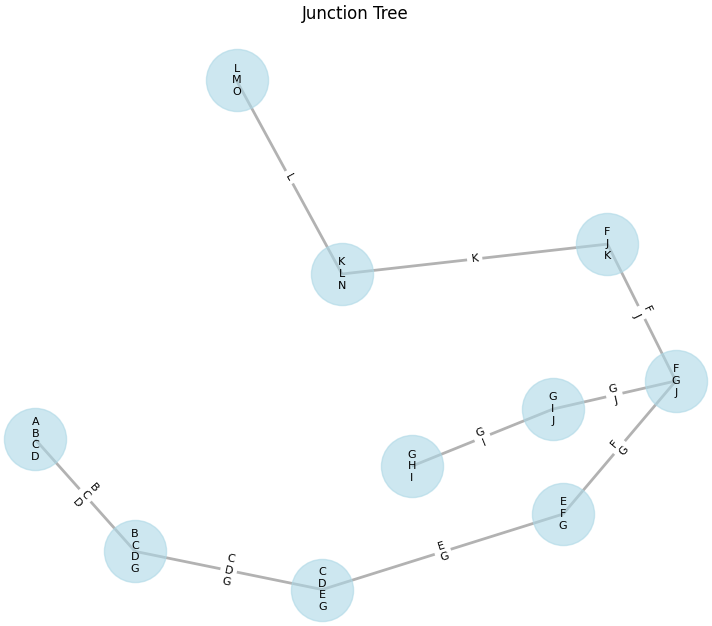

Figure 4: Junction Tree of the Synthetic Dataset

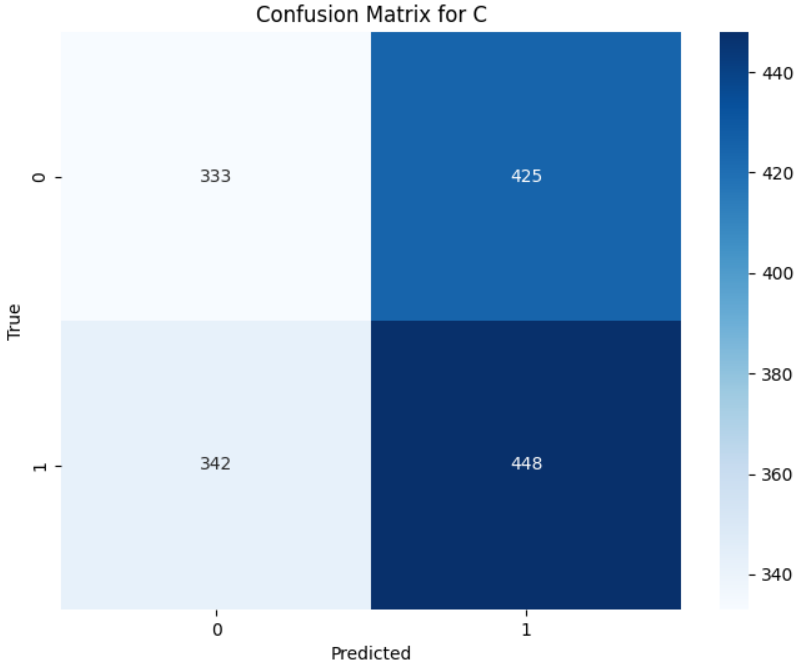

Figure 5: Confusion Matrix of Imputing one Random Variable (such as C)

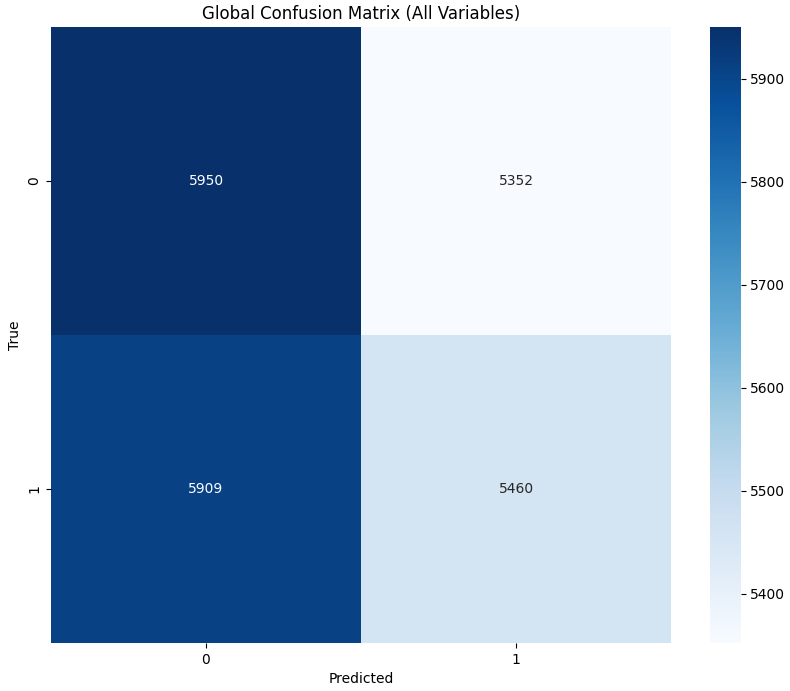

Figure 6: Global Confusion Matrix of Imputing the Datatset

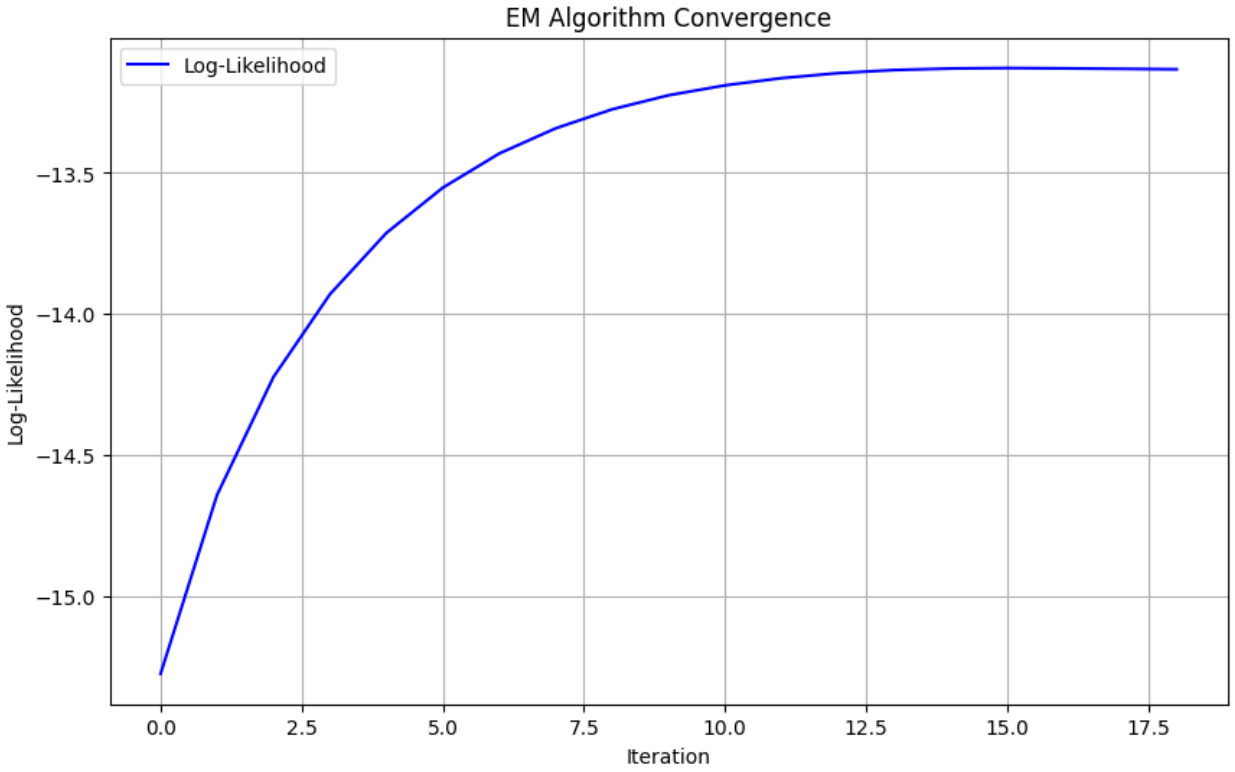

Figure 7: Expectation Maximization Convergence

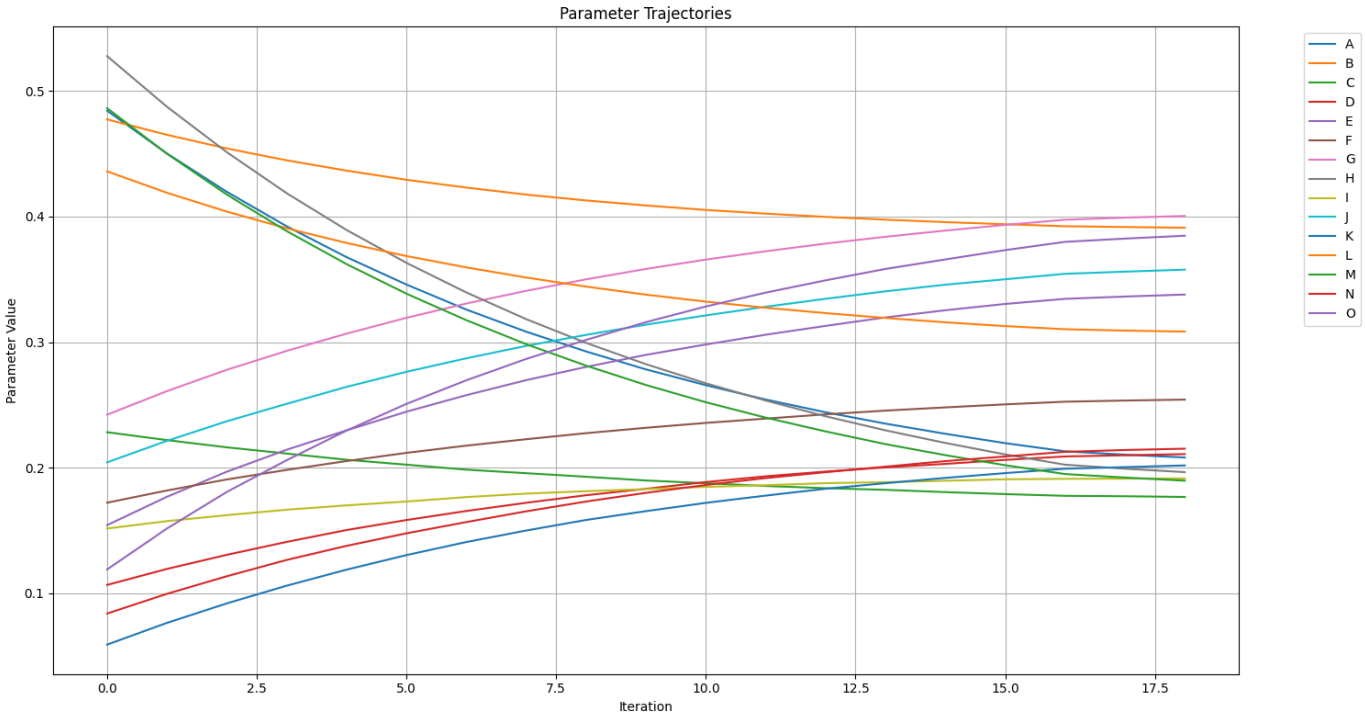

Figure 8: M-Step Parameter Evolution

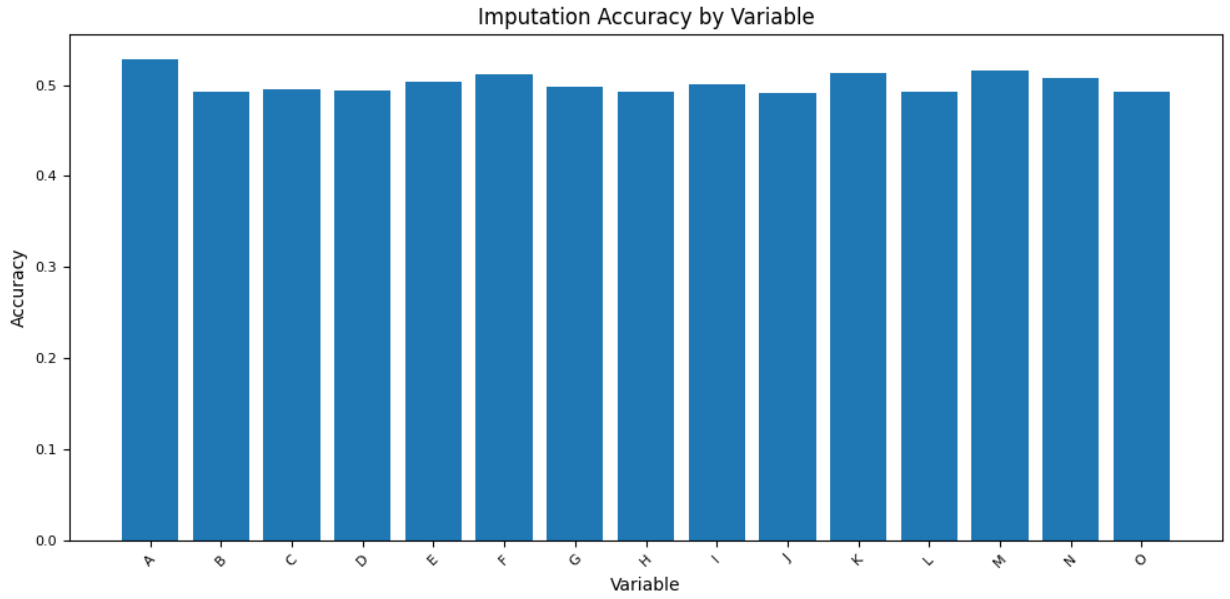

Figure 9: Imputation Accuracy

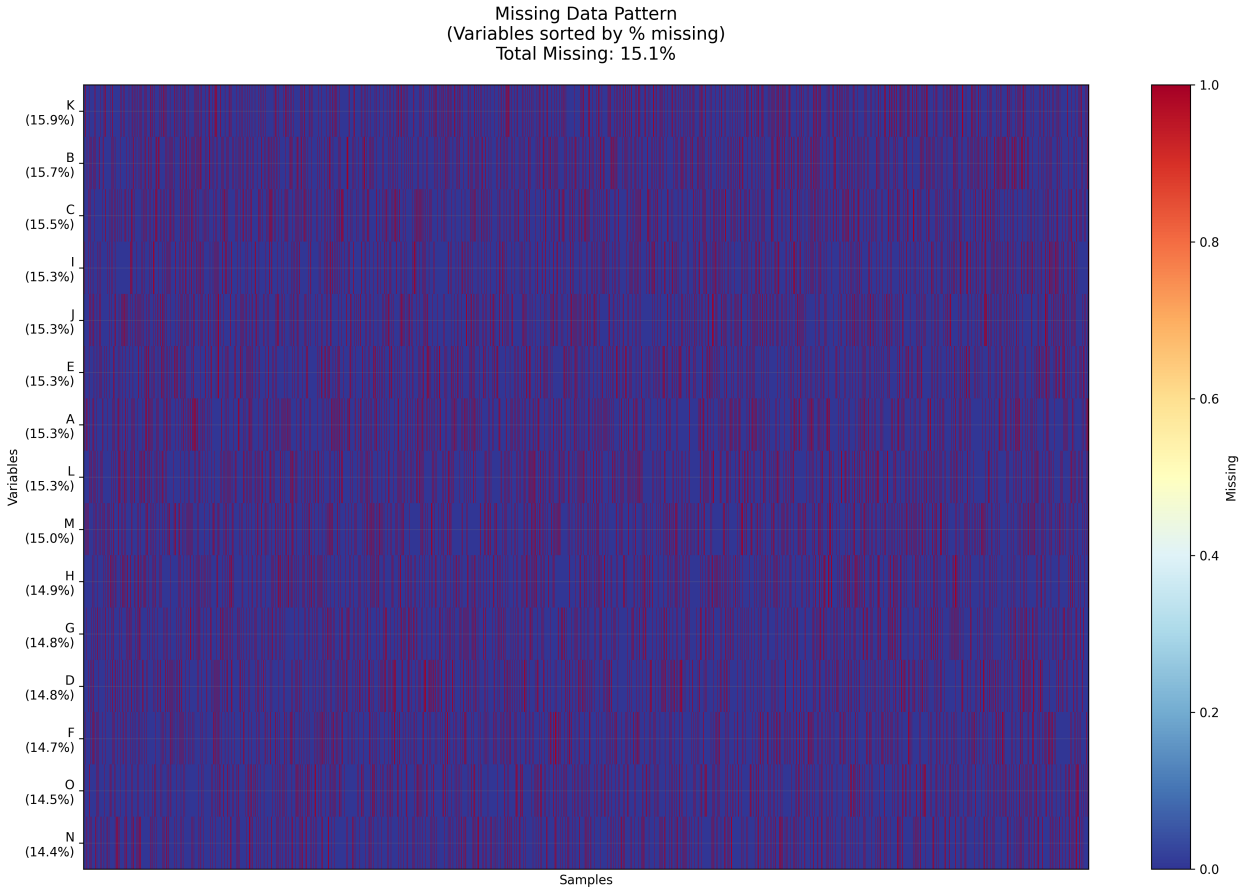

Figure 10: Synthetic Dataset with Missing Values

Node	Accuracy	TN	FP	FN	TP
A	0.4948	140	639	134	617
B	0.4965	618	146	643	160
C	0.5045	333	425	342	448
D	0.4912	422	316	435	303
E	0.5130	454	302	445	333
F	0.5024	516	232	499	222
G	0.5027	594	131	605	150
H	0.5184	189	533	185	584
I	0.4899	250	548	235	502
J	0.4853	358	399	391	387
K	0.5006	381	415	379	415
L	0.5272	584	202	520	221
M	0.4920	108	645	118	631
N	0.5309	564	167	509	201
O	0.5014	439	252	469	286
Global	0.5033	5950	5352	5909	5460

Table 1: Overall Metrics for All Nodes

Node	**Precision**	**Recall**	**F1-Score**	**Support**
Node A				
Value 0	0.5109	0.1797	0.2659	779
Value 1	0.4912	0.8216	0.6148	751
Accuracy	0.4948			
Macro Avg	0.5011	0.5006	0.4404	1530
Weighted Avg	0.5013	0.4948	0.4372	1530
Node B				
Value 0	0.4901	0.8089	0.6104	764
Value 1	0.5229	0.1993	0.2885	803
Accuracy	0.4965			
Macro Avg	0.5065	0.5041	0.4495	1567
Weighted Avg	0.5069	0.4965	0.4455	1567
Node C				
Value 0	0.4933	0.4393	0.4648	758
Value 1	0.5132	0.5671	0.5388	790
Accuracy	0.5045			
Macro Avg	0.5033	0.5032	0.5018	1548
Weighted Avg	0.5035	0.5045	0.5025	1548

Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Value 0	0.4905	0.5628	0.5241	2278
Value 1	0.5091	0.4353	0.4697	2393
Accuracy	0.4986			
Macro Avg	0.4998	0.4991	0.4969	4671
Weighted Avg	0.5000	0.4986	0.4966	4671

Table 3: Global Metrics

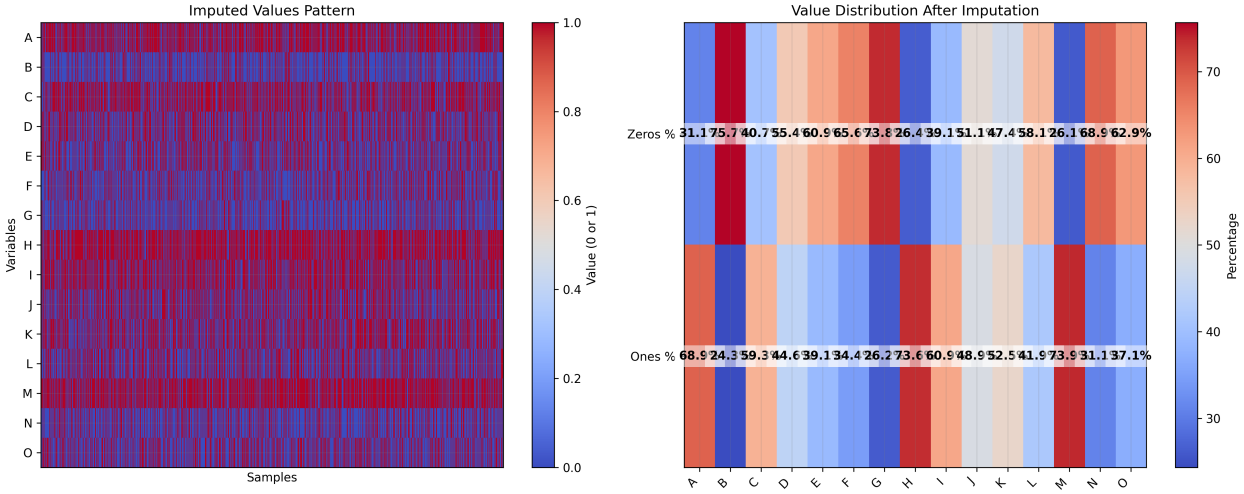

Figure 11: Synthetic Dataset with Imputed Values

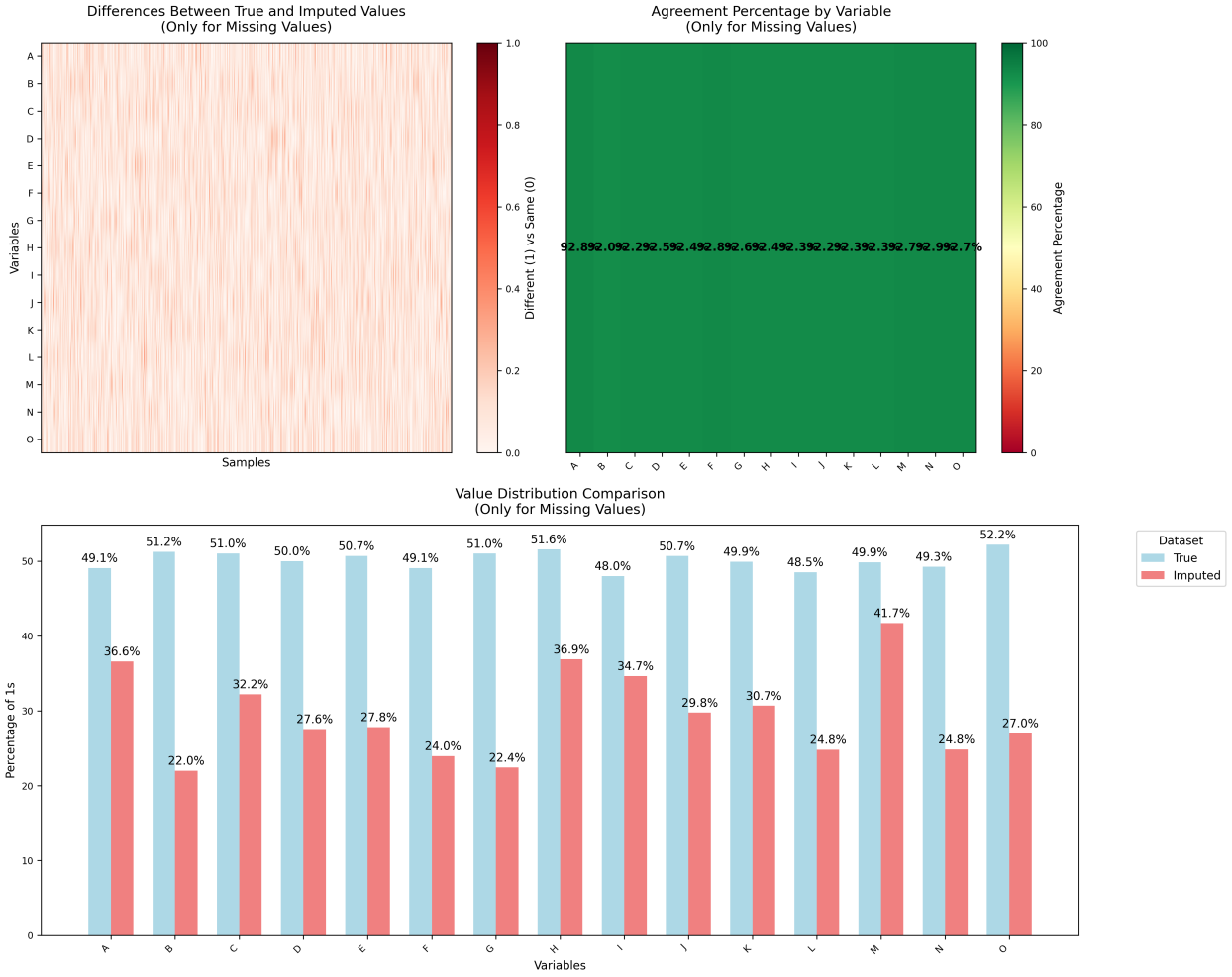

Figure 12: Comparison between both Datasets

Missing Data Summary:
Total samples: 10,000
Total variables: 15
Total missing values: 22,671 (15.1%)

Variable	Count	Percentage
K	1,590	15.9%
B	1,567	15.7%
C	1,548	15.5%
I	1,535	15.3%
J	1,535	15.3%
E	1,534	15.3%
A	1,530	15.3%
L	1,527	15.3%
M	1,502	15.0%
H	1,491	14.9%
G	1,480	14.8%
D	1,476	14.8%
F	1,469	14.7%
O	1,446	14.5%
N	1,441	14.4%

Table 4: Missing percentages per variable

1.3 Choose a real-world dataset with missing values. For example:

- **Air Quality Dataset** (UCI Machine Learning Repository) - includes sensor data with missing entries;
- **Housing Prices Dataset** (Kaggle) - contains features like house prices, lot size, and number of rooms, with missing values in some columns.

You are free to choose another real-world dataset if you wish.

Steps:

1. Preprocess the data as necessary (e.g., handle categorical variables, normalize numerical features).
2. Use your EM algorithm to impute the missing values.
3. Visualize the dataset before and after imputation.
4. Analyze how the imputed values compare with domain knowledge.
5. Evaluate the overall impact of imputation on the dataset's structure (e.g., correlation matrix).

The dataset taken into consideration is the Air Quality Dataset (UCI Machine Learning Repository)

This dataset consists of 9,358 instances of hourly averaged responses collected from an array of five metal oxide chemical sensors installed in an Air Quality Chemical Multisensor Device. The device was deployed in a highly polluted area at road level in an Italian city. Data were collected over one year, from March 2004 to February 2005, making it the longest freely available dataset of on-field air quality chemical sensor device responses. One-hour averaged ground truth concentrations of CO, Non-Methane Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx), and Nitrogen Dioxide (NO2) were provided by a co-located certified reference analyzer. The dataset shows the presence of cross-sensitivities, as well as concept and sensor drifts, due to the measurements performed in changing conditions, and is described in more detail in De Vito et al., Sens. and Act. B, Vol. 129, Issue 2, 2008 (cite.), hence affecting sensor-based concentration estimates accuracy. Missing values are tagged with -200.

Variable Information and Feature Mapping

Feature Name	Variable Name	Description
Date	Date (DD/MM/YYYY)	Date in the format DD/MM/YYYY
Time	Time (HH.MM.SS)	Time in the format HH.MM.SS
CO(GT)	True hourly averaged CO	CO concentration in mg/m^3 (reference analyzer)
PT08.S1(CO)	PT08.S1 (tin oxide)	Sensor response (nominally CO targeted)
NMHC(GT)	True hourly averaged NMHC	Non-Methanic HydroCarbons in μg/m^3 (reference analyzer)
C6H6(GT)	True hourly averaged Benzene	Benzene concentration in μg/m^3 (reference analyzer)
PT08.S2(NMHC)	PT08.S2 (titania)	Sensor response (nominally NMHC targeted)
NOx(GT)	True hourly averaged NOx	NOx concentration in ppb (reference analyzer)
PT08.S3(NOx)	PT08.S3 (tungsten oxide)	Sensor response (nominally NOx targeted)
NO2(GT)	True hourly averaged NO2	NO2 concentration in μg/m^3 (reference analyzer)
PT08.S4(NO2)	PT08.S4 (tungsten oxide)	Sensor response (nominally NO2 targeted)
PT08.S5(O3)	PT08.S5 (indium oxide)	Sensor response (nominally O3 targeted)
T	Temperature	Temperature in °C
RH	Relative Humidity	Relative humidity in percentage (%)
AH	Absolute Humidity	Absolute humidity in g/m^3

Table 5: Variable information and mapping with feature names.

Some examples are listed below, where missing values are tagged with -200 values (listing up to the first 8 features because of size constraints).

```
Date,Time,CO(GT),PT08.S1(CO),NMHC(GT),C6H6(GT),PT08.S2(NMHC),NOx(GT)
10/03/2004,18.00.00,2.6,1360.0,150.0,11.9,1046.0,166.0
10/03/2004,19.00.00,2.0,1292.0,112.0,9.4,955.0,103.0
10/03/2004,20.00.00,2.2,1402.0,88.0,9.0,939.0,131.0
11/03/2004,03.00.00,0.6,1010.0,19.0,1.7,561.0,-200
11/03/2004,04.00.00,-200,1011.0,14.0,1.3,527.0,21.0
12/03/2004,03.00.00,0.8,889.0,21.0,1.9,574.0,-200
12/03/2004,04.00.00,-200,831.0,10.0,1.1,506.0,21.0
```

Column	Missing Values	Percentage
NMHC(GT)	8,443	90.23%
CO(GT)	1,683	17.99%
NO2(GT)	1,642	17.55%
NOx(GT)	1,639	17.52%
PT08.S1(CO)	366	3.91%
C6H6(GT)	366	3.91%
PT08.S2(NMHC)	366	3.91%
PT08.S3(NOx)	366	3.91%
PT08.S4(NO2)	366	3.91%
PT08.S5(O3)	366	3.91%
T	366	3.91%
RH	366	3.91%
AH	366	3.91%

Table 6: Missing values and their percentages for each column

Where Date, Time, .., AH are continuous numerical variables (features), so there is a need to convert them into binary variables (discrete), which implies establishing some thresholds:

Column	Threshold
CO(GT)	2.00
PT08.S1(CO)	1063.00
NMHC(GT)	500.00
C6H6(GT)	5.00
PT08.S2(NMHC)	909.00
NOx(GT)	100.00
PT08.S3(NOx)	806.00
NO2(GT)	40.00
PT08.S4(NO2)	1463.00
PT08.S5(O3)	963.00
T	25.00
RH	60.00
AH	1.00

Table 7: Threshold values for each column.

As it can be seen, the processed dataset doesn't have Date and Time features anymore, mainly because of their datatypes which couldn't be treated as the others (which are, in fact, pure numerical).

The created Bayesian Network for this task is shown below, based on the following domain knowledge:

- **CO(GT)** influences **PT08.S1(CO)** (direct sensor relationship).
- **NMHC(GT)** influences **PT08.S2(NMHC)** and **C6H6(GT)**.
- **NOx(GT)** influences **PT08.S3(NOx)** and **NO2(GT)**.
- **NO2(GT)** influences **PT08.S4(NO2)**.
- **O3** is measured by **PT08.S5(O3)**.
- **Temperature (T)**, **Relative Humidity (RH)**, and **Absolute Humidity (AH)** are related.

```
13 0
RH; ; 0.5
Temp; ; 0.5
AH; RH Temp; 0.5 0.5 0.5 0.5
NMHC_GT; ; 0.5
C6H6_GT; NMHC_GT; 0.5 0.5
CO_GT; ; 0.5
NOx_GT; CO_GT Temp; 0.5 0.5 0.5 0.5
NO2_GT; NOx_GT; 0.5 0.5
PT08_S1_CO; CO_GT; 0.5 0.5
PT08_S2_NMHC; NMHC_GT; 0.5 0.5
PT08_S3_NOx; NOx_GT; 0.5 0.5
PT08_S4_NO2; NO2_GT; 0.5 0.5
PT08_S5_O3; NOx_GT RH Temp; 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
```

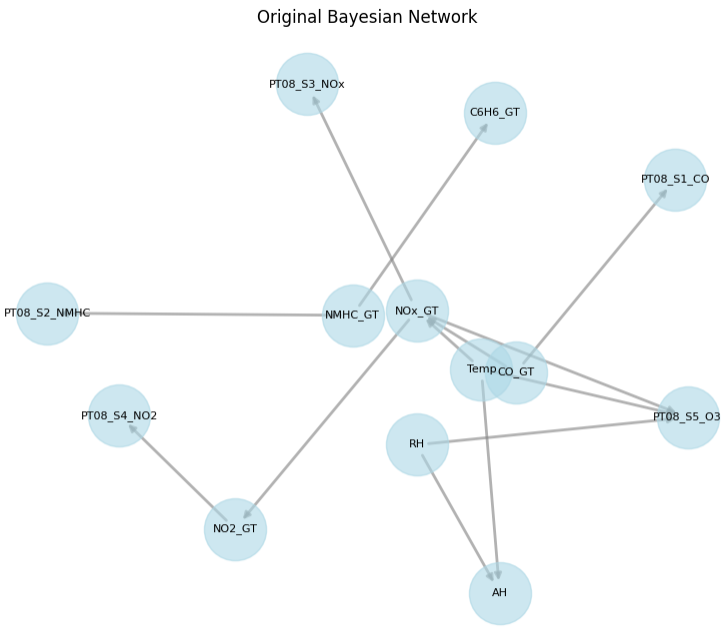

Figure 13: Initial Bayesian Network of the Air Quality Dataset

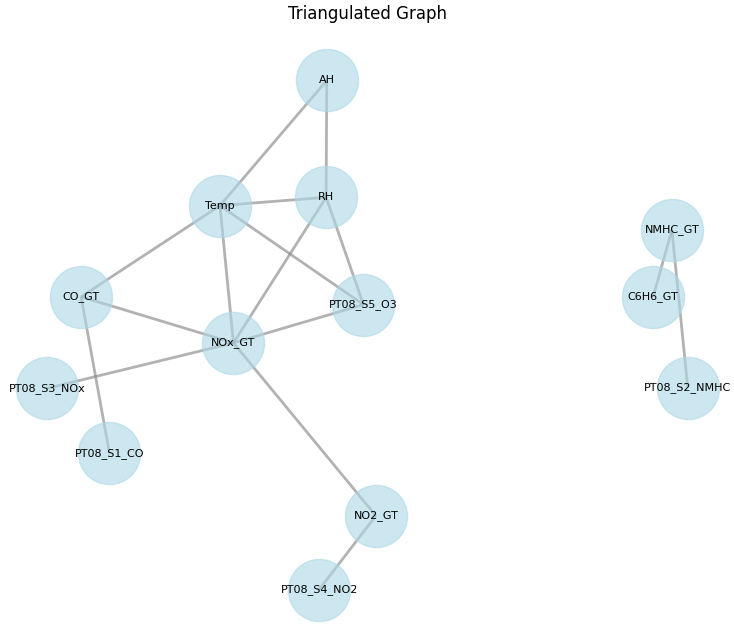

Figure 14: Triangulated Bayesian Network of the Air Quality Dataset

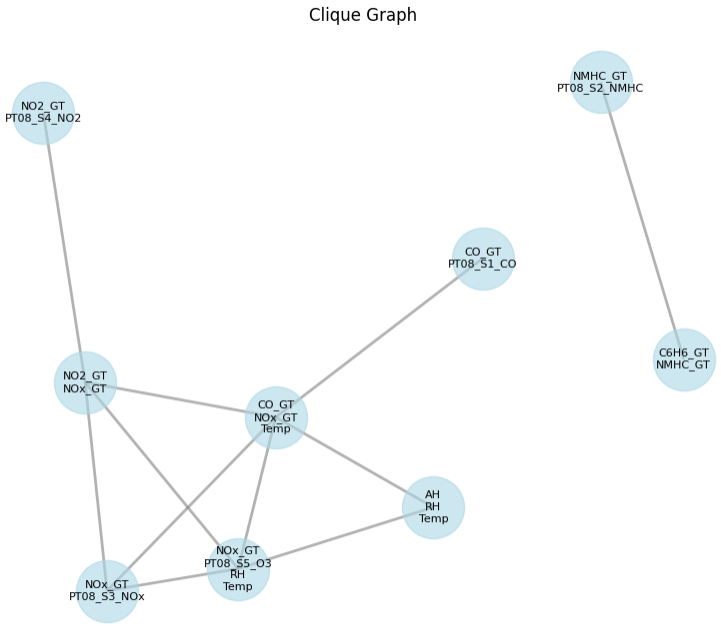

Figure 15: Clique Graph of the Air Quality Dataset

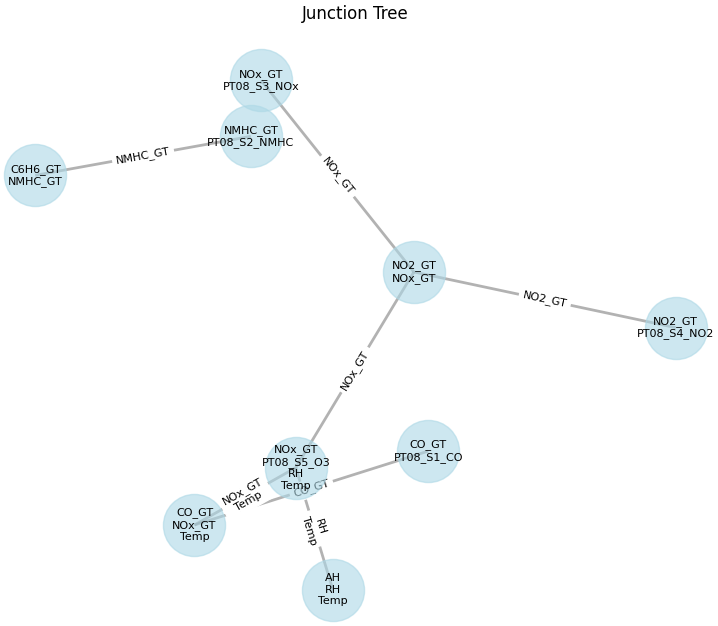

Figure 16: Junction Tree of the Air Quality Dataset

Missing Data Summary:
Total samples: 9,358
Total variables: 13
Total missing values: 16,701 (13.7%)

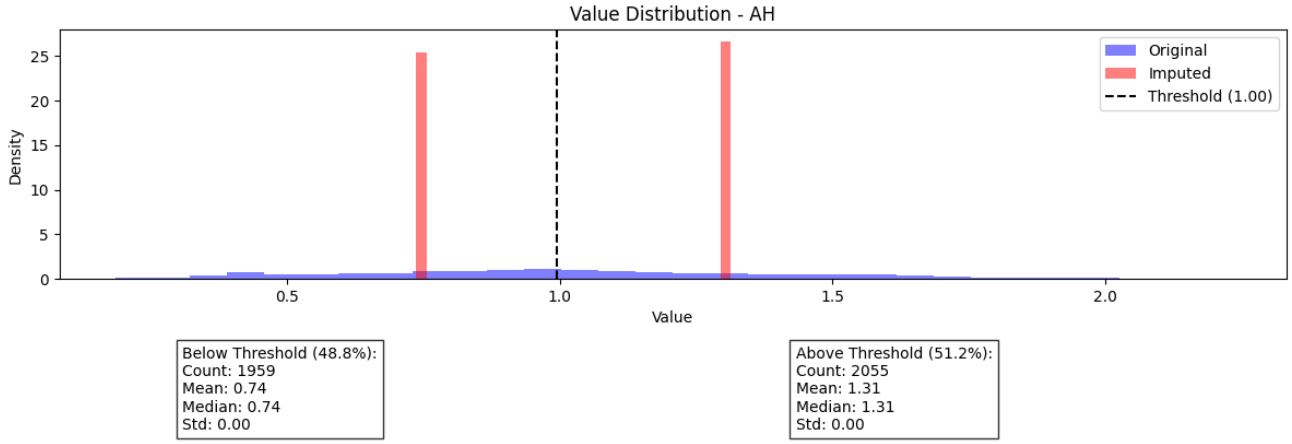

Figure 17: Distribution of one Random Variable / Feature (such as AH)

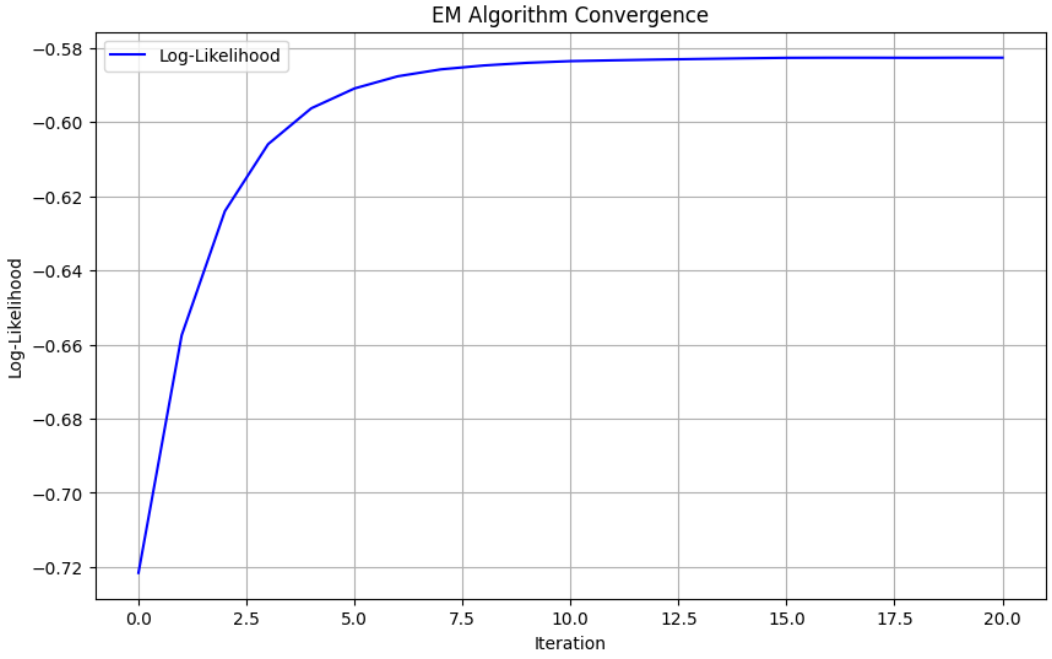

Figure 18: Expectation Maximization Convergence

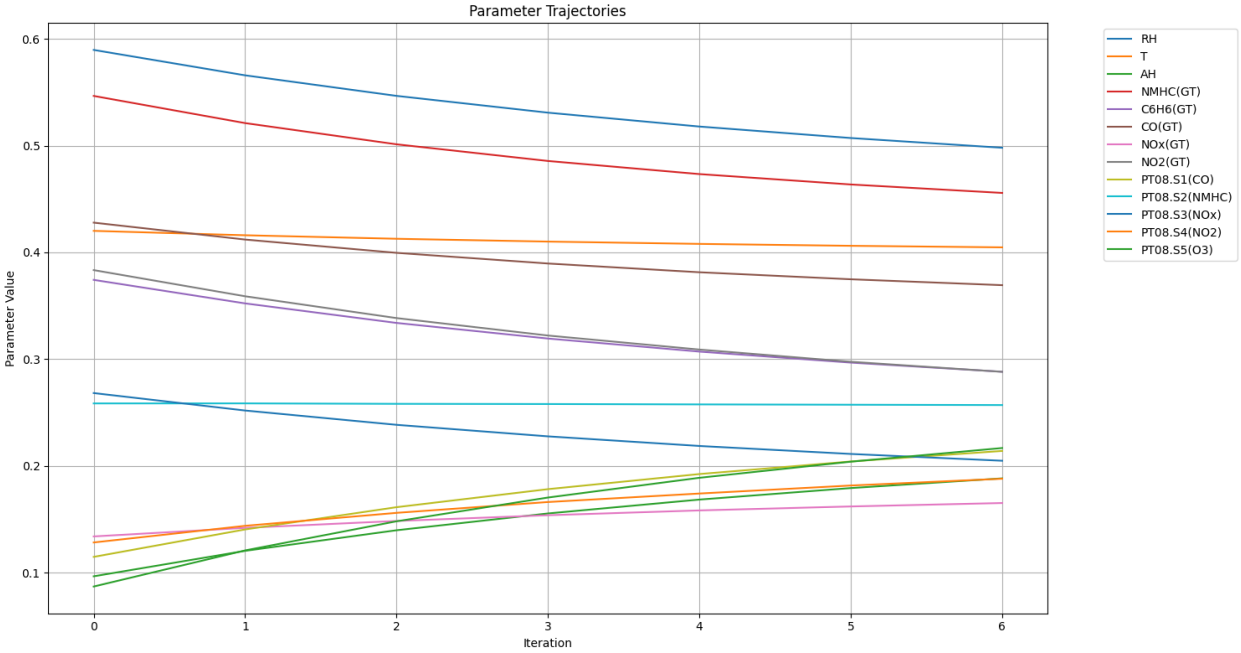

Figure 19: M-Step Parameter Evolution

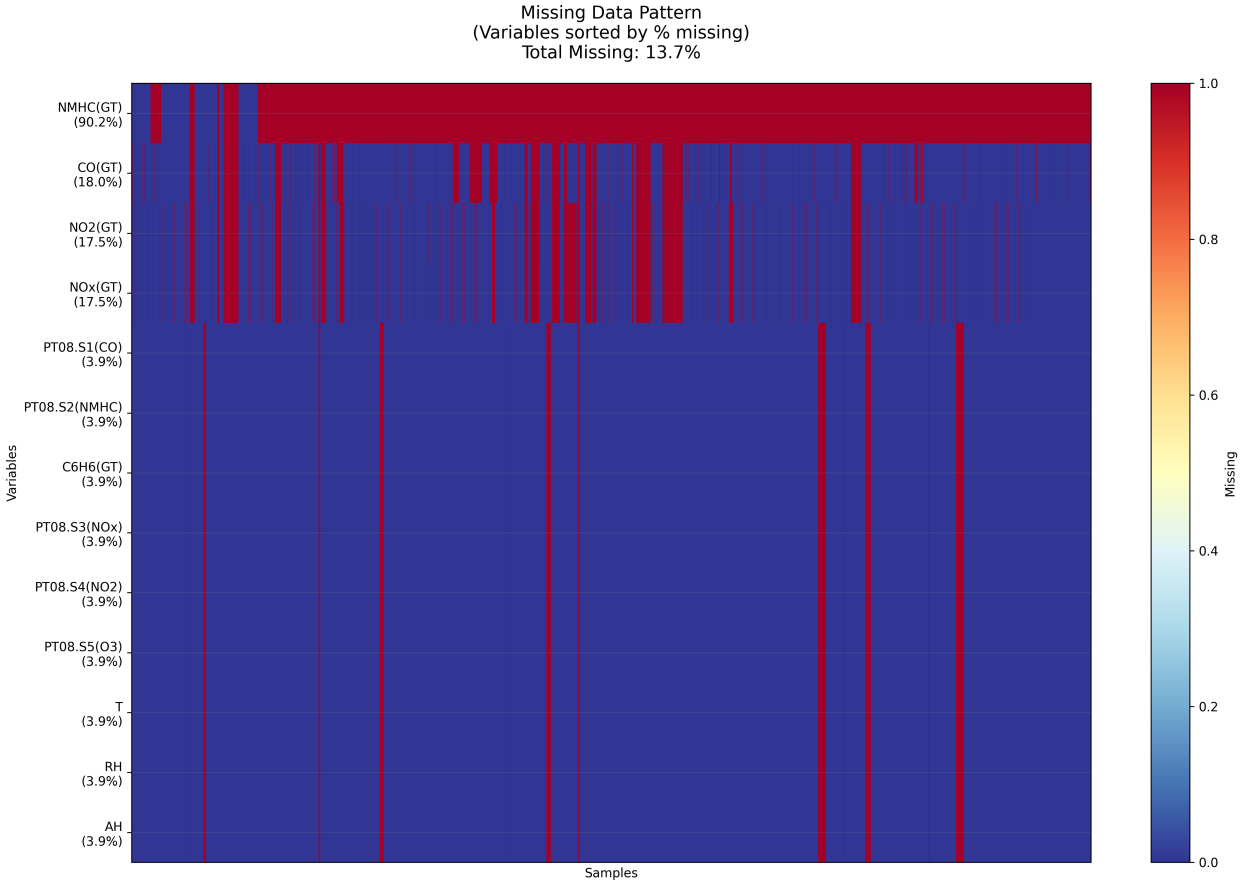

Figure 20: Air-Quality Dataset with Missing Values

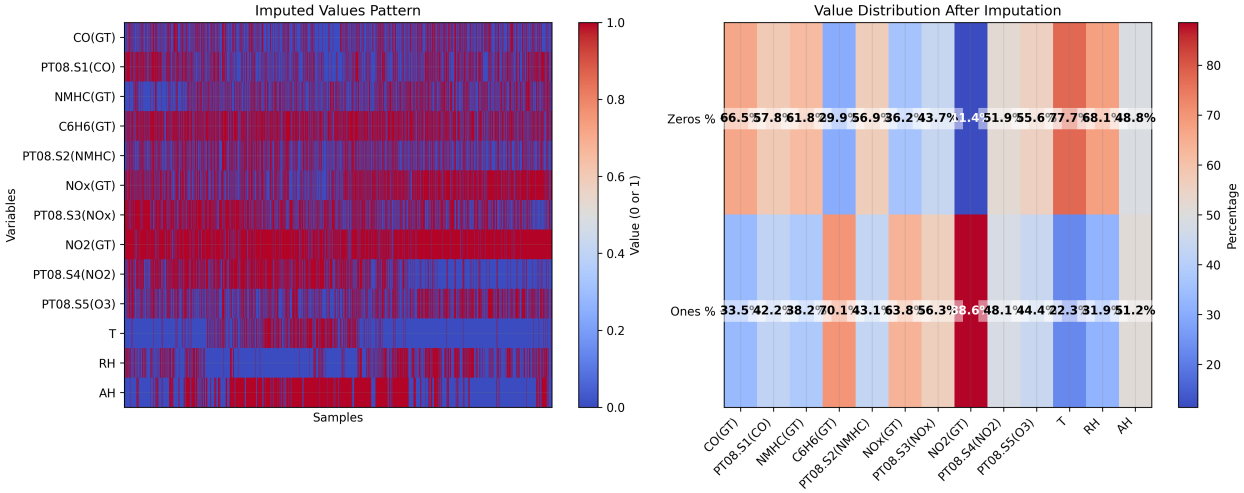

Figure 21: Air-Quality Dataset with Imputed Values

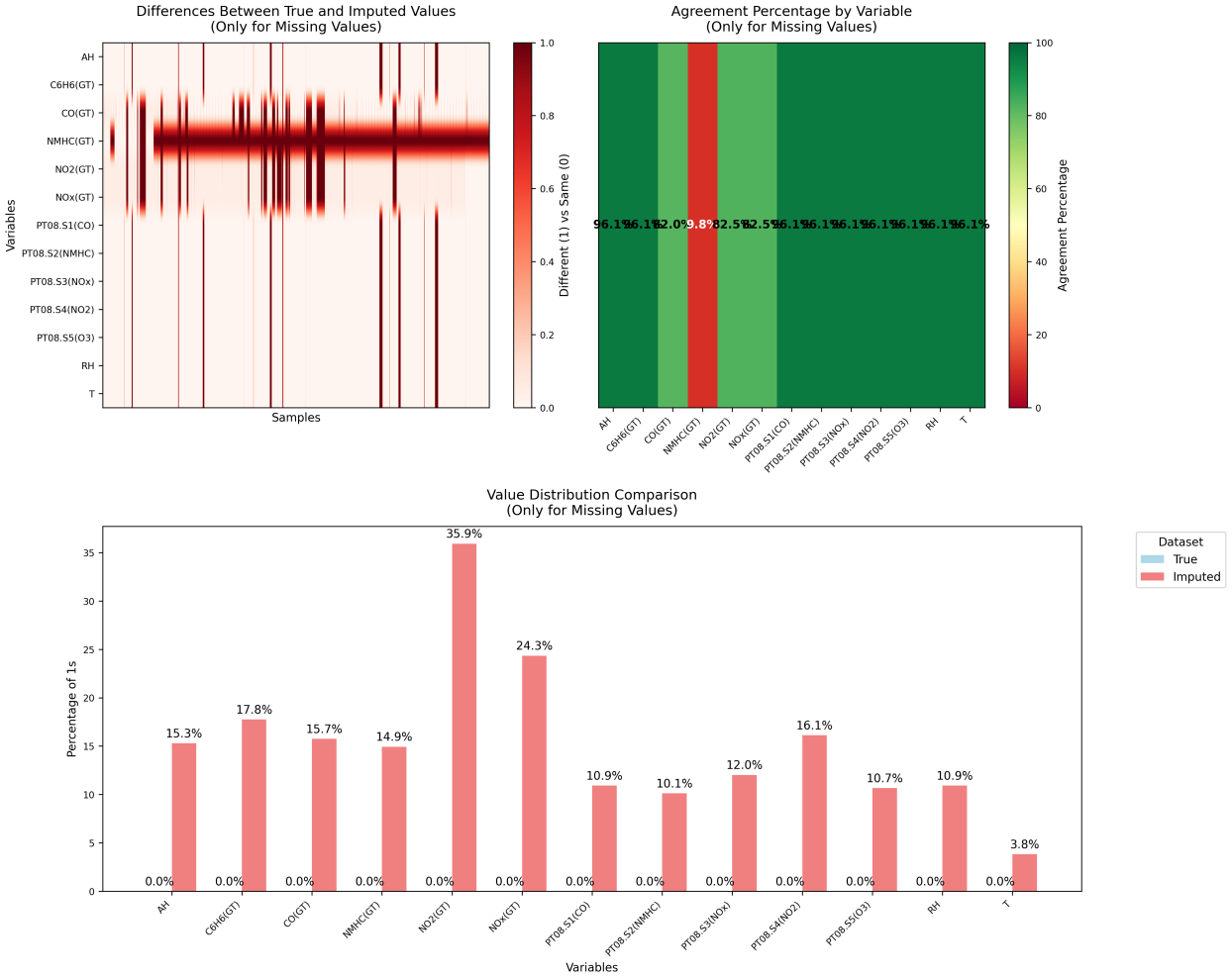

Figure 22: Comparison Between Real and Imputed Values of Air-Quality Dataset

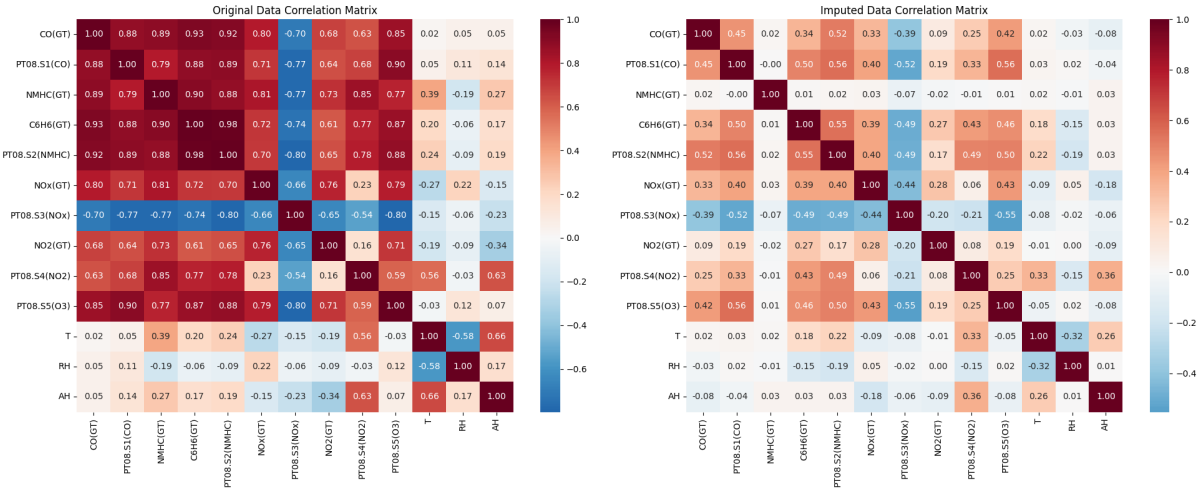

Figure 23: Analysis of Correlation of Air-Quality Dataset

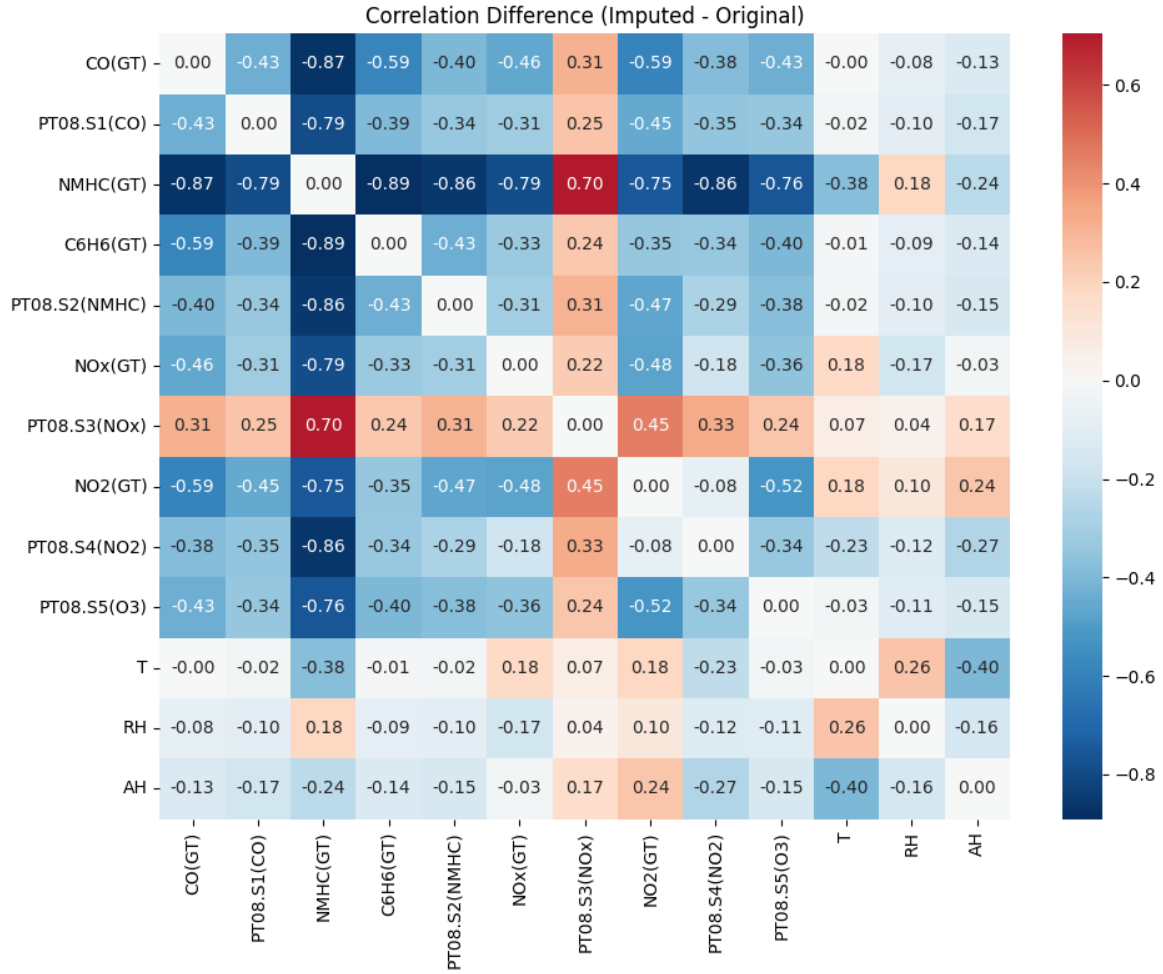

Figure 24: Correlation Difference of Air-Quality Dataset

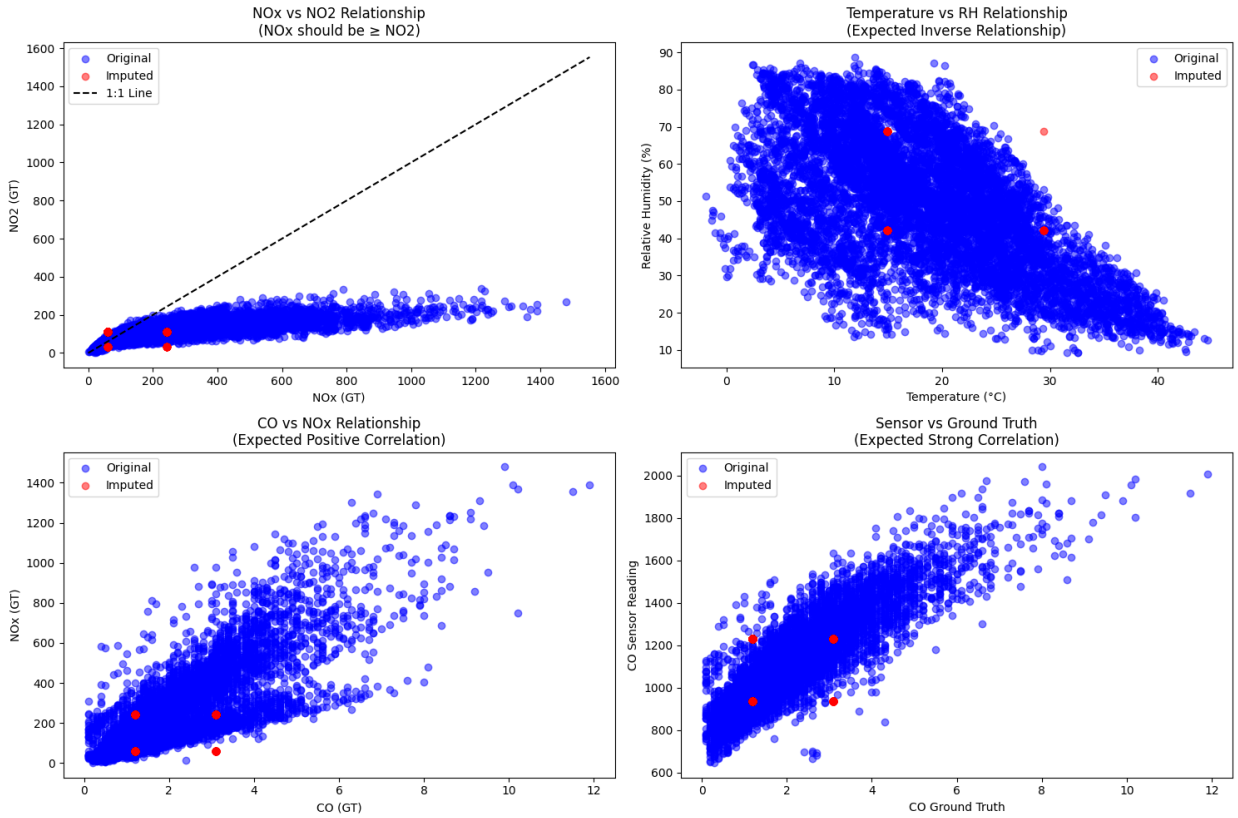

Figure 25: Domain Knowledge Analysis of Air-Quality Dataset

Unfortunately, Expectation Maximization doesn't look like it converges well for this dataset. Of course, several design choices can be improved, such as the structure of the Bayesian Network by asking a domain expert and the selected thresholds based on a much deeper analysis of the data concerning tendencies, standard deviations, medians, and correlations.