Imputing Missing Values Using CPT-v Model

Christine Hwang
Department of Statistics
Harvard University
Email: chaehwang@college.harvard.edu

Weiwei Pan
Professor for Data Science CS 109
Harvard University
Email: weiweipan@g.harvard.edu

Missing data is an inevitable problem that is growing as big data becomes highly coveted to make data driven decisions. Missing data can be ignorable if the probability of the data being missing is independent of the value itself. It becomes non-ignorable if the value of the missing data is correlated to missingness. In this paper, I will approach the challenge of imputing non-ignnorable missing data using a probabilistic model, specifically the CPT-v model which is dervied from a multinomial mixture model to impute these missing values while taking into account the dependency of missingness on the value. I present my findings both with synthetic data and observed data to showcase the accuracy of this model compared to a standard method of imputing missing values using the mean.

1 Introduction

1.1 Background

Data collection is growing to be one of the most valuable assets to make data driven decisions. Big data can be used to track consumer behavior, make business decisions, improve security, and expand our knowledge to simply understand and process more of our surroundings. However, as data grows larger and carries more information, there is a higher probability of obtaining missing data. In real life, data collection is messy, often compiled from voluntary responses or from automated processes that can have glitches that impede recording data. However, not all missing data is treated equally. Data can be missing in two ways: one that is ignorable and one that is non-ignorable. In the instance where there is an automated process that glitches and misses to record data every 10 seconds, the data is missing at random because the value of the observation does not have a correlation to whether it is missing or not. However, if the machine glitches anyone the person marks gender as Female, then the data is no longer missing at random and cannot be ignored. In this paper, I will aim to approach the challenge of handling non-ignorable missing data.

1.2 Motivation

Handling non-ignorable missing data is crucial because simply dropping these values creates biased parameter estimates when fitting a model. Take for example, collecting data from movie ratings. Because not everyone rates every single movie they watch, missing data becomes an issue. When observing human behavior in rating movies, we often tend to rate great movies that we love and poor movies that we did not enjoy, and often decide it's not worth rating an average, mediocre movie. Therefore, the data is non-ignorable because it is not missing at random and is skewed towards ratings for only terrible and great movies. Since the probability of the data being missing is dependent on the value of the observation, the parameter estimates for models built off this model can affect the significance of our results. The statistics from the sample of observed data is no longer a unbiased estimator of the population we want to analyze. Therefore, handling missing data is an interesting and worthwhile problem to explore because it can increase the predictive power of many models that attempt to draw conclusions using datasets with non-ignorable missingness.

2 Probabilistic Model

I will be using a probabilistic modeling approach to impute the missing values for non-ignorable missing datasets. By doing so, I will be able to estimate the parameters of the distribution that our data comes from and randomly sample from these distributions when trying to predict missing values. For our specific data application, I will be assuming a multinomial mixture model.

Suppose I am given a dataset $\mathbf{Y}_{n,m}$ and $\mathbf{R}_{n,m}$. $\mathbf{R}_{n,m}$ represents the missing data matrix where $R_{n,m} = 1$ if the data is observed and $R_{n,m} = 0$ if the data is missing. Each value of $y_{n,m}$ represents a categorical value from the range $\{1...V\}$. Each observation n comes from a latent variable z that is derived from a dirichlet prior, $P(Z = z) = \theta_z$. Since each value in $\mathbf{Y}_{n,m}$ takes on an integer value of a fixed range, every $\mathbf{Y}_m \mid z_n \sim$ Multi($\beta_{m,z}$,N) where $\beta_{v,m,z} = P(Y = v|z)$ and comes from a dirichelet prior. I will also define $\mu_v = P(R_{n,m} = 0|Y_{n,m} = v)$. This essentially defines the probability of a data point being

missing given the true value of the data point. μ_v will come from a beta distribution because it is a prior for a binomial distribution. Lastly, I will define $\phi_{z,n} = P(z = k|Y_n, R_n)$ as the posterior distribution of the latent variable z and k is a hyperparameter that defines the number of latent variables.

2.1 Graphical Model

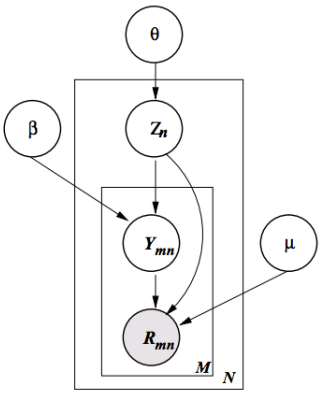

Figure 1: Combined data and selection model.

2.2 CPT-v Model

Using the model described above and through the graphical model, I will use the results cited from Marlin et al (1) to derive the EM algorithm for the multinomial mixture model, specifically the CPT-v model. The CPT-v model is based on the assumption that the probability of data being missing is only dependent on the value itself, not on other predictors as well. The likelihood of the CPT-v model derived by Marlin et al (1) is below.

$$L(\theta, \beta, \mu|\mathbf{Y}, \mathbf{R}) = \sum_{n=1}^{N} log \sum_{z=1}^{k} \theta_z \prod_{m=1}^{M} \gamma_{m,z,n}$$

$\lambda_{v,m,z,n}$ and $\gamma_{m,z,n}$ are intermediate variables defined as the following.

$$\lambda_{v,m,z,n} = (\delta(y_{m,n}), \mu_v)^{r_{m,n}} (1 - \mu_v)^{1 - r_{m,n}} \beta_{v,m,z}$$
$$\gamma_{m,z,n} = \sum_{v=1}^{V} \lambda_{v,m,z,n}$$

2.3 Prediction Modification

In a standard CPT-v model, after the parameter estimates are fit, the data can be imputed by drawing a single value from the estimated distribution. I first took $\phi_{z,n}$, which represents the posterior distribution of the latent variable z, and selected the z' for each n where $\phi_{z',n}$ is the highest. Then I imputed the value by sampling from $Multi(\beta_{m,z'}, 1)$. However, upon initial experimentation, the CPT-v model was not performing better than the standard mean imputation. My hypothesis was that the mean imputation was very consistent in its prediction and showed low volatility in accuracy whereas imputing with a single sample from the distribution had very high variance. Rather than drawing a single observation from the estimated distribution to impute the value, I drew a sample of values from the distribution and took the mean($Multi(\beta_{m,z'}, nsample)$), immensely improving my accuracy. Subsequent results were calculated using this method of prediction. As you can see from the Figure 1, the trade-off between the computation time and increase in accuracy leveled off when nsample = 10.

Algorithm 1 EM Algorithm for CPT-v Model

procedure E STEP:(▷)Expectation Step
  $\lambda_{v,m,z,n} = (\delta(y_{m,n}), \mu_v)^{r_{m,n}} ((1 - \mu_v)\beta_{v,m,z})^{1 - r_{m,n}}$
  $\gamma_{m,z,n} = \sum_v^V \lambda_{v,m,z,n}$
  $\phi_{z,n} = \frac{\theta_{z_n} \prod_m^M \gamma_{m,z,n}}{\sum_k^K \theta_{z'} \prod_m^M \gamma_{m,z,n}}$
end procedure
procedure M STEP:(▷)Maximization Step
  $\mu_v = \frac{\sum_n^N \sum_{z=1}^K \phi_{z,n} \sum_m^m r_{m,n} \lambda_{v,m,z,n} / \gamma_{m,z,n}}{\sum_m^m \sum_{z=1}^k \phi_{z,n} \sum_m^M \lambda_{v,m,z,n} / \gamma_{m,z,n}}$
  $\theta_z = \frac{\sum_n^N \phi_{z,n}}{\sum_n^N \sum_{z=1}^K \phi_{z,n}}$
  $\beta_{v,m,z} = \frac{\sum_n^N \phi_{z,n} \lambda_{v,m,z,n} / \gamma_{m,z,n}}{\sum_n^N \phi_{z,n}}$
end procedure

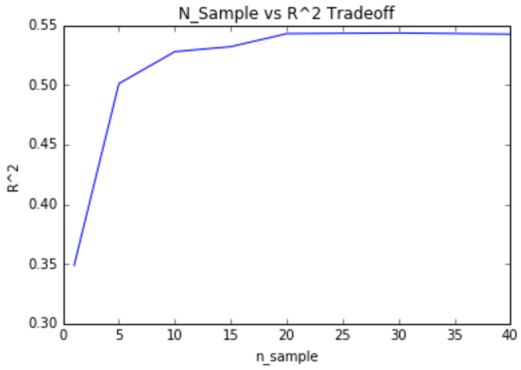

Fig. 1: Prediction Method Modification

3 Experimental Procedure

Using the mechanism detailed by Marlin et al, I ran both synthetic data experiments as well as real observed data experiments to understand the performance of this model compared to the standard missing data imputation of using the mean and also identifying under which conditions the CPT-v model performed best. The conditions I experiment over are the level of noise amongst the predictors, the level of missingness, and the number of latent variables that is present in the dataset. By running both synthetically generated data and real data, I can understand how to CPT-v model performs under heavily controlled environments and how that contrasts to a real life application.

3.1 Synthetic Data Procedure

3.1.1 Pseudocode for Data Generation

By generating synthetic data, I am able to take control of the noise to determine how the performance of the CPT-v model varies as we control for different aspects of the data. I generate the data by drawing the true β values from a Dirichlet distribution and assigning each observation to a latent variable z. Given that I know which latent variable each observation belongs to, I can populate the data by sampling from the beta distribution that corresponds with the latent variable. I then create a μ using a random number generator to determine what the relationship between the value of the data is to the probability of it being missing. After I populate μ to decide which values are more likely to be missing, I populate the $\mathbf{R}_{n,m}$ to drop the values and create a synthetically generated data set.

Algorithm 2 Data Generation

procedure GENERATE SYNTHETIC DATA(▷)
  Generate $\beta_{v,m,z}$ using Dirichlet distribution for each z and m
  Generate which latent variable z_k each observation came from using randomint.
  Populate $\mathbf{Y}_{n,m}$ where each $Y_m|z_n \sim Multi(\beta_{m,z}, N)$
  Generate μ using np.random.random or manually
  Populate $\mathbf{R}_{n,m}$ to identify missing data points
  Populate target variable as a linear combination of predictors with added random normal noise
end procedure

3.2 Model Performance

Without altering the noise, missingness or latent variables, I ran a controlled experiment where z = 1, m = 5, v = 5, n = 1000, 5 times to compare the performance of the CPT-v model against the baseline mean model. The average R^2 for the CPT-v model is .55 and the average R^2 for the standard mean model is .44. The CPT-v model beats the baseline model by a difference of .11 on average.

3.2.1 Number of Latent Variables

One of the factors that can be controlled for is the number of latent variables. In real life, not all data is drawn from the same distribution and by adding a latent variable z, we are able to acknowledge that each data point can come from a different distribution. We want to see how robust the CPT-v model is to increasing number of latent variables compared to the standard imputation model of using the mean. By being able to identify the latent variable class, I hypothesized that the CPT-v model would perform much better than the standard imputation model as z increases.

I conducted an experiment by running 5 iterations of the CPT-v model on a synthetically generated data set where I vary the number of latent variables from z = [1,...,9], n = 1000, v = 5, and m = 5. I then took the average difference between the R^2 of the model using the CPT-v model imputed data and the R^2 of the model using the standard mean imputed data. As seen above in Figure 2, the CPT-v model performed much better relative to the standard model when z = 3. As the number of latent variable increased, the performance of the CPT-v model became similar to the standard mean imputation model. This could be because as the number of latent variables increases, the sample size of data in each latent variable decreases, resulting in a higher variance in estimating the distribution parameters.

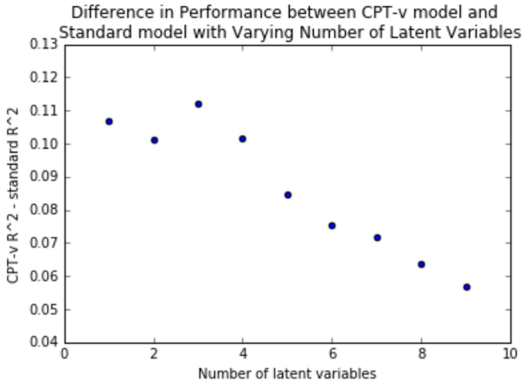

Fig. 2: Latent Variables

From our results, we can also see that the difference between the R^2 of the CPT-v model and standard model is above 0, which means that the CPT-v model better captures the true value of the data when the missingness is dependent on the value. In addition, we can see that the CPT-v model peaks in performance with a moderate number of latent variables.

3.2.2 Level of Missingness

Another control I can place on the synthetic data is to understand how well the CPT-v model performs under varying levels of missing data. I ran an experiment to control for μ by altering the range in the uniform distribution that it samples from. Recall that $\mu_v = P(\mathbf{R}_{n,m} = 0|\mathbf{Y}_{n,m} = v)$. I define low level of missingness to be when μ_v is sampled from $Unif(0, .5)$, which means that each value cannot have more than a 50% chance of being missing, middle level of missingness is when μ_v is sampled from $Unif(.25, .75)$, and high level of missingness is when μ_v is sampled from $Unif(.5, 1)$. We also want to compare this against the control group, which is when μ_v is sampled from $Unif(0, 1)$. We then ran a linear regression and compared the R^2 of the data imputed with the CPT-v model with the data imputed with the standard mean model and compared the performance depending on the varying levels of missingness.

For this experiment, I ran 5 repetitions with n = 1000, m = 5, v = 5 and z = 3, per the results of the prior experiment. Similar to the controlled experiments, we can see that the CPT-v model on average performs higher than the standard imputation model on all levels of missingness and the

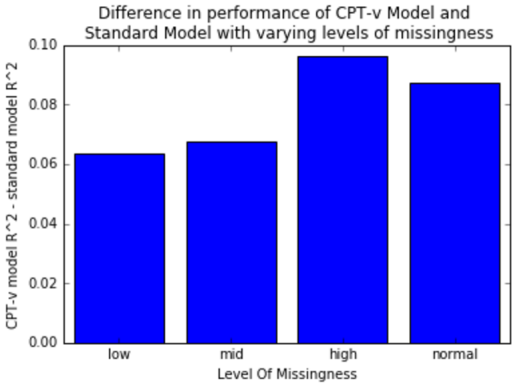

Fig. 3: Levels of Missingness

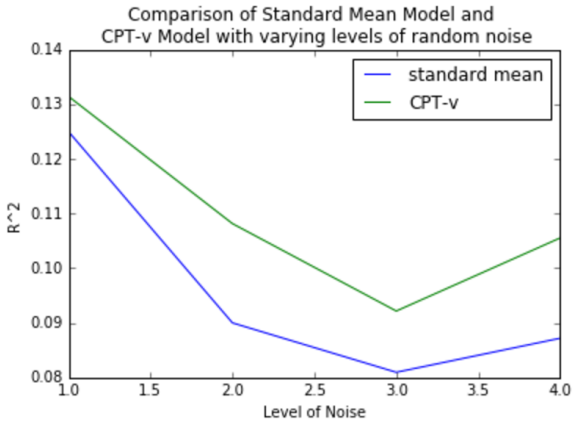

Fig. 4: Random Noise

control. In terms of the conditions that the CPT-v model performs relatively better in, it seems that the higher the missingness of a dataset, the higher the CPT-v model performs in relation to the standard mean imputation model. We can see that for datasets with a high missingness, the CPT-v model has an R^2 value that is on average .10 better than the standard mean imputation whereas for datasets with a low missingness, the CPT-v model has an R^2 value that is only .06 better than the standard mean imputation.

These results make sense because in the case of high missing data, calculating the mean to impute the missing values will give you a biased estimate of the mean because the data is missing in a non-random manner. Therefore, you are filling in a higher proportion of the dataset using a biased mean estimate. Therefore, the CPT-v is a relatively more robust model of data imputation when the proportion of missing data is greater than 50%.

In addition, in cases with low levels of missingness, there should not be a large discrepancy in the R^2 of the two methods because a majority of the training and testing data is the same since the only different values are the imputed missing values. Because I train the models without tuning and under the same conditions, the discrepancy in the performance is most evident as missingness increases because the two datasets become increasingly different.

3.2.3 Noise

Lastly, I aimed to control the amount of noise amongst the predictor variables to gauge how well the CPT-v model performed under those conditions. By doing so, we want to see if the CPT-v model still performs better than the standard mean model even if the dataset is corrupt or people report data that is higher or lower than the true value. Because the values must take on integer values, I added random noise using a random integer generator. The range of the noise varies from +-1, +-2, +-3, and +-4. Acknowledging that in the case where V $\leq$3, adding noise of +-4 could affect the model because it would allow for negative observations. To account for this, after the noise was added, all negative values were converted to 1, which is considered to be the lowest value in the multinomial distribution.

Each experiment was repeated 5 times to calculate the average R^2 of the model. As seen from the Figure 4, both the standard mean model and the CPT-v model have decreasing performance as the level of noise increases. However, the CPT-v model consistently has a higher R^2 than the baseline meaning that under conditions of high noise, the CPT-v model does a better job of capturing the true value of the data than imputing with the mean. To compare under what conditions the CPT-v model performs relatively best in, I want to compare the difference in the R^2 of the standard mean model and the CPT-v model.

There is not a clear pattern to suggest the CPT-v model performs better than the standard model with respect noise. While there is a peak when the noise is at +-2 and +-4 in Figure 5, there is not a clear upward trend to provide evidence that the difference between CPT-v model's performance and the standard model's performance increases with the level of noise. However, because the difference is consistently above zero, there is evidence to suggest that in the cases where there could be noise in the data due to corrupt data input or faulty voluntary responses, the CPT-v model will better be able to take this into consideration in imputing the data with values that are more like the true distribution.

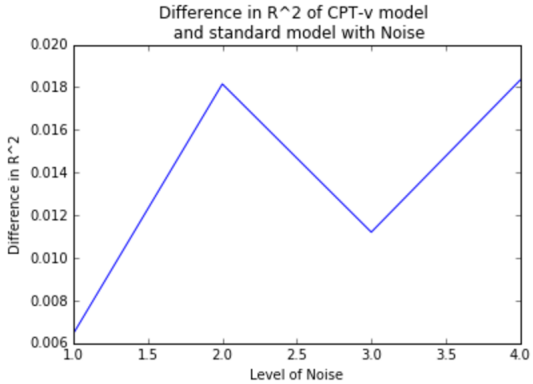

Fig. 5: Noise

3.3 Real Data Procedure

3.3.1 Background

For our real data, I use a Mammographic Mass Dataset from UCI Machine Learning Repository. Mammography is a method of detecting breast cancer and this dataset aims to predict the severity of mammographic mass lesions using several predictors. It uses the predictors age: which takes an integer value, shape: categorical variable that is round,oval, lobular or irregular, margin: categorical variable that is circumscribed, microlobulated, obscured, ill-defined, and spiculated, and density: ordinal variable that is high, low, iso, and fat-containing. The target variable is a binary variable severity, which takes on a value 1 if it is malignant and 0 if it is benign. This dataset has 5 missing age variables, 31 missing shape variables, 48 missing margin variables, and 76 missing density variables. This dataset is fitting to impute the values using CPT-v model because each value in the dataset takes on an integer value and therefore we can model it using a multinomial mixture model.

3.4 Procedure

In our synthethic data generation, I had a consistent V among the different m columns to create consistency in the shape of the matrices. However, this is not practical for real life data since all the columns do not have the same V unless you are dealing with movie ratings data. In the case of the Mammographic Mass Dataset, there are 3 predictors with a significant level of data missingness. Because these three predictors have different V, I split these up into 3 separate datasets where z = 1, n = 961, m = 1, v = predictor.max() to run through the EM algorithm. I then concatenate the three columns with the filled in values to create the full dataset with missing values imputed by the CPT-v model. In our synthetic data study, I saw that the CPT-v model performed best under conditions of high missingness, latent variable around 4, and blank data. I control for these same factors in the Mammographic Data set to compare how the performance of CPT-v model differs in synthetic data and real data.

3.5 Model Performance

Without controlling for noise, missingnes, or latent variables, the performance of the CPT-v model slightly better than the standard model. The average accuracy score of the CPT-v model imputed data was .810 and the average accuracy score of the standard mean imputed method was .808. On average, the CPT-v model method performs about .00189 better in accuracy score. However, there is quite a difference in performance compared to the synthetic data. Recall that the difference in performance using the synthetic data is .11, which is much higher than the current difference of .00189 using the Mammographic Data. This could be expected because real life data may not be able to control for the assumptions of the CPT-v model such as the probability of the data being missing being only dependent on the value itself, not on other predictors. In real life, shape and margin may be correlated because round mass legions may be more likely to be circumscribed, and if one value is missing, it may be more correlated that the other is as well. While it is expected that the difference in performance between the Mammographic dataset and the synthetic dataset is different, I will aim to control for levels of missingness and number of latent variables to analyze under what conditions of real data the CPT-v model will perform better than the standard mean baseline model by a higher margin.

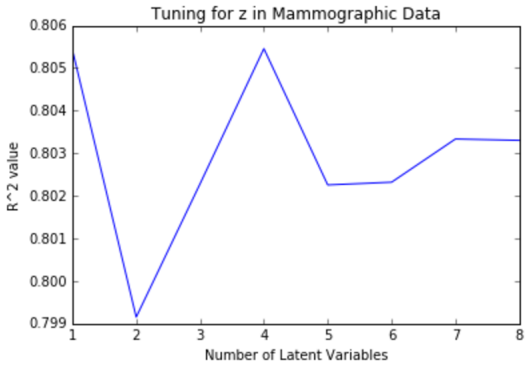

Fig. 6: Tuning Z Mammographic Data

3.5.1 Tuning for Latent Variable

Unlike the cases in the synthetic data set, in real life data, I do not know how many latent variables there can be. Therefore, I will tune for then number of latent variables by using 5-fold cross validation. To do so, I run the EM algorithm 5 times on 5 different sets of training and validation sets for each z in range 1 to 9 and take the average R^2 value. My hypothesis is that the data will have the highest R^2 and perform the best when the z that I am tuning is the closest to the true z value of the data.

As shown in Figure 6, the model performs the highest when z = 4. Therefore, for the remainder of the experiments, I will assume that using z = 4 can best captures the information in the data.

3.5.2 Level of Missingness

As seen from the analysis done with synthetic data with regards to different levels of missingness, there was the highest margin in the difference in performance between the CPT-v model and the baseline mean model when there are high levels of missing data. In synthetic data, we saw that approximately 50% of the data is missing when the margin in performance is highest, but in this Mammographic dataset, only about 10% of the data is missing. Therefore, in such low levels of missing, it is difficult to see a difference in the performance of the model because the dataset from the baseline model and the dataset from the CPT-v model are 90% similar.

Therefore, I attempt to experiment by generating a higher proportion of missingness in the dataset by sampling approximately a similar number of complete data rows as missing data rows. I essentially control for the number of complete data rows I sample from the dataset to alter the proportion of

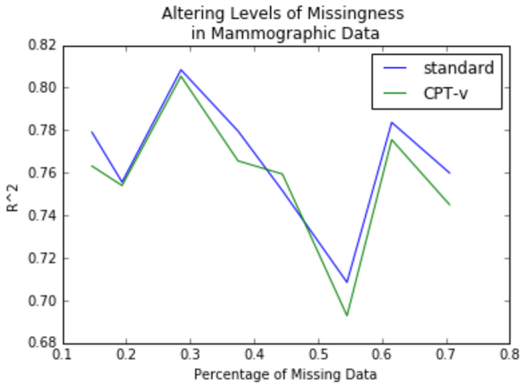

Fig. 7: Levels of Missingness

missing data. Each proportion was repeated 5 times to take the expected performance of the model based on the level of missingness.

As seen in Figure 7, the difference between the R^2 of the standard model and the CPT-v model do not increase as we alter the level of missingness. In fact, it seems that the performance of the CTP-v model is on average slightly lower as we increase the level of missingness.

3.5.3 Analysis and Drawbacks

As we can see from the results from both the synthetic data experiments and the real data experiments, there is quite a difference in the performance of the CPT-v model between the two. While we did tune for z, the number of latent variables and the levels of missingness, it seems that the performance of the CPT-v model still does not show a large margin of increase in R^2 performance. This is likely because the difference in R^2 between the CPT-v model and the standard model with added noise is quite small even in our synthetic data runs as seen in the axis of Figure 5. Because real life data often contains a lot more noise than synthetic data, the large margin by which the CPT-v model often performs better by is much lower when we use the model on real data. We saw an average difference of around .01. Considering also that the CPT-v model assumes that the probability of missingness is entirely dependent on only the value of the missing data, the real mammographic data could violate these assumptions, also leading to slightly lower performance than what was seen in the synthetic data.

4 Conclusion and Future Work

Imputing missing values helps capture the true value of missing data and allows us to encapsulate more information as missingness increases in data collection. Acknowledging that data is not always missing in a random pattern and that dependencies exist help us increase our accuracy in imputation and ultimately gives our data more predictive power to create useful internal tools and predictions. Though the CPT-v model has limitations in assumptions because real life data is not always missing depending on the value alone, it allows us to consider complex methodologies using a probabilistic model and estimate the distribution of the data. Future work to improve the performance of the CPT-v model is to experiment with dependent noise. With correlated noise, noise in one column could be connected to the next, and the CPT-v model may perform much higher than the baseline model under such conditions. If given more time, I should also compare against other baseline models, such as the mode, or using KNN or regression to impute missing values and evaluate the runtime and performance there as well. Another option would be to also explore the Logit-v,mz model that is also detailed in Marlin et al(1). This model does not assume that missingness does not solely depend on the value and allows for interactive effects on missingness.

5 Acknowledgement and Materials and Methods

Thank you to Weiwei Pan for allowing me to research this methodology and provide guidance thorughout the semester. The experiments were run using Python v2 and the dataset was used from UCI Machine Learning Repository. Papers read are included in the references. Code can be found on https://github.com/chaehwang/stat91r/blob/master/CPT-v%20Model%20Implementation.ipynb

References

[1]B. Marlin, R. S. Zemel, and S. T. Roweis. Unsupervised learning with non-ignorable missing data. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005.

[2]Marlin, B.M., Zemel, R.S., Roweis, S., Slaney, M.: Collaborative filtering and the missing at random assumption. In: Proceedings of the 23rd COnference on Uncertainity in Artificial Intelligence (2007)

[3] Blei, David, Mixture Models and Expectation - Maximization, 2012, https://www.cs.princeton.edu /courses/archive/spring12/cos424/pdf/em-mixtures.pdf

[4] Bryan, Li; Recap: Gaussian Mixture Modeling, 2014, http://web.stanford.edu/ lmackey/stats306b/doc/sta ts306b-spring14-lecture3$_{s}cribed.pdf$

[5] B.M. Marlin, R.S. Zemel, S.T. Roweis, and M. Slaney. Recommender systems: missing data and statistical model estimation. In IJCAI, 2011.

[6]http://www.cs.ubc.ca/ bmarlin/research/presentations/lni md$_{g}roup_{t}alk.pdf$

[7] https://pdfs.semanticscholar.org/2845/eda7ce8de14e35 1d41182f92b73ece8873ef.pdf