Predictive Mean Matching for Images

Nathan de Lara
McGill University
MATH 598

Abstract

Predictive Mean Matching (PMM) is a Multiple Imputation (MI) technique for handling missing data. This report proposes PMM for missing image labels by projecting each image onto $\mathbb{R}^c$ where c is the number of classes. We evaluate two ways of doing this projection, the first is with an autoencoder and the second is by learning a classifier that projects samples onto $\triangle^c$. We compare these two strategies to a naive strategy of multiple imputation which imputes according to predicted probabilities from a classifier trained on the observed data. The three strategies are evaluated on datasets where the missingness is Missing Conditionally at Random (MAR) or Missing Not at Random (MNAR) and an ablation study on the performance with respect to the perverseness of missingness is performed. We find that using PMM in conjunction with Convolutional Neural Networks (CNNs) outperforms a naive approach of using CNNs predicted probabilities. Our work shows that by reframing PMM as a method that can be combined with any projection from observations into a metric space opening the door for other ways of using PMM in new data paradigms different than the tabular one.

Keywords: Missing Categorical Data, Predictive Mean Matching, Machine Learning for Image Labelling, Missing Conditionally at Random, Missing Not at Random,

1 Introduction

In the age of big data and data analysis, organizations have hundreds of thousands of datapoints, often without direct human labels. To make sense of this wealth of data it is imperative for companies to have labels or annotations. Typically companies having huge datasets of unlabelled images will hire people to label a subset of the entire dataset, then, they will train a statistical model on the labeled data and use it to predict labels for the unlabelled data. When the images that get assigned labels are chosen at random then the learned distribution from image pixel data to label is equal to the real distribution. But, when the images chosen for labelling are not chosen at random, there is a distribution shift between the training distribution and the real distribution that can lead to biased imputations.

There are three levels of missingness, they are: Missing Completely at Random (MCAR), Missing Conditionally at Random (MAR), and Missing Not at Random (MNAR) [Van18]. When data is Missing Completely at Random the observed distribution is the same as the underlying true distribution and so traditional statistical techniques can be used as they would on a dataset of full observations and imputations can be done in a straightforward way. The Missing Conditionally at Random assumption states that for an image i given the covariates $\mathbf{x_i}$, whether or not an image is missing M_i is independent of it's label y_i. Missing Not at Random is then the assumption that given any of the observed covariates $\mathbf{x_i}$ the missingness of the image M_i is dependent on it's label y_i. MCAR is clearly the easiest to deal with followed by MAR for which there are a host of readily established techniques to deal with tabular datasets in a manner that requires no domain knowledge from the statistical practitioner. MNAR on the other hand, is the hardest among the three types of

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

missingness and requires domain knowledge as stricter assumptions about the data-generating process need to be made. The graph below shows the graphical form for what the MCAR, MAR and MNAR assumptions specify about the dependencies between $\{x_i\}_{i\in D}, \{y_i\}_{i\in D}, and \{M_i\}_{i\in D}$.

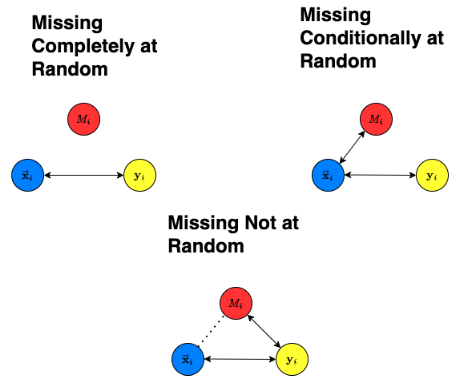

Figure 1: Graph shows the conditional distributions that exist for the MCAR (top left), MAR (top right), and MNAR (bottom) assumptions to be made, the dotted line the MNAR drawing is to signify that the relation between $\tilde{\mathbf{x}}_\mathbf{i}$ and M_i does not impact the MNAR assumption if M_i is dependent on y_i

In the Missing Not at Random (MNAR) and Missing Conditionally at Random (MAR) cases techniques for handling the distribution shift from fully observed data to the underlying data generation process have to be used alongside the statistical models that practitioners want to implement. Multiple Imputation is a method for dealing with missing values that generates multiple potential values the missing covariates could take, runs analysis on them, and then pools the results from each imputation to get a general estimate. Predictive Mean Matching (PMM) is a method for Multiple Imputation in MAR situations originally proposed by Rubin [Rub86] and further developed by Little [Lit88]. The main idea behind PMM is to learn a function from the covariates on which every entry is observed to the covariates that have missingness. With that function imputation is done by getting the distances between predictions for all entries and then for the ones with missingness, sampling, proportional to their distance, an observed value of y from the k closest fully observed entries. PMM has found success due to being being robust to model misspecification and straighforward to apply.

This report explores applying PMM to image data that has a majority of it's labels missing. Leveraging the idea of using distances between predicted values, we train different models to project images onto a smaller dimensional space where we then use the distances between them to impute values in the same way that PMM does, we then evaluate the different models on their imputation accuracy. We explore two different ways of performing this projection from images onto a smaller dimensional space, the first is by learning an image classifier that takes images as inputs and returns the probabilities for each class, in this case the classifier projects images onto a c-dimensional simplex, where c is the number of classes, and the standard euclidean distance is used to generate imputations according to PMM. The second projection is gained by training an autoencoder, which has a bottleneck layer in it's architecture, to compress the image information through the bottleneck and then reconstruct the input image, we do this by minimizing the distance between an input image and the output. By using a bottleneck layer the high-dimensional pixel information get's compressed into a much smaller dimensional representation that we use to compute distances and generate imputations. The most important aspect of this is to get a projection f from the image to a smaller space where there is some structure to the space $\{f(x_i)\}_{x_i\in D}$ directly related to the label, in the classifier case this structure is directly imposed as the $f(x_i)$ are probabilities for each label, in the case of the auto-encoder, it learns low-dimensional representations of images and so the underlying assumption that images look like specific labels gives us reason to believe that it learns a useful structure.

We evaluate a PMM strategy that uses CNN classifiers; an alternative PMM strategy that uses CNN autoencoders; and a naive strategy of using the classification probabilities outright for imputation on a synthetic dataset of images with some missing labels. We use the CIFAR-10 image dataset which contains 6,000 32x32 pixel rgb images each of which belongs to one of ten classes to generate the datasets upon which we evaluate. CIFAR-10 has an image for every label but we carefully create functions which map an image to a probability of being missing while respecting either

MAR or MNAR, and then according to those probabilities construct datasets with missing labels. We compare the performance of the three strategies in MAR and MNAR situations and observe how their performances change with respect to the amount of missingness in data. We find the convolutional autoencoder with PMM outperforms the convolutional classifier with PMM or naive predicted probability imputation and that all methods are robust to increasing amounts of missingness. This work opens the door for more methods that lean on PMM and leverage general metric space structure instead of pure distances between predicted values for multiple imputation.

2 Relevant Background

2.1 Missing Conditionally at Random (MAR)

Formally the Missing Conditionally at Random assumption is that for some observed covariates x, $y \perp M|x$ [Van18]. Under this assumption, classical complete-case analysis leads to biased estimates since the observed joint probability $P(x, y|M = 1) \neq P(x, y)$. To counteract this multiple methods have been proposed that leverage some of the distributions $P(x|M), P(y|x), P(M|x), P(M), P(x)$ such as Inverse Probability Weighting (IPW) [SW13], Expectation-Maximization (EM) [ZLK10], and Multiple Imputation (MI) [Roy04]. Since the observed covariates are all that is needed to account for the bias from the missingness these methods do not require significant domain knowledge as opposed to the MNAR case. Central to all of them is that the complete case data distribution is different than the underlying distribution of the data and so estimators for the data need to account for this, IPW accomplishes it by weighting samples in the dataset inversely proportional to their probability of being missing, MI accomplishes it by imputing suitable values for missing entries and then running the analyses on the imputed datasets.

2.2 Missing Not at Random (MNAR)

The Missing not at Random assumption more formally is that there exists no observed x such that $y \perp M$ [Van18]. This once again causes the complete case distribution to be different than the underlying distribution resulting in biased complete case results; however, unlike the MAR case because there are no covariates that can explain the missingness, domain knowledge and assumptions stemming from that domain knowledge become more important. We will not be developing approaches for MNAR, but rather evaluating PMM, a MAR approach, in a MNAR setting. MNAR is a commonly appearing phenomenon and practitioners may sometimes make the wrong assumptions so it is worthwhile to get a good understanding of the MNAR assumption and how models built for the MAR assumption perform under MNAR.

2.3 Multiple Imputation

Multiple Imputation is a general approach to missing data in which several copies of a dataset with missingness are made, the missing entries are filled in with suitable values, analyses are performed on each copy, and then the analyses are pooled into a single result [Mur18]. Using multiple different copies where the missing entries take on multiple values allows the modelling of uncertainty in the potential missing values we impute. Additionally, the flexibility of the method opens it up to any other method which defines a probabilistic function from the covariates to the domain of the missing values. Because of this the second step of filling in missing values with suitable ones is the heart of Multiple Imputation and where different methods diverge. In this report we build off of Predictive Mean Matching, a Multiple Imputation technique for probabilistic imputation which requires no domain knowledge on the part of the statistical practitioner and is straightforward to implement. A more thorough review of other Multiple Imputation techniques can be found in [Mur18], [ALR17].

2.4 Predictive Mean Matching

Predictive Mean Matching (PMM) is a Multiple Imputation technique first proposed by Rubin [Rub86] and developed more fully by Little [Lit88]. It estimates a function f, parameterized by β from the completely observed covariates to covariates with missingness and then implements a variation of the following algorithm to generate an imputation for an observation with missingness x_i: 1) calculates $\hat{y} = f_{\hat{\beta}}(x_{i,obs})$ for entry x_i, 2) choose the k closest fully observed entries x_j

according to $d(f_{\hat{\beta}}(x_{i,obs}), f_{\hat{\beta}}(x_{j,obs}))$, 3) choose a single entry x' from the k entries either randomly or proportional to their distance with x_i, 4) use the observed values of x' in the missing covariates as the imputed values for x_i. Predictive Mean Matching has 4 main variants in the algorithm described above, Type 0 is the algorithm described above. Type 1 is the algorithm described above but chooses the k fully observed entries according to $d(f_{\dot{\beta}}(x_{i,obs}), f_{\hat{\beta}}(x_{j,obs}))$ where $\hat{\beta}$ is the estimate of the optimal parameters and $\dot{\beta}$ is either a draw from the parameters posterior or a perturbed version of $\hat{\beta}$ [Van18]. Type 2, uses the same $\dot{\beta}$ and chooses k fully observed entries according to $d(f_{\dot{\beta}}(x_{i,obs}), f_{\dot{\beta}}(x_{j,obs}))$. Finally Type 3, creates two draws $\dot{\beta}, \ddot{\beta}$ in the same procedure as type 1, then the k fully observed entries are chosen according to $d(f_{\dot{\beta}}(x_{i,obs}), f_{\ddot{\beta}}(x_{j,obs}))$ [Van18]. The default value for k is 5 in the mice package [vG11], but $k = 10$ is also a suggested possibility [Van18].

2.5 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a form of Artificial Nerural Network in which a sequence of interleaved convolutional layers and pooling layers are followed by a series of linear layers, culminating in a final layer specially designed for the task being trained, additionally, between every layer an activation function gets applied [KSH17]. The convolutional layers are n x n grids, where n is much smaller than the height or width of the image, such that each cell on the grid has a weight. The convolutional layer then get's applied to a n x n subsection of an image, multiplies each pixel value by the weight in the layer's corresponding cell, and then sums them all, the layer then strides over the image to perform that computation for various overlapping subsections of the input image [AMA17]. Convolutional layers can have multiple output channels in which case there are multiple grids of the same size that construct future layers independent of each other, in figure 2 each square in the stack after a convolutional layer is a single output channel [AMA17], for understanding what a channel is it grounding to realize that rgb images are 3 channels with one for each colour. Next, a pooling layer divides the image into overlapping sections by striding over the image and performs a computation for each section, a common computation for pooling layers is the max pool operation where for each section, the maximum value in it is passed forwards to the next layer [AMA17]. Both the convolutional and pooling layers can reduce or maintain the dimension of the image but only convolutional layers can increase the number of channels, [AMA17]. After the image is sent through the sequence of convolutional and padding layers it is passed through to a sequence of traditional fully connected layers, in the case of classification the last layer has as many nodes as classes and in the regression task the last layer has as many nodes as values being predicted per data point. In this way performing classification with CNN's becomes a simple task, after all it is what the breakthrough CNN AlexNet was created for [KSH17].

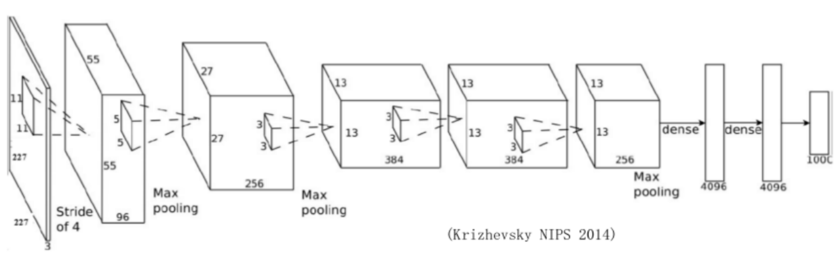

Figure 2: AlexNet, a CNN proposed in [KSH17] where each box mapping to a point is either the respective convolutional or pooling layer being applied to a specific subsection

3 Methodology: Predictive Mean Matching With Images

Recall the problem setting is imputing labels for image data when there are significant levels of missingness in the dataset. Here, we introduce some notation, let $\mathcal{D}$ be the dataset containing full observations and observations with missing labels, let $\mathcal{D}_o$ be the subset of $\mathcal{D}$ that is completely observed and $\mathcal{D}_m$ the subset of $\mathcal{D}$ which has missing labels, additionally, let C be the number of potential labels for images. Now for individual observations, for an rgb image i of width w, height h we define $x_i \in \mathbb{R}^w \times \mathbb{R}^h \times \mathbb{R}^3$ as the observation which contains every pixel in the image and the pixels three red, blue, or green values; $y_i = 1, 2, \ldots, C$ is the label of image i; and lastly, M_i is the missingness indicator for image i so $M_i = 1$ means y_i is observed and $M_i = 0$ means y_i is missing.

3.1 Naive Approach

The naive approach is to train a CNN to classify images and then use the CNN's predicted class probabilities to impute values. This is done by training a CNN classifier f_θ that maps images to class probabilities, so $f_\theta(x_i) \in \triangle^C$ ($\triangle^C$ is the probability simplex of size C). f_θ is trained by minimizing the cross entropy loss between predicted probabilities and real labels. This is still a Multiple Imputation method since it can generate a distribution over potential values for an entry with a missing label and those values can be used as copies in the MI setup; however, this approach is not PMM since there is no comparison to or selection from existing complete case entries. Additionally this approach only uses $\mathcal{D}_o$ to learn it's mapping and so it is susceptible to datasets with very few observed labels.

3.2 CNN Classifier

The CNN Classifier approach trains a classifier in the same way as described above but then uses the predicted probabilities, $f_\theta(x_i)$, for every image in the dataset to execute the PMM algorithm and generate multiple imputations. Like the naive approach this method also only uses samples in $\mathcal{D}_o$ and is susceptible to datasets with large portions of missingness

3.3 Convolutional Autoencoder

I define a convolutional autoencoder as a convolutional Neural Network with a bottleneck layer in the middle, The task for the neural net is to predict the red, blue, and green values for each pixel of a given input image. The bottleneck layer is what makes this a non-trivial problem since the image data must be compressed into a low-dimensional setting and then the compression must be "decoded" into a full image reconstruction. Hence, due to the forward nature of Artificial Neural Networks, the output at the bottleneck layer for an image x_i can serve as a low-dimensional encoding of that image. To get this autoencoder we learn two functions: g_ϕ, g_ψ that compose to map images $x_i \in \mathbb{R}^w \times \mathbb{R}^h \times \mathbb{R}^3$ back to themselves, so $g_\psi(g_\phi(x_i)) \in \mathbb{R}^w \times \mathbb{R}^h \times \mathbb{R}^3$. g_ϕ is a mapping from images to the bottleneck layer and g_ψ is a mapping from the bottleneck layer back to the image. ϕ, ψ are learned by using automatic differentiation and linear optimization to minimize the mean squared error between x_i and $g_\psi(g_\phi(x_i))$. Once ψ and ϕ are sufficiently learned we use g_ϕ to calculate the distances in the PMM algorithm and execute the multiple imputation strategy. Unlike the other two approaches this method uses the entire dataset $\mathcal{D}$ and so it's mapping should be more robust to high levels of missingness but there may be strong imbalances in the label distribution that then cause mis-imputations when choosing from the k closest images with observed labels.

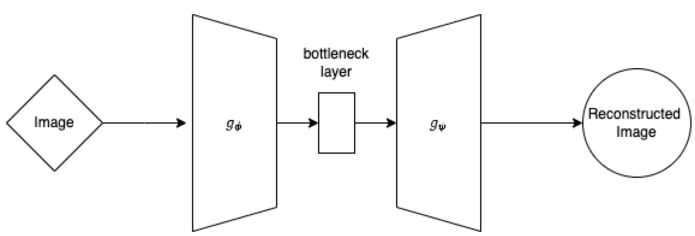

Figure 3: General Diagram of the architecture for the Convolutional Autoencoder

4 Experimental Setup

The experiments in this paper answer the following questions:

1. How do the three approaches perform in the MAR case?

2. How do the performances of the three approaches change as the level of missingness increases?
3. How do the three approaches perform when applied to MNAR data?
4. How do the three approaches perform as the level of missingness increases?
5. How do the three approaches perform as the variance between label-missingness probabilities decreases?

To create the synthetic MAR and MNAR data setting we will use the CIFAR-10 image datasets and augment them so that certain labels are missing according to different missing mechanisms be it the MAR or the MNAR setting. Because the classification algorithms operate on labelled data they are given more structure in learning their projection, but, at the caveat of needing sufficient fully observed entries in the data to learn that mapping. For this reason understanding the performances of the different algorithms with respect to the levels of missingness in the data is an important question to have answered for practitioners looking to apply PMM to image data. Then, in the MNAR case we look at how the approaches perform as level of missingness increases and how they perform as the variance between label missingness shrinks to build an understanding of the role of label imbalances in the performance of the models. Additionally we will experiment with the different types of PMM by slightly permuting the parameters of the learned neural nets and evaluating how each type performs for both the CNN classifier and the convolutional autoencoder. In each experiment, 5 simulations were run and the mean accuracy and variance in the accuracy measure from the runs are presented.

4.1 Generating MAR Data

Recall that the Missing Conditionally at Random assumption states that given the observed covariates, whether or not the label is missing is independent of the label. Hence any simple function which takes the pixel values of an image and returns a binary probability can be considered an MAR missingness generator. For this experiment we chose to take the squared sum of values across all colours and pixels and then institute a threshold τ such that images with values above the threshold went missing with probability p and all images with values below the threshold were observed. For the evaluation of the different types of PMMs, $\tau = 8000$ and $p = 0.7$ were chosen and for the experiment in which the level of missingness was increased τ slid over values 1000, 5000, 10000, and 14000 while $p = 0.7$ was held constant. Mathematically, letting the red, blue, green values of a pixel in location i, j be $r_{i,j}, b_{i,j}, g_{i,j}$ we can define the missingness function as:

$$H(x_i) = \sum_{i=0}^{w} \sum_{j=0}^{h} r_{i,j}^2 + b_{i,j}^2 + g_{i,j}^2$$

$$P(M|x_i) = \begin{cases} p & \text{if } H(x_i) > \tau \\ 0 & \text{otherwise} \end{cases}$$

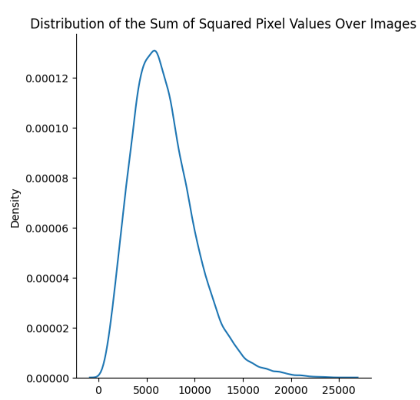

Figure 4: Distribution over all images of each image's sum of squared pixel values

4.2 Generating MNAR Data

Recall that the Missing Not at Random assumption requires that regardless of any of the observed covariates, whether or not an image label is missing is dependent on the image label. This form of missingness can be readily simulated by setting images in the CIFAR-10 dataset to be missing according to probabilities directly defined by the labels of the images. For the evaluation over the different types of PMM we used the following conditional probabilities $P(M_i|y_i)$ for each label below:

Class Label	$P(M\|y)$
Airplane	0.95
Automobile	0.6
Bird	0.4
Cat	0.75
Deer	0.6
Dog	0.7
Frog	0.1
Horse	0.8
Ship	0.2
Truck	0.6

For the experiment with increasing levels of missingness each $P(M|y)$ was drawn from a beta distribution with the parameters to the beta distribution skewing it more towards 1 as the amount of missingness increases. So the hyperparameters to the beta distribution define how the missingness was increased, we used beta parameters (1,1), (10,1), and (20,1) as well as (10,10) and (50,50) for another experiment to observe how the relative differences between the missingness per label affects the PMM algorithms.

5 Experimental Results

5.1 MAR: Evaluation of Methods as Missingness Increases

Table 1 below presents the results to the first and second question showing the performance of the three approaches as the level of missingness in the dataset is decreased by sliding τ into larger and larger values decreasing the number of images with non-zero missingness probability while keeping p constant.

τ	Convolutional Autoencoder	Convolutional Classifier PMM	Convolutional Classifier Prediction
1,000	0.248 ($\pm$0.0024)	0.110 ($\pm$0.0011)	0.111 ($\pm$0.00001)
5,000	0.250 ($\pm$0.0016)	0.104 ($\pm$0.0011)	0.059 ($\pm$0.000)
10,000	0.238 ($\pm$0.0042)	0.151 ($\pm$0.0022)	0.188 ($\pm$0.003)
14,000	0.268 ($\pm$0.0074)	0.095 ($\pm$0.0055)	0.096 ($\pm$0.0000)

Table 1: The mean and standard deviation for the accuracy of imputed values over 5 different imputations for the 3 different approaches as the missingness threshold is lowered (leading to more missingness in data) in the MAR setting

Here we see instantly that in all missingness settings the convolutional autoencoder massively outperforms the convolutional classifier using PMM or prediction. The convolutional autoencoder is able to learn about the general image space structure without the labels and learning this structure appears to be more important than learning a function from images to the variable we want to impute as seen in the strength of the autoencoder relative to the classifier models. Additionally, the amount of missingness does not appear to have a significant impact on the results of any of the three algorithms, the autoencoder still learns the same structure but has less labels to pull from in the PMM step and the classifiers have to train on smaller datasets, it is probable that these are simply bad cutoff points for missingness that cause us to miss a general relationship but these findings currently point towards the robustness of these methods in the face of increasing missingness. Lastly, the convolutional classifier using PMM appears to be about equal in imputation strength to the convolutional classifier with prediction in this MAR setting. This may not always be true, in the case that some class was significantly missing but the MAR assumption still held then we believe the convolutional classifier with PMM would work better than the classifier that predicts as other observed members of the high-missingness class would be given similarly wrong predicted probabilities.

5.2 MAR: Evaluation of Type 0, 1, 2, and 3 PMM

Table two shows the results of applying small random noise to the parameters in the models in a way that we accomplish type 1, 2, and 3 PMM. For this analysis we set $\tau = 8,000$ and $p = 0.7$

Modelling Approach	Type 0	Type 1	Type 2	Type 3
Convolutional Autoencoder	0.251 (± 0.0013)	0.206 (± 0.0029)	0.240 (± 0.003)	0.209 (± 0.0029)
Convolutional Classifier	0.105 (± 0.0022)	0.105 (± 0.00116)	0.105 (± 0.0025)	0.106 (± 0.0022)

Table 2: The mean and standard deviation for the accuracy of imputed values over 5 different imputations for the 2 PMM aproaches compared using the different types of PMM under the MAR setting

With Artificial Neural Networks there is not a posterior from which one can draw so instead we applied random noise with variance 0.01 to each parameter in the networks to get the $\dot{beta}, \ddot{beta}$ for type 1, 2, and 3 PMM. It seems that the different types of PMM doe not readily carry over in value to this application that uses convolutional Neural Networks in the MAR settings. The convolutional classifier's results are generally invariant to the types, but, the convolutional autoencoder performs significantly worse when two different functions are used to project the missing entry for imputation and the observed entries (type 1 and type 3 PMM). These results point towards the limits of how well PMM can be applied to images using convolutional Neural Networks as the weakness of the autoencoder in type 1 and type 3 shows not everything is easily translated from the traditional PMM setting to this deep learning use. Importantly, since Artificial Neural Networks don't have a posterior or meet traditional identifiability concerns jittering the parameters is a crude form of having uncertainty in the learned function which is what the different types of PMM are for.

5.3 MNAR: Evaluation of Methods as Missingness Distributions Change

Table three below shows the performance of the three approaches when the level of missingness per label increases as they are drawn from beta distributions which are increasingly skewed towards 1.

In the MNAR setting new patterns arise between the relation in the amount of missingness and the performance of the models. As seen the performance of the convolutional autoencoder weakens as the amount of missingness increases showing that it is more susceptible to increased amounts of missingness under the MNAR case. Once again the convolutional classifier performs very similarly when using the PMM algorithm or prediction to impute with even less variation between the two than in the MAR case. Interestingly the performance of the convolutional classifier methods improve as the level of missingness increases.

It is unclear if this is a genuine property of the algorithms or if they are performing better because the beta distribution pushes missingness probabilities for each label to me more concentrated leading to a more equal distribution between labels and so class imbalances that would result in poor performance do not occur. To test this hypothesis we ran the same experiment with beta parameters (1,1), (10,10), and (50,50) the results are shown in table 4 below. Key to this experiment is that the expected amount

Beta Parameters	Convolutional Autoencoder	Convolutional Classifier PMM	Convolutional Classifier Prediction
$\alpha = 1, \beta = 1$	0.173 ($\pm$0.0016)	0.054 ($\pm$0.0014)	0.006 ($\pm$0.0000)
$\alpha = 10, \beta = 1$	0.122 ($\pm$0.0013)	0.095 ($\pm$0.0011)	0.087 ($\pm$0.000)
$\alpha = 20, \beta = 1$	0.120 ($\pm$0.0001)	0.095 ($\pm$0.00007)	0.101 ($\pm$0.00)

Table 3: The mean and standard deviation for the accuracy of imputed values over 5 different imputations for the 3 different approaches as beta distribution parameters increase how much missingness there is. done in the MNAR setting

of missingness is equal across the three situations but the dispersion between each labels probability of being missing shrinks as the parameters go from (1,1) to (50,50), hence if the results improve credibility will be given to the likelihood that it is not the amount of missingness that led to better results in the previous table but the variance between each labels likelihood of being missing. In general it seems once again not clear as to if there exists a relationship between the performance of these algorithms as the amount of missingness increases in the MNAR case.

Beta Parameters	Convolutional Autoencoder	Convolutional Classifier PMM	Convolutional Classifier Prediction
$\alpha = 1, \beta = 1$	0.212 ($\pm$0.0012)	0.084 ($\pm$0.0008)	0.01 ($\pm$0.0000)
$\alpha = 10, \beta = 10$	0.096 ($\pm$0.0008)	0.066 ($\pm$0.0014)	0.046 ($\pm$0.000)
$\alpha = 50, \beta = 50$	0.100 ($\pm$0.0016)	0.101 ($\pm$0.0011)	0.0922 ($\pm$0.00)

Table 4: The mean and standard deviation for the accuracy of imputed values over 5 different imputations for the 3 PMM aproaches compared using the different beta distributions to explore how the variance in the missingness probabilities given the labels impacts the performance of the approaches

The results from this study suggest that it was indeed the variance between the probabilities of being missing given the labels that contributed to the pattern of increasing performance for the convolutional classifiers and decreasing performance for the convolutional autoencoder. This points towards some cases in which MNAR may be true but the data appears to be very similar to MCAR and results of modelling as if it was MCAR (classifier using prediction) are equal to modelling as if it was MAR (autoencoder or classifier using PMM).

5.4 MNAR: Evaluation of Type 0, 1, 2, and 3 PMM

Table 5 shows the results of permuting the parameters in the two PMM models so that they resemble type 1, 2 and 3 PMM.

Modelling Approach	Type 0	Type 1	Type 2	Type 3
Convolutional Autoencoder	0.198 ($\pm$0.0019)	0.146 ($\pm$0.0018)	0.188 ($\pm$0.0017)	0.122 ($\pm$0.0007)
Convolutional Classifier PMM	0.192 ($\pm$0.0011)	0.188 ($\pm$0.0002)	0.186 ($\pm$0.0002)	0.194 ($\pm$0.0006)

Figure 5: The mean and standard deviation for the accuracy of imputed values over 5 different imputations for the 2 PMM aproaches compared using the different types of PMM in the MNAR setting

Once again the same patterns observed for the MAR case persist into the MNAR setting. convolutional autoencoders continue to do significantly worse when taking the distance of two values that were projected using different encoding functions. The convolutional classifier using PMM has it's accuracy being somewhat invariant to the types but the variance in its accuracy increases as the we go from type 0 towards type 3 which is to be expected as each parameter gets more and more noise when progressing through each type.

6 Discussion

The strength of the convolutional autoencoder in this case is largely due to the association between what an images pixel values take on and the subject of the image, but, there may be cases in which the value that is missing does not have such a direct connection and the autoencoder performs worse than the classifier. One such example may be a dataset of drawings and the value that is missing is some judges score which is based on the political message that the piece is communicating. In this case a representation based purely in the image is unlikely to do well since drawings may be parodies of past artpieces in which they directly attack the political message of the art they are parodying, but, since the images are grouped together by similarity, PMM in this case is likely to mis-impute the value for the parodical art piece, while, the classifier may pick up on the subtleties that parodies have which make then parodical from the observed judges scores which the autoencoder is not trained on. Hence, great attention to the relationship between fully observed covariates and the covariates with missingness must be paid as it can greatly influence which of the approaches works best.

7 Conclusion

Predictive Mean Matching has been shown to be an effective strategy for multiple imputation in tabular data and this report takes the main ideas, reshapes the implementation details to work with images and shows that the same principles underlying Predictive Mean Matching also lead to good imputations for image data with missing labels. In comparing the convolutional autoencoder to the convolutional classifier this report shows that the key to the success of predictive mean matching is the measuring of distance between observed and missing values and how thinking about Predictive Mean Matching in that perspective allows for it to be applied to entirely new data paradigms.

This report develops two new approaches for the multiple imputation of image labels and evaluates the accuracy of these approaches compared to a naive classifier prediction strategy. Multiple Imputation is one of the most natural reactions to the presence of missing data but missingness comes in a variety of ways and it is important to have a large toolset of strategies and an understanding of how those strategies perform under optimal and unoptimal situations. This report adds to the current toolset by proposing two new approaches for the multiple imputation of image labels under MAR settings that leverage the notions of existing strategies for dealing with tabular MAR data. We show that within the Predictive Mean Matching framework there are a variety of ways in which distances between images can be taken by focusing on two methods that first project them into lower dimensional spaces. By using a convolutional autoencoder and a convolutional classifier this report illustrates that methods not grounded in learning representations with relation to the label can still learn representations more valuable than those learned by a classifier. Overall, this work on representation learning and it's connection to Multiple Imputation through Predictive Mean Matching opens the door to future work which leverages self-supervised ans semi-self-supervised representation learning for missing data which uses both observed and missing entries to learn better representations that can then be used to impute missing values.

Additionally, this report explores the relation between the proposed methods and different missing mechanisms, missingness distributions, and between-label missingness variation. This report shows how the methods are generally robust to increased levels of missingness. This report shows the effectiveness of the different approaches in the MAR setting they were made for and the MNAR setting they may be accidentally used in and shows that in the MNAR setting the variation between class missingness probabilities has noticeable impacts to the performance of all three approaches.

Finally this report shows that while the main idea from Predictive Mean Matching of using distances between observed and missing entries in a dataset does carry over successfully to convolutional Neural Networks, the four types of Predictive Mean Matching do not have such readily available counterparts in the Artificial Neural Network literature due to the lack of posteriors over a networks parameters. There has been work in ANNs that aim to infuse Bayesian modelling of uncertainty into the models parameters but more research will have to be done to explore how those models can be used in PMM-esque algorithms and whether they have readily available equivalents to type 1, 2, and 3 predictive mean matching.

8 Future Work

This work can be extended and puts a spotlight on bridges between deep learning and missing data methods as distinct possibilities for statisticians to deal with missingness in the age of big data. One of the main message of this report is that all one needs to perform Predictive Mean Matching is a mapping to a metric space, for this reason one interesting direction for future research is to look at how PMM performs under different mappings into metric spaces, this report provides two in the form of the autoencoder and classifier but there exist other approaches that try to maintain local and global structure of high-dimensional data in lower-dimensional spaces such as t-SNE or UMAP which could be usefully combined with PMM to generate suitable imputations. Additionally, there are other ideas in missing data which are compatible with machine learning, inverse probability weighting can be made to be compatible with deep learning techniques for regression by weighting the mean squared error by the inverse probability weight, hopefully this report helps push the idea that working at the intersection of machine learning and missing data approaches can provide valuable methods for missingness in big data. For improving upon the approaches introduced in this report we see two main directions for future research, the first is to augment the models with self-supervised goals like those already seen in image learning where an image is slightly altered and the distance between the projection of the image and it's altered version is minimized. The second is to experiment with how the results of this report change with respect to scaling up both the network architectures, the size of the images, and the number of labels which each image can take, that will make the problem significantly harder but more in touch with real world applications of image datasets in which there is often more information that companies want than just a few labels. Lastly, this report provided easy ways in which the CIFAR-10 dataset can be made into a MAR dataset or an MNAR dataset but the method provided for converting it into an MNAR dataset meant that given y, we had $M \perp x$, research into a missingness mechanism that is directly dependent on both x and y will help the development of algorithms that are as robust as possible to MNAR settings and further the understanding of MAR approaches under different MNAR settings.

References

[Rub86] Donald B Rubin. "Statistical matching using file concatenation with adjusted weights and multiple imputations". In: *Journal of Business & Economic Statistics* 4.1 (1986), pp. 87–94.

[Lit88] Roderick JA Little. "Missing-data adjustments in large surveys". In: *Journal of Business & Economic Statistics* 6.3 (1988), pp. 287–296.

[Roy04] Patrick Royston. "Multiple imputation of missing values". In: *The Stata Journal* 4.3 (2004), pp. 227–241.

[ZLK10] Yan Zhou, Roderick J. A. Little, and John D. Kalbfleisch. "Block-Conditional Missing at Random Models for Missing Data". In: *Statistical Science* 25.4 (Nov. 2010). DOI: 10.1214/10-sts344. URL: https://doi.org/10.1214%2F10-sts344.

[vG11] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03.

[SW13] Shaun R Seaman and Ian R White. "Review of inverse probability weighting for dealing with missing data". In: *Statistical methods in medical research* 22.3 (2013), pp. 278–295.

[ALR17] Olanrewaju Akande, Fan Li, and Jerome Reiter. "An Empirical Comparison of Multiple Imputation Methods for Categorical Data". In: *The American Statistician* 71.2 (2017), pp. 162–170. DOI: 10.1080/00031305.2016.1277158. eprint: https://doi.org/10.1080/00031305.2016.1277158. URL: https://doi.org/10.1080/00031305.2016.1277158.

[AMA17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[Mur18] Jared S. Murray. "Multiple Imputation: A Review of Practical and Theoretical Findings". In: *Statistical Science* 33.2 (2018), pp. 142–159. DOI: 10.1214/18-STS644. URL: https://doi.org/10.1214/18-STS644.

[Van18] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.