Data Science Homework 3

Andrés Ponce, **彭思安** P76107116

April 4, 2022

1 Introduction

Current machine learning and deep learning models have shown impressive abilities to learn patterns from data. From object recognition [1] to text generation [2], many different fields have been influenced by machine learning. These models will often take a data sample as well as its label, and constantly adjust its weights to minimize the loss function. Machine learning models often require large amounts of data, which they use to learn the patterns that will be used when they are tested. A constant issue with current models is obtaining a large enough amount of data which accurately represents the data the model will encounter after training.

When trying to train a model, an important first step is ensuring the quality of the input data. Due to the messy and often chaotic nature of data collection in real applications, a preprocessing step is necessary to ensure data quality. Properties of the data such as the mean and standard deviation can tell us a lot about the nature of the data. One issue that is not so straightforward to solve is missing data. Sometimes data is missing completely at random, with no relation between the missing pieces of data, while other times some factor has an influence on the missing attributes.

To address missing data several *imputation* methods have been developed, where missing data attributes are filled in using some other properties of the data. Some methods rely on statistical properties such as the mean, while others use clustering techniques to fill in the missing data. Still others use an entire neural network to estimate the missing values.

The present assignment investigates different data imputation methods given different datasets and different amounts of missing data in each to determine the effectiveness of these methods. First, we introduce the methods used, followed a discussion on the experiments. Finally, we discuss the results of the experiments and provide our conclusions.

2 Methods

2.1 Mean

Perhaps the simplest imputation method we tested is using the mean. In this method, the mean along the columns of the non-missing data is used to fill in all the missing values of that column. Using the column means will preserve the means of that column, which might

be desirable since we would not want to change the properties of the data. Mean calculation can also be efficiently done in numerical libraries, and since we only calculate the mean once per column in our dataset, mean imputation was the quickest method we tested. This speed could be reason enough to choose mean imputation.

2.2 K-Nearest Neighbors

the k-nearest neighbors algorithm has been widely used in unsupervised learning scenarios where we assign a label to a data point based on the value of the k closest points. With imputation, we assign the mean value of the k nearest points, where "closeness" is measured in our experiments using a modified version of the euclidean distance.

While running the algorithm, we impute a different value to each missing number. For every missing number, we must find its closest k points and calculate their distance. Since we impute a different value for each missing number, this method performs better than the mean imputation. However, since we have to impute each missing vlaue individually, the running time also increases using this method.

2.3 Multivariate Imputation by Chained Equations

In this method, the columns with missing values are treated as a function of the other columns [3]. This method essentially treats imputation as a regression problem with the other columns as inputs.

At each iteration, we choose a feature column with missing values y as the regression target, and the other columns x as the inputs. Then a linear regressor is trained on the known values of y, and this regressor is then used to impute the missing values of y. We can repeat this process multiple times for a more accurate regressor.

A potential problem with this approach is that as the percentage of missing values grows, the regressor will have less and less data to train with, which might potentially result in lower accuracy. If all the data is distributed similarly, training a regressor could be a quick way to impute values with low percentages of missing values. With linear regression we should expect a more accurate guess than with mean imputation, especially at lower missing percentages.

2.4 Missing Data Imputation using Generative Adversarial Nets

This method [4] uses the GAN architecture [5] to impute missing values. The architecture consists of two competing networks, the Generator G which takes a vector with missing values and attempts an imputation. The Discriminator G then attempts to determine which values in the vector have been imputed by G and which come from the original input.

D also takes in an input called the *hint* H, which specifies the probability that a specific sample was observed in the conditional probability $P(\boldsymbol{X}|\hat{\boldsymbol{X}} = \hat{\boldsymbol{x}})$. The training objective is then

$$V(D, G) = \mathbb{E}_{\hat{\boldsymbol{X}},\boldsymbol{M},\boldsymbol{H}}[\boldsymbol{M}^T \log D(\hat{\boldsymbol{X}}, \boldsymbol{H}) + (1 - \boldsymbol{M})^T \log(1 - D(\hat{\boldsymbol{X}}, \boldsymbol{H}))]$$

3 Experimental Analysis

4 Conclusions

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[4] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning*, PMLR, 2018, pp. 5689–5698.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.