Report on final paper in STAT 7331*

GAIN: Missing Data Imputation using Generative Adversarial Nets

Léon Yuan

2023-11-30

> **Tip**
>
> This report summaries the paper: GAIN: Missing Data Imputation using Generative Adversarial Nets. Visual illustration of this approach and Experimental Table are attached to the end of this report.

The paper that I am going to discuss and summarize is called GAIN: Missing Data Imputation using Generative Adversarial Nets (Yoon, Jordon, and Schaar 2018). There are so many popular and traditional statistical imputation methods on the market such as MICE (van Buuren and Groothuis-Oudshoorn 2011), MissForest, and Expectation Maximization. However, nowadays as the computation power becomes accessible and its cost is reduced a lot, the more complexity algorithms is doable to be realized in imputation area.

This novel approach has two essential components. One is called the **generator** and the other is called **discriminator**. The generator imputes the missing values while the discriminator tells which values are imputed by the generator. The goal is to maximize the ability of the discriminator to distinguish and the ability of the generator to fool the discriminator.

The following notations and equations are simplified version of the original paper because there are too many technique details in the full theoretical analysis. The generator function is defined as following:

$$X_{imp} = G(X_{ori}, M, (1-M)Z)$$

$$X_{com} = MX_{ori} + (1-M)X_{imp}$$

where X_{ori} is a d-dimensional original data vector with observed and unobserved data, M is the d-dimensional missing indicator vector that 1 means observed and 0 means unobserved, and X_{imp} is the d-dimensional imputed vector even for observed components. Z is a random d-dimensional vector independent of any X. X_{com} is the d-dimensional complete data vector.

*Thank Dr. Heitjan for teaching this course this semester. Happy Final!

The discriminator function is to distinguish which components of such X_{com} are truly observed and which components of them are later imputed by the generator function defined as above. The goal of such discriminator is to predict the d-dimensional missing vector M. The good aspect is that ground true M is known to us, which means this algorithm becomes supervised learning. This helps a lot on the final accuracy. There is another important **input** to the discriminator function. That is called **Hint** H which provides partial information of missing vector M to help discriminator distinguish the observed and imputed components. This d-dimensional Hint vector is draw from the customized distribution $H|M$ as following: $H|M = B \odot M + 0.5(\mathbf{1} - B)$ where

$$B_j = \begin{cases} 1 \text{ if } j \neq k \\ 0 \text{ if } j = k \end{cases}$$

for every j random sampling k from $1 \cdots d$. The paper has a Theorem and Proposition that assert if H is independent of M then generator is unable to learn the true data generating process. The whole discriminator function is defined as following: $D(X_{com}, h)$. The output of a discriminator function is a d-dimensional vector and each element of it is the predicted probability of being observed.

The final object is to first train the discriminator to maximize its ability to distinguish the observed and imputed components. Then it is to train the generator to minimize the ability of such discriminator. The object function is defined as following:

$$\min_G \max_D Q(D, G) = E_{Q(X_{ori}),M,H} \left[M^T \log D(G(X_{ori}), H) + (\mathbf{1} - M)^T \log(\mathbf{1} - D(G(X_{ori}), H)) \right]$$

where $Q(D, Q)$ is the final loss function that takes discriminator and generator as input then returns a scalar for loss measurement. $\mathbf{1}$ is a d-dimensional vector of all 1.

> **Important**
>
> This novel approach of imputing missing values is based on the assumption that MDM is MCAR.

This min max optimization approach is solved by iterative update algorithm (Goodfellow et al. 2014).

> **Note**
>
> Both generator and discriminator are fully connected neural networks.

The discriminator net is first trained by **mini-batch stochastic gradient descent** algorithm as following:

$$\max_D \sum_{i=1}^{batch-size} \left[\sum_{j:B_j=0} (m_{ij} \log(D(\hat{x}_i, m_i)_j) + (1 - m_{ij}) \log(1 - D(\hat{x}_i, m_i)_j)) \right]$$

The generator net is then trained by **mini-batch stochastic gradient descent** algorithm as following:

$$\min_{G} \sum_{i=1}^{batch-size} \left[- \sum_{j:B_j=0} (1 - m_{ij}) \log \left(D(\hat{x}_i, m_i)_j \right) + \alpha \sum_{w=1}^{d} m_{iw} (x_{iw}^{ori} - x_{iw}^{imp})^2 \right]$$

where $batch-size$ and α are two of hyper parameters. These two nets are trained on $j : B_j = 0$ because the generator needs to learn information where Hint H vector is 0.5 instead of when $H = 1or0$. The term $m_{iw}(x_{iw}^{ori} - x_{iw}^{imp})^2$ ensures that output values of observed components by generators are as close to the observed values as possible.

This new proposed imputation method is tested and compared against other popular imputation methods such as MICE, Matrix Completion, MissForest, Auto-Encoder and Expectation-Maximization(EM) across different dataset. 20% of the full data is removed from each test dataset to form **MCAR**. The comparison is repeated 10 times and within each time, 5-fold cross validation is applied to measure each RMSE and standard deviation. As we can observe, at least on these 5 selected dataset, GAIN outperforms all other methods in terms of RMSE and Standard Deviation. This paper also presents that GAIN outperforms other methods on post imputation prediction. GAIN also shows very good congeniality property when the ordinary logistical regression is applied on the post imputation dataset compared to the complete dataset.

The experimental table and architecture of GAIN are displayed in the attachment section.

Attachment

The full mini-batch SGD architecture of this apporach

compare GAIN with other methods on imputation performace

References

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Networks." https://arxiv.org/abs/1406.2661.

van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "mice: Multivariate Imputation by Chained Equations in r." *Journal of Statistical Software* 45 (3): 1–67. https://doi.org/10.18637/jss.v045.i03.

Yoon, Jinsung, James Jordon, and Mihaela van der Schaar. 2018. "GAIN: Missing Data Imputation Using Generative Adversarial Nets." https://arxiv.org/abs/1806.02920.

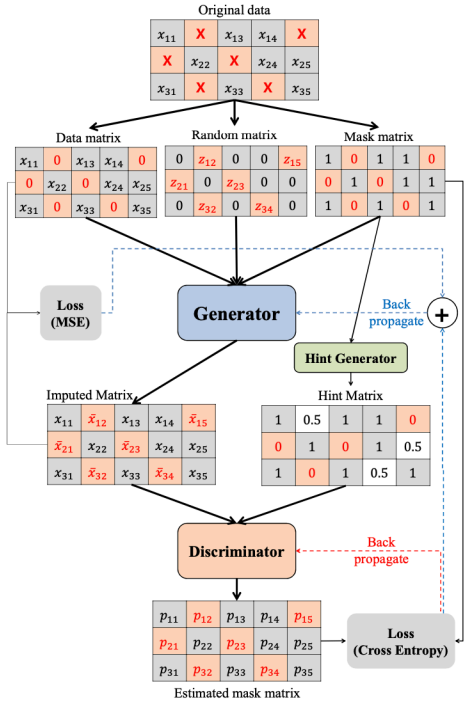

Figure 1. The architecture of GAIN

Figure 1: This graph is cited from the original paper

Table 2. Imputation performance in terms of RMSE (Average ± Std of RMSE)

Algorithm	Breast	Spam	Letter	Credit	News
GAIN	**.0546 ± .0006**	**.0513± .0016**	**.1198± .0005**	**.1858 ± .0010**	**.1441 ± .0007**
MICE	.0646 ± .0028	.0699 ± .0010	.1537 ± .0006	.2585 ± .0011	.1763 ± .0007
MissForest	.0608 ± .0013	.0553 ± .0013	.1605 ± .0004	.1976 ± .0015	.1623 ± 0.012
Matrix	.0946 ± .0020	.0542 ± .0006	.1442 ± .0006	.2602 ± .0073	.2282 ± .0005
Auto-encoder	.0697 ± .0018	.0670 ± .0030	.1351 ± .0009	.2388 ± .0005	.1667 ± .0014
EM	.0634 ± .0021	.0712 ± .0012	.1563 ± .0012	.2604 ± .0015	.1912 ± .0011

Figure 2: This table is cited from the original paper