2019.02.15

Missing Data Imputation using Generative Adversarial Nets

이지윤

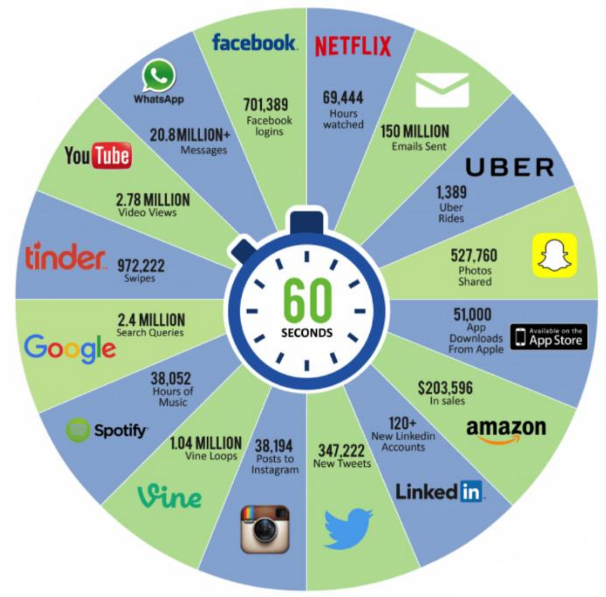
Contents

Introduction

– Data completeness

❖ 대용량 데이터 출현

- 사물 지능 통신(M2M) 확산에 따른 센서 데이터 증대
- 기업의 고객 데이터 트래킹/수집 행위 증가
- 소셜 네트워크 서비스(SNS)의 급격한 확산과 비정형 데이터의 폭증
- 데이터 저장 매체 가격 하락



이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

Introduction

– Data completeness

❖ 데이터 완결성

- 데이터 분석 알고리즘의 대표적인 가정은 데이터 완결성
- 데이터 집합에 속하는 모든 개체의 속성 값이 빠짐 없이 존재하는 것 의미
- 그러나, 현실에서 수집한 데이터들은 다양한 이유로 인해 결측치 존재
  - – 환자의 사망, 장비 오작동, 수신자 응답 거부
  - – 데이터 처리 효율성 저하, 분석의 어려움, 편향된 데이터 구조, 예측 성능 감소

〈완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

계측 데이터(observed data)

결측치 데이터(missing value data) → 대체 데이터(imputed data)

Introduction

– Data completeness

❖ 데이터 완결성

- ➢ 데이터 분석 알고리즘의 대표적인 가정은 데이터 완결성
- ➢ 데이터 집합에 속하는 모든 개체의 속성 값이 빠짐 없이 존재하는 것 의미
- ➢ 그러나, 현실에서 수집한 데이터들은 다양한 이유로 인해 결측치 존재
  - – 환자의 사망, 장비 오작동, 수신자 응답 거부
  - – 데이터 처리 효율성 저하, 분석의 어려움, 편향된 데이터 구조, 예측 성능 감소

〈완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

계측 데이터(observed data)

결측치 데이터(missing value data) → 대체 데이터(imputed data)

Introduction

– Data completeness

❖ 데이터 완결성

- 데이터 분석 알고리즘의 대표적인 가정은 데이터 완결성
- 데이터 집합에 속하는 모든 개체의 속성 값이 빠짐 없이 존재하는 것 의미
- 그러나, 현실에서 수집한 데이터들은 다양한 이유로 인해 결측치 존재
  - – 환자의 사망, 장비 오작동, 수신자 응답 거부
  - – 데이터 처리 효율성 저하, 분석의 어려움, 편향된 데이터 구조, 예측 성능 감소

〈완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

계측 데이터(observed data)

결측치 데이터(missing value data) → 대체 데이터(imputed data)

Introduction

– Data completeness

❖ 데이터 완결성

- ➢ 데이터 분석 알고리즘의 대표적인 가정은 데이터 완결성
- ➢ 데이터 집합에 속하는 모든 개체의 속성 값이 빠짐 없이 존재하는 것 의미
- ➢ 그러나, 현실에서 수집한 데이터들은 다양한 이유로 인해 결측치 존재
  - – 환자의 사망, 장비 오작동, 수신자 응답 거부
  - – 데이터 처리 효율성 저하, 분석의 어려움, 편향된 데이터 구조, 예측 성능 감소

〈완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	0	0

계측 데이터(observed data)

결측치 데이터(missing value data) → 대체 데이터(imputed data)

Introduction

– Missing value imputation

❖ 결측치 종류

(1) MCAR(missing completely at random)

가장 높은 수준의 임의성을 나타내는 결측치로서, 특정 개체의 특정 속성에 결측치가 발생할 확률은 해당 속성의 값이나 해당 개체의 다른 속성 값들에 영향을 미치지 않음 → 결측치 대체 연구 배경

Bulb ID	Brightness
A	130
B	115
C	N/A

Missing because of the bulb life

(2) MAR(missing at random)

중간 수준의 임의성을 나타내는 결측치로서, 특정 개체의 특정 속성에 결측치가 발생할 확률은 해당 개체의 다른 속성 값으로부터 영향을 받지만, 해당 속성의 값에는 영향을 받지 않음

Bulb ID	Brightness	Temp.
A	130	24
B	115	23
C	N/A	37

When the temp is high, missing has occurred.
=Missing depends on other variables

(3) MNAR(missing not at random)

가장 낮은 수준의 임의성을 나타내는 결측치로서, 특정 개체의 특정 속성에 결측치가 발생할 확률은 해당 속성의 값에 영향을 받는 경우

Bulb ID	Brightness
A	130
B	115
C	N/A

When the brightness is high, missing has occurred.
=A certain range has missing

Introduction

– Missing value imputation

❖ 알고리즘에 대한 종속성 여부와 결측치 대체 방식을 통해 구분

알고리즘 내부에서
결측치 대체

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

알고리즘 외부에서
독립적으로 결측치 대체

- Deletion
- Simple imputation (mean/median)
- Machine learning based imputation algorithm
  - Discriminative methods
  - Generative methods

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

Introduction

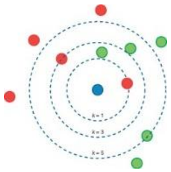
– Missing value imputation

❖ 알고리즘에 대한 종속성 여부와 결측치 대체 방식을 통해 구분

알고리즘 내부에서 결측치 대체

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

- Classification and regression tree(CART)
- Naïve Bayesian Classifier
- K-Nearest Neighbor



→ 상대적으로 단순한 기법, 추가적인 프로세스 필요 없음

→ 큰 예측 성능 향상 기대하기 어려움

알고리즘 외부에서 독립적으로 결측치 대체

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

- Deletion
- Simple imputation (mean/median)
- Machine learning based imputation algorithm
  - Discriminative methods
  - Generative methods

Introduction

– Missing value imputation

❖ 알고리즘에 대한 종속성 여부와 결측치 대체 방식을 통해 구분

알고리즘 내부에서
결측치 대체

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

알고리즘 외부에서
독립적으로 결측치 대체

- Deletion
- Simple imputation (mean/median)
- Machine learning based imputation algorithm
  - Discriminative methods
  - Generative methods

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

Introduction

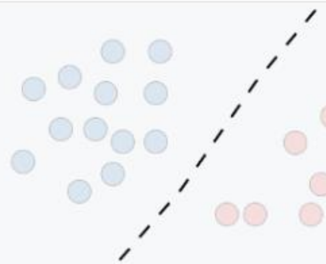
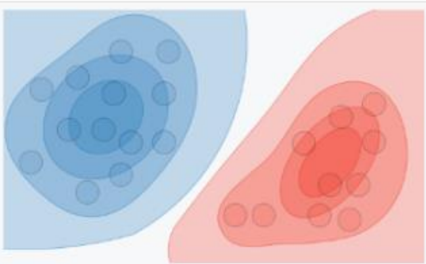
– Discriminative models vs Generative models

❖ Discriminative models based method

- ➢ Learn a function that maps the input x to an output y
- ➢ Conditional probability $p(y|x)$
- ➢ MICE, MIssForest

❖ Generative models based method

- ➢ Tries to learn a joint probability of the input x and the output y at the same time
- ➢ Joint probability $p(x, y)$
- ➢ EM, DAE, GAN

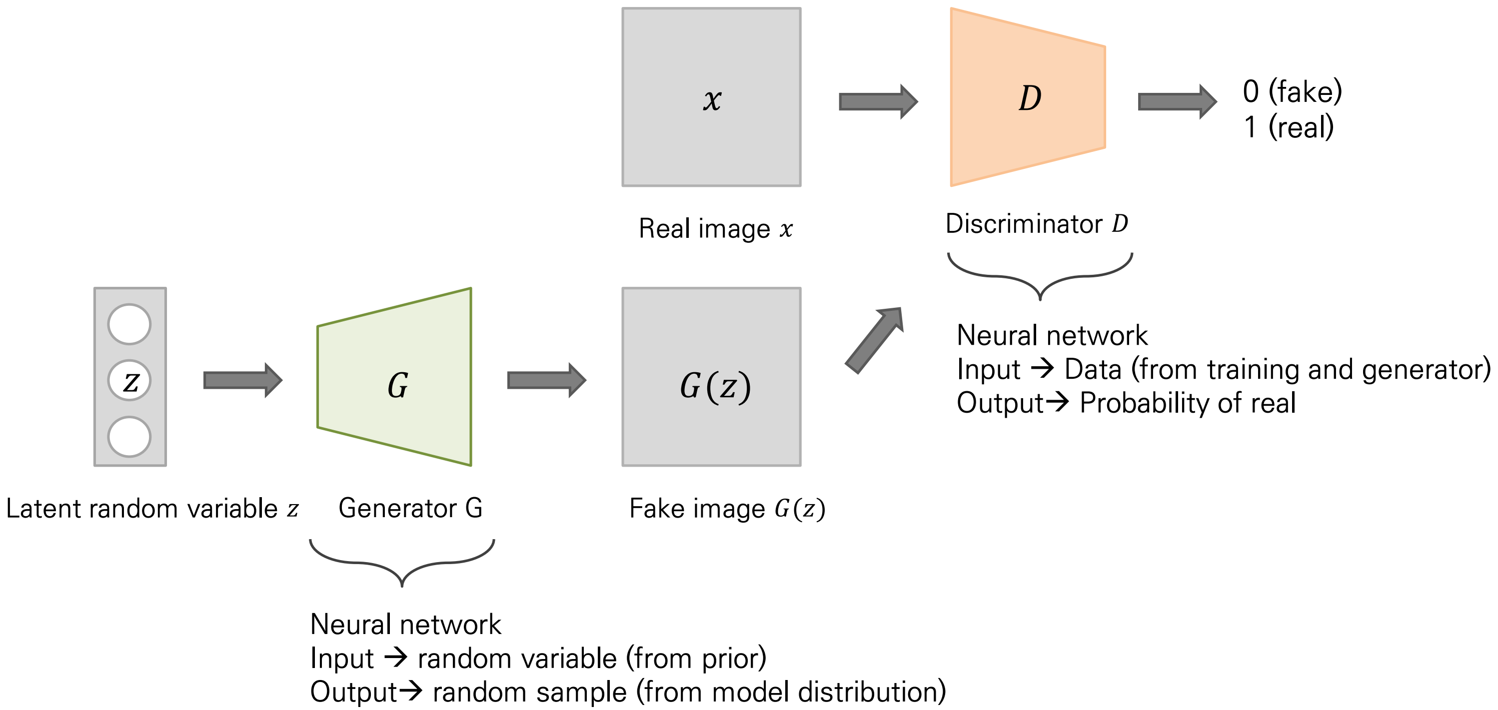
	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		

Literature Reviews

– Generative Adversarial Nets

❖ GAN은 두개의 네트워크로 구성

- Generator(G): 진짜 같은 가짜(fake)를 생성하는 네트워크
- Discriminator(D): 가짜(fake)와 진짜(real)를 구별하는 네트워크



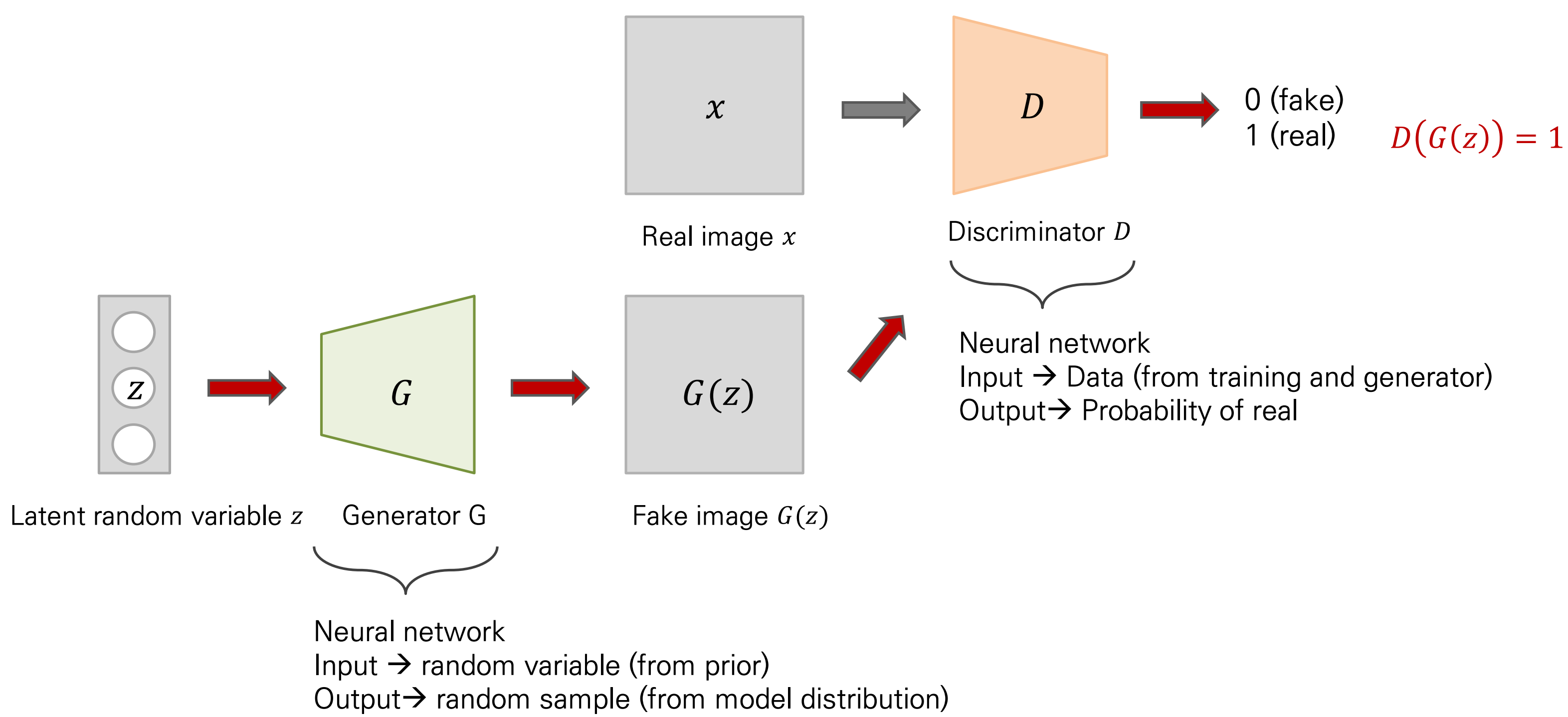
Literature Reviews

- Generative Adversarial Nets

❖ GAN은 두개의 네트워크로 구성

➢ Generator(G) 목적: $D\big(G(z)\big) = 1$

→ 진짜 같은 가짜를 생성



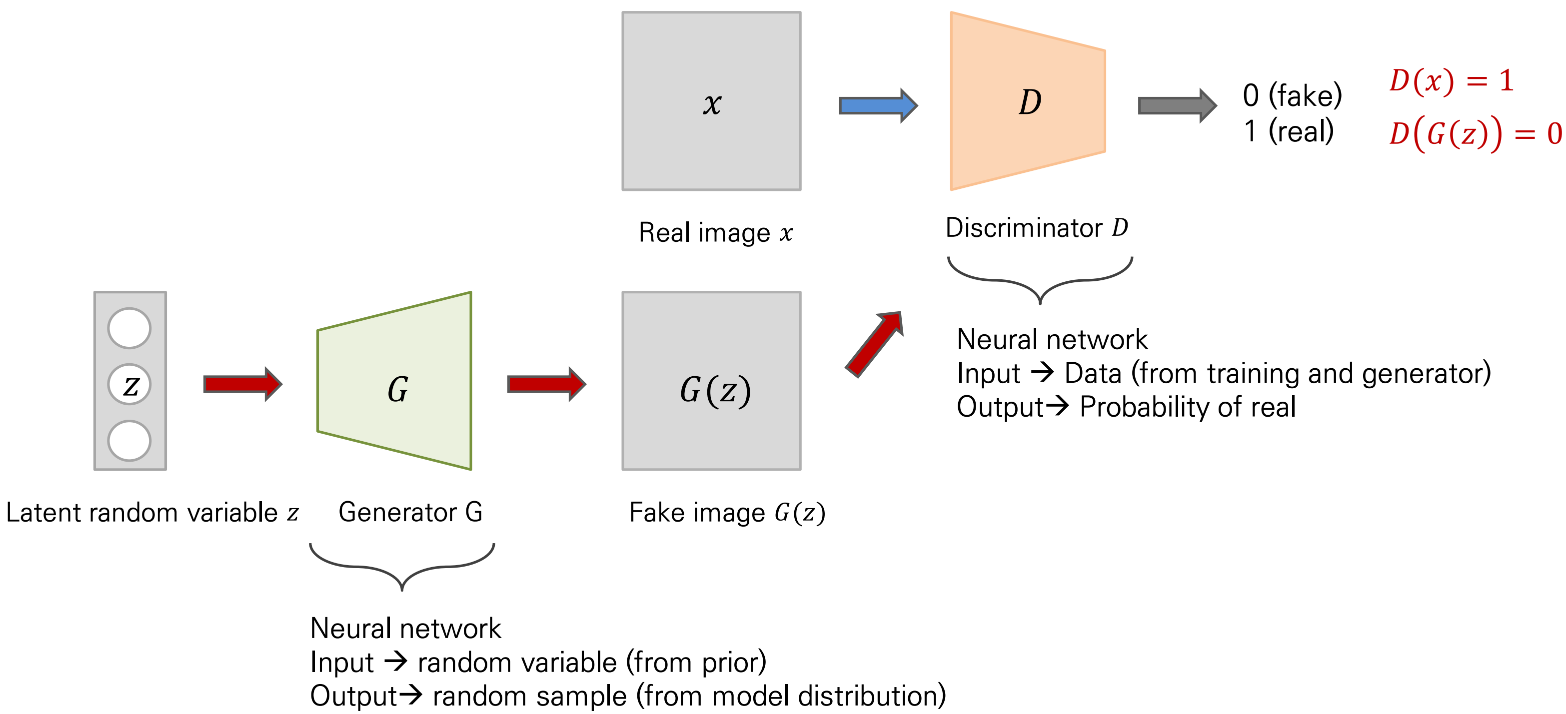
Literature Reviews

– Generative Adversarial Nets

❖ GAN은 두개의 네트워크로 구성

➢ Discriminator(D) 목적: $D\big(G(z)\big) = 0,\ D(x) = 1$

→ 진짜와 가짜를 정확히 구분

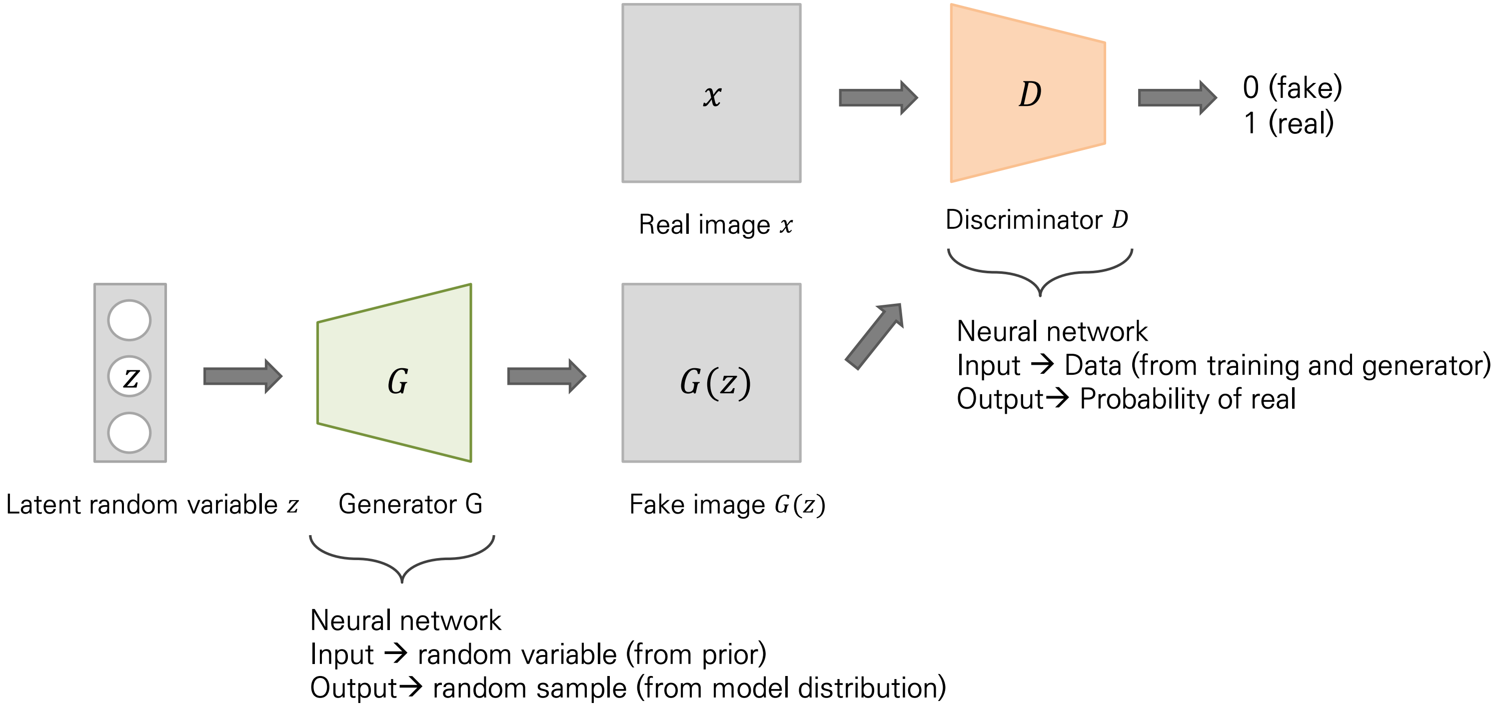


Literature Reviews

– Generative Adversarial Nets

❖ GAN objective function (adversarial)

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$



Literature Reviews

– Generative Adversarial Nets

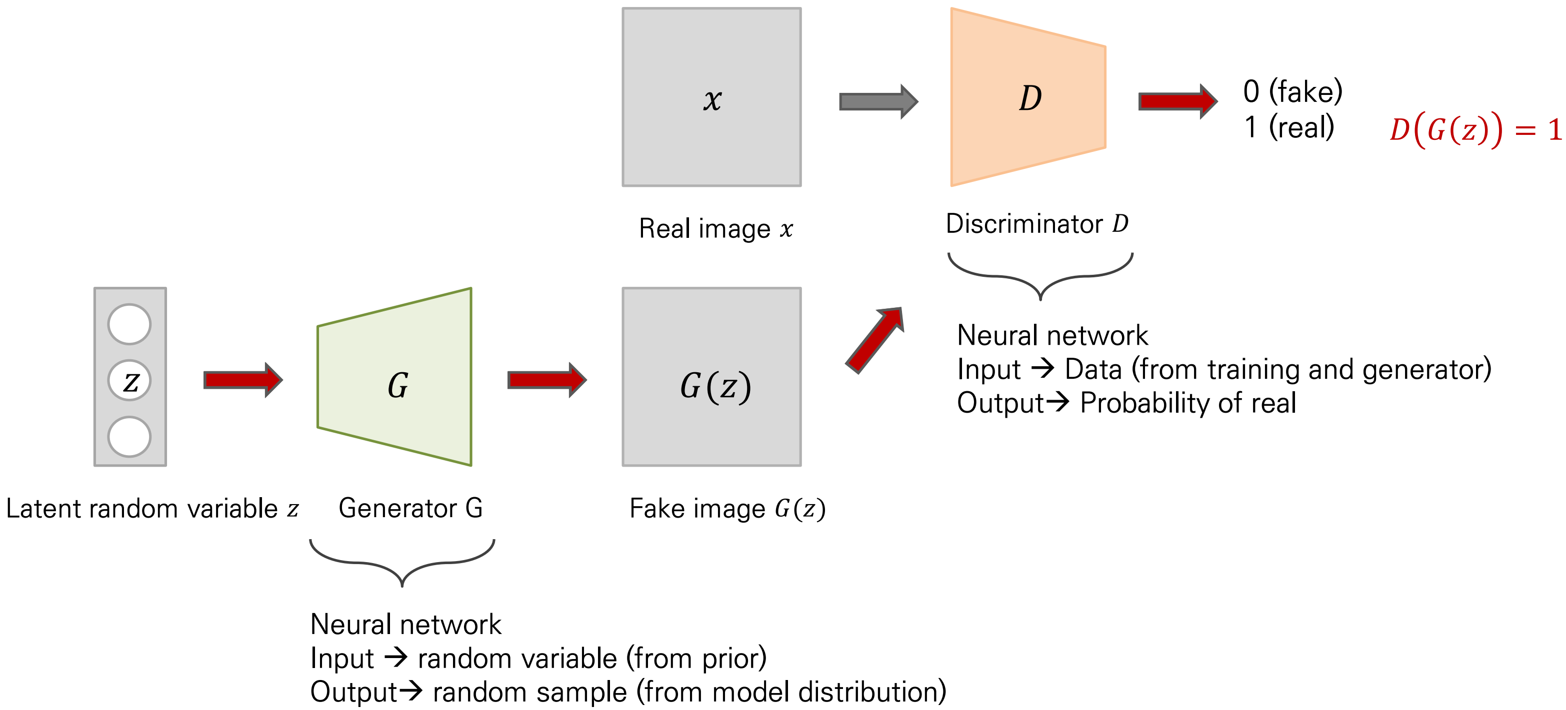
❖ GAN objective function (generator)

G is independent of this part

$$\min_{G}\max_{D} V(D,G) = \cancel{E_{x\sim p_{data}(x)}[\log D(x)]} + E_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

G should minimize $V(D,G)$

Minimum when $D(G(z) = 1$



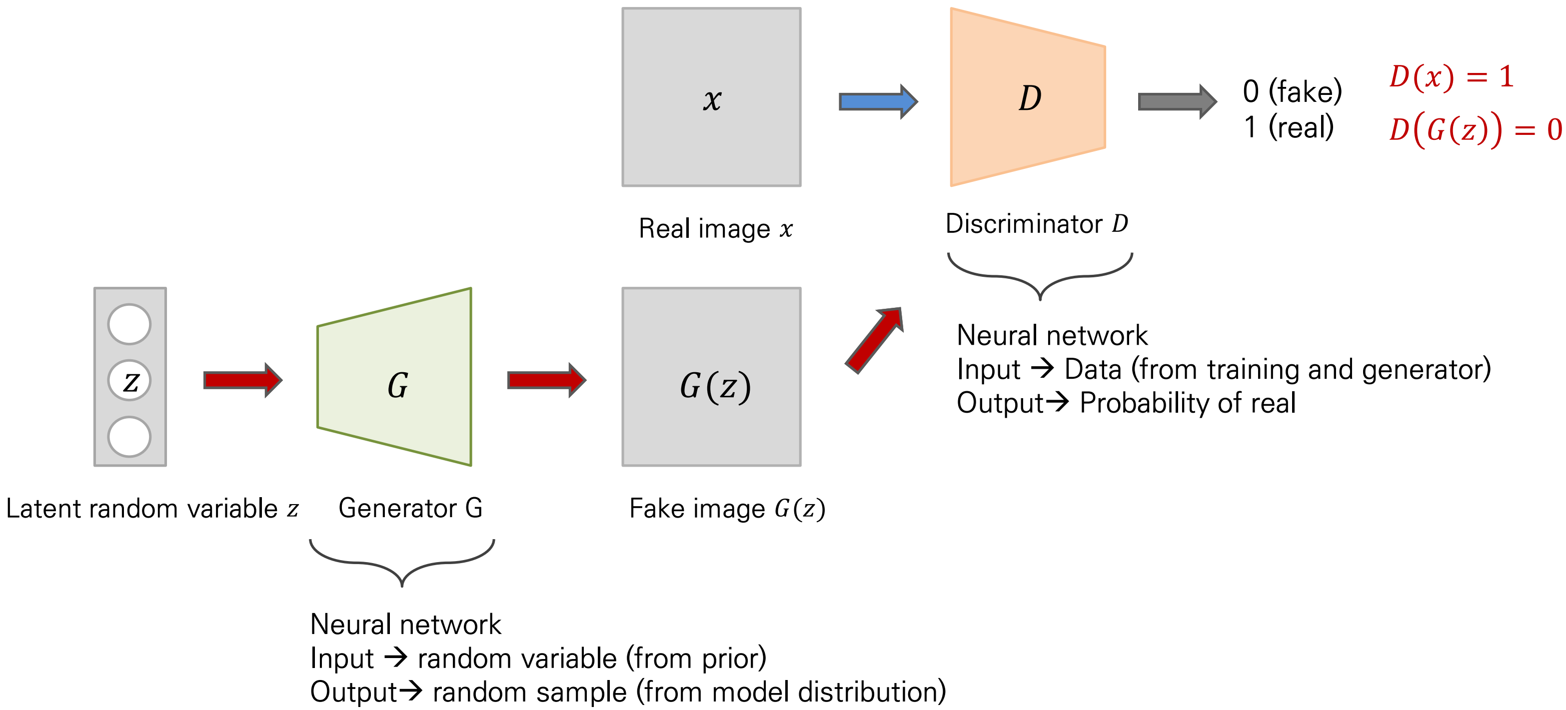
Literature Reviews

– Generative Adversarial Nets

❖ GAN objective function (discriminator)

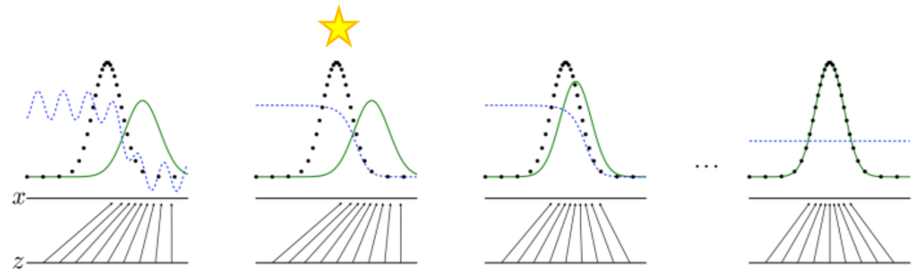
$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,[\log\left(1 - D\left(G(z)\right)\right)]$$

D should maximize $V(D,G)$ Maximum when $D(x) = 1$ Maximum when $D(G(z) = 0$



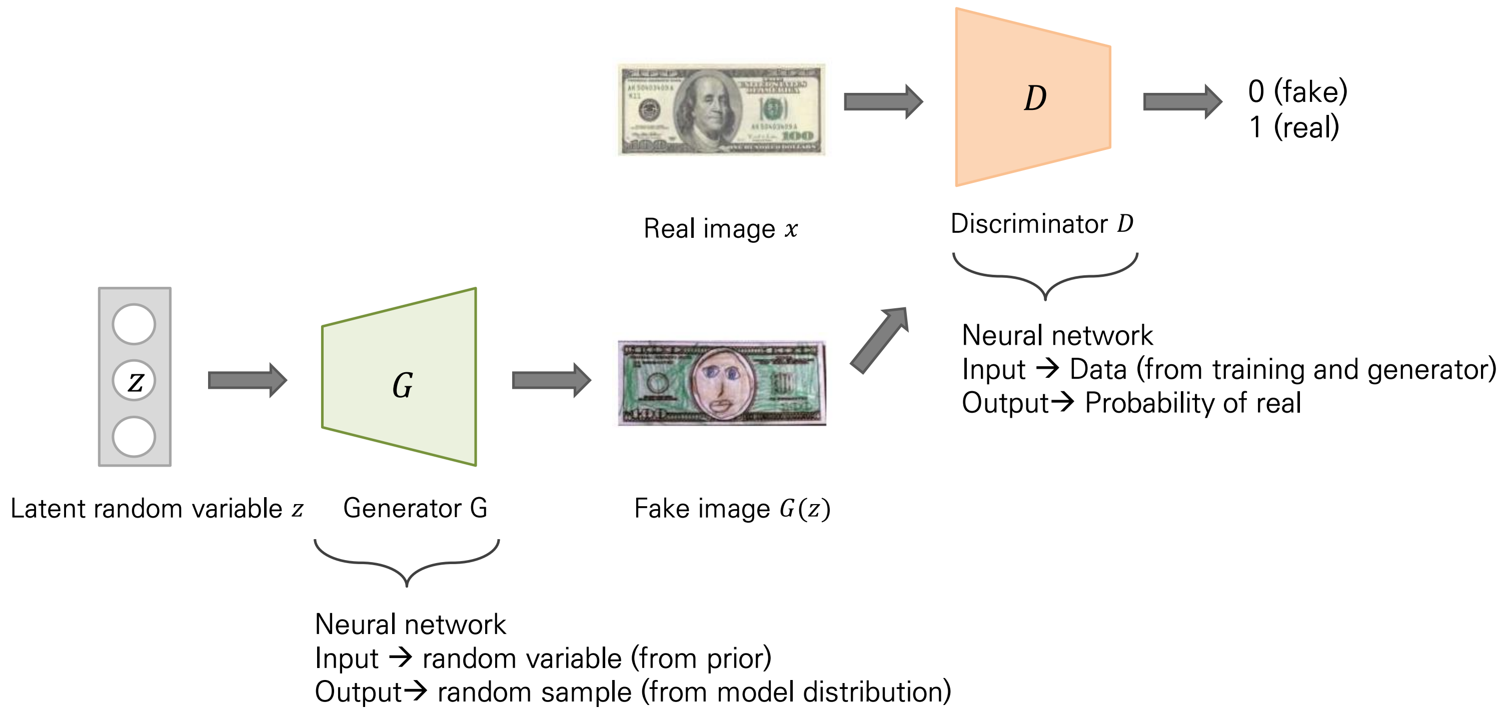
Literature Reviews

– Generative Adversarial Nets



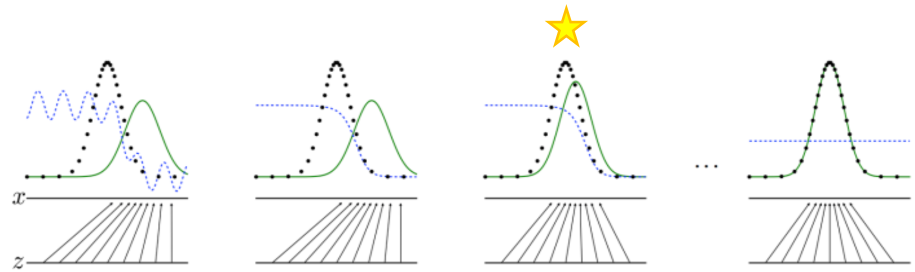
❖ GAN objective function

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$



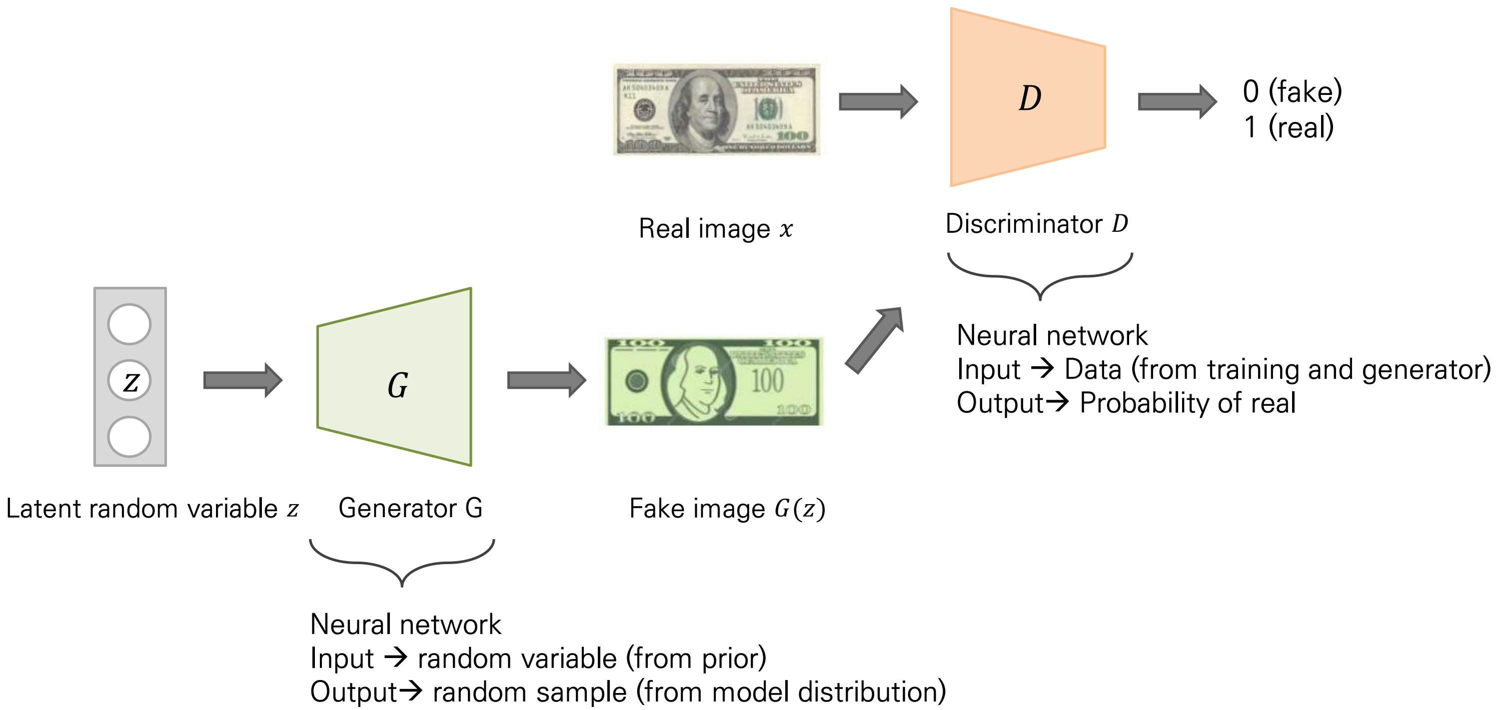
Literature Reviews

– Generative Adversarial Nets



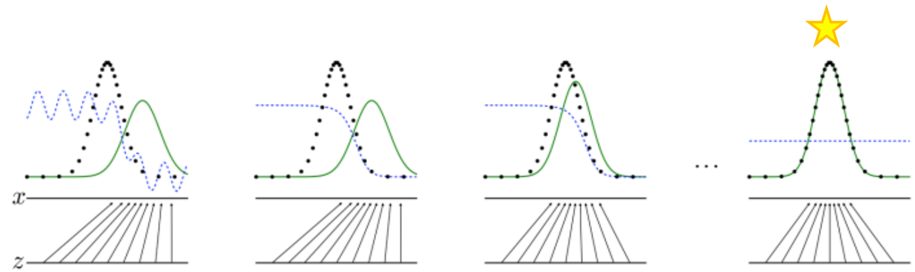
❖ GAN objective function

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[logD(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$



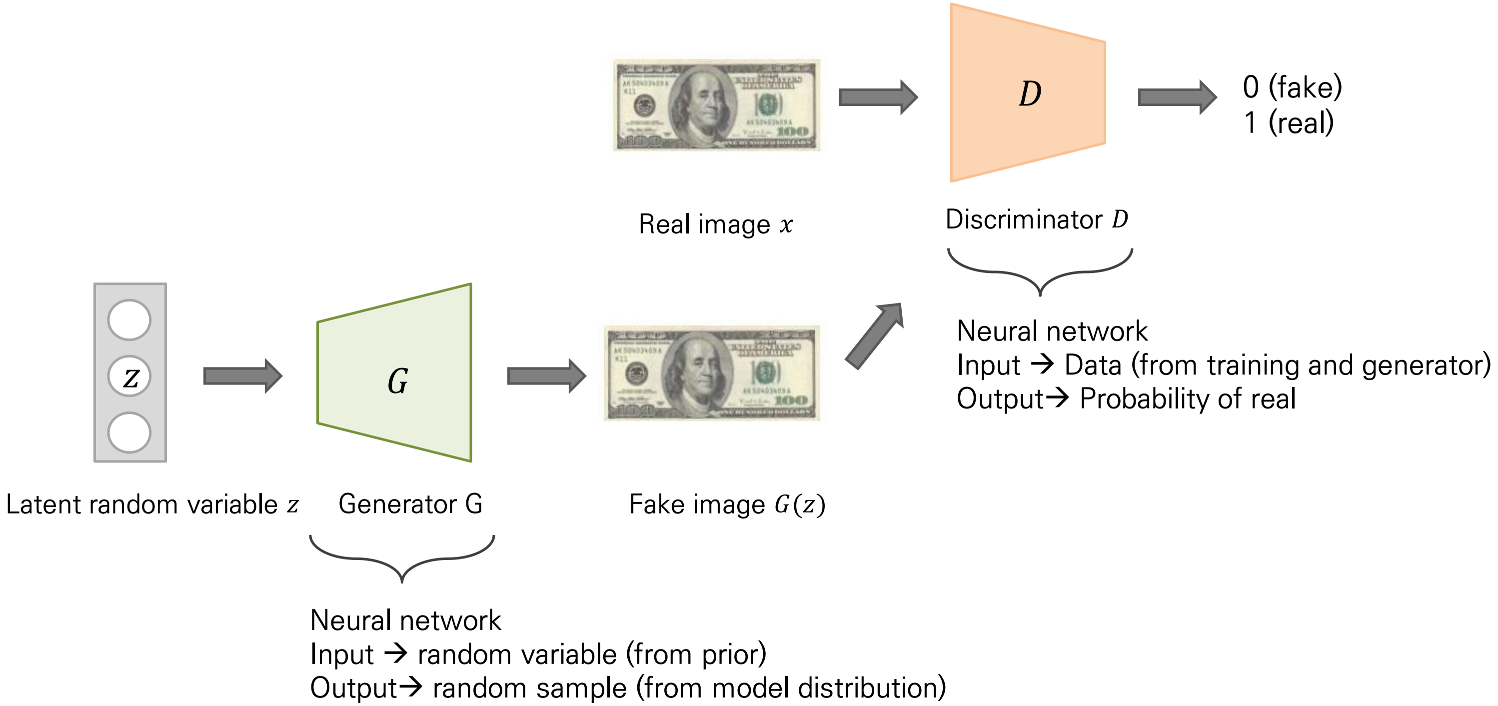
Literature Reviews

– Generative Adversarial Nets



❖ GAN objective function

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}\left[logD(x)\right] + E_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$



Literature Reviews

– Generative Adversarial Nets for Imputation

❖ GAN objective function

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

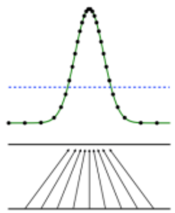
$$\begin{aligned} V(D,G) &= E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,\left[\log\left(1 - D\left(G(z)\right)\right)\right] \\ &= E_{x\sim p_{data}(x)}[logD(x)] + E_{x\sim P_g(x)}[\log(1 - D(x))] \\ &= \int_x P_{data}(x)\log\left(D(x)\right) + p_g(x)\log(1 - D(x))dx \end{aligned}$$

$for\ G\ fixed, the\ optimal\ discriminator\ D\ is\ maximized\ when$

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Literature Reviews

– Generative Adversarial Nets for Imputation



❖ GAN objective function

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,[\log\left(1 - D\left(G(z)\right)\right)]$$

$$\begin{aligned}
C(G) &= V(D^*,G)\\
&= E_{x\sim p_{data}(x)}[logD^*(x)] + E_{x\sim P_g(x)}[\log(1 - D^*(x))]\\
&= E_{x\sim p_{data}(x)}\left[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim P_g(x)}\left[\log(1 - \frac{p_{data}(x)}{p_{data}(x)+p_g(x)})\right]\\
&= E_{x\sim p_{data}(x)}\left[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim P_g(x)}\left[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right]\\
&= -\log(4) + KL(p_{data} \parallel \frac{p_{data}(x)+p_g(x)}{2}) + KL(p_g \parallel \frac{p_{data}(x)+p_g(x)}{2})\\
&= -\log(4) + 2 \times JSD(p_{data} \parallel p_g)
\end{aligned}$$

The global minimum of the virtual training criterion $C(G)$ *is achieved if and ony if*

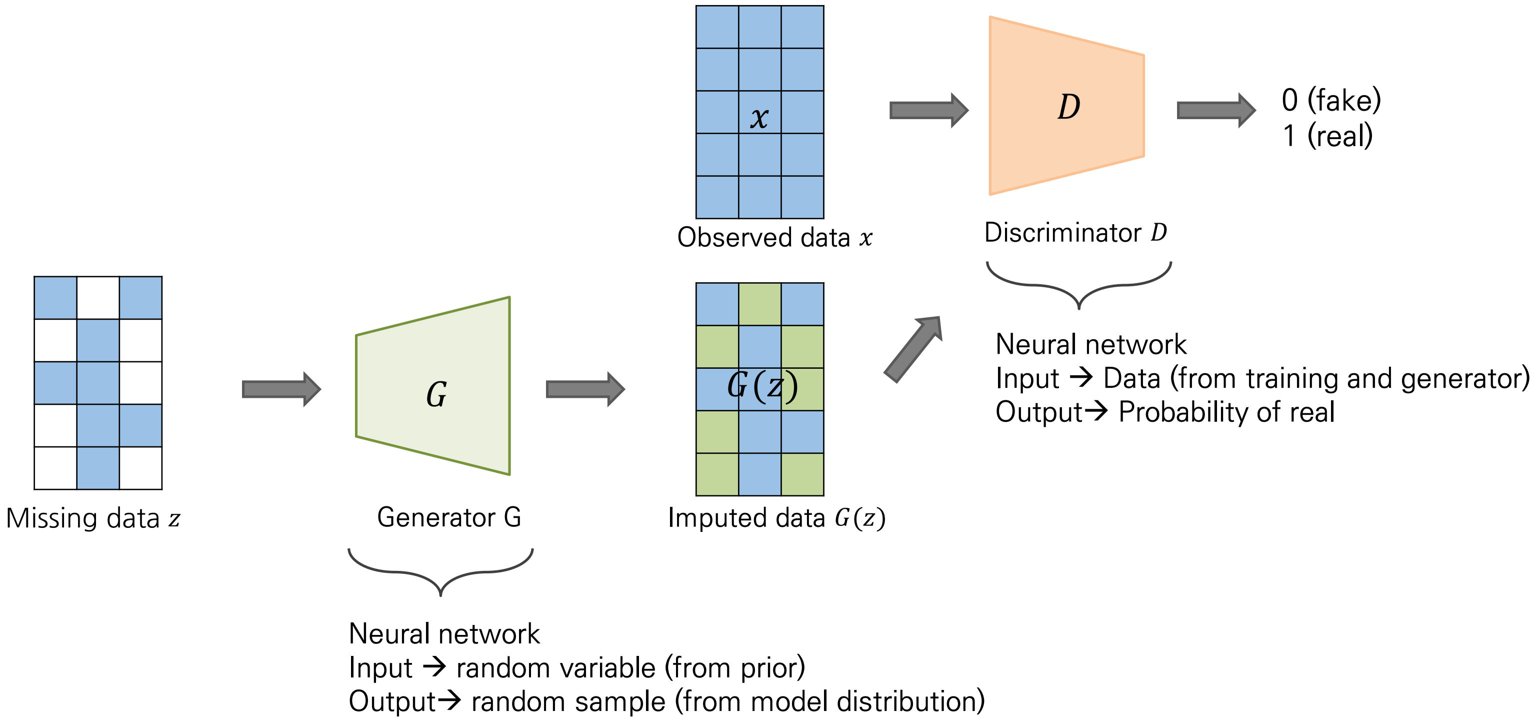
$p_g = p_{data}$. *at that point,* $C(G)$ *achieves the value* $-\log(4)$

Literature Reviews

– Generative Adversarial Nets for Imputation

❖ GAN objective function

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\,[logD(x)] + E_{z\sim p_z(z)}\,\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$



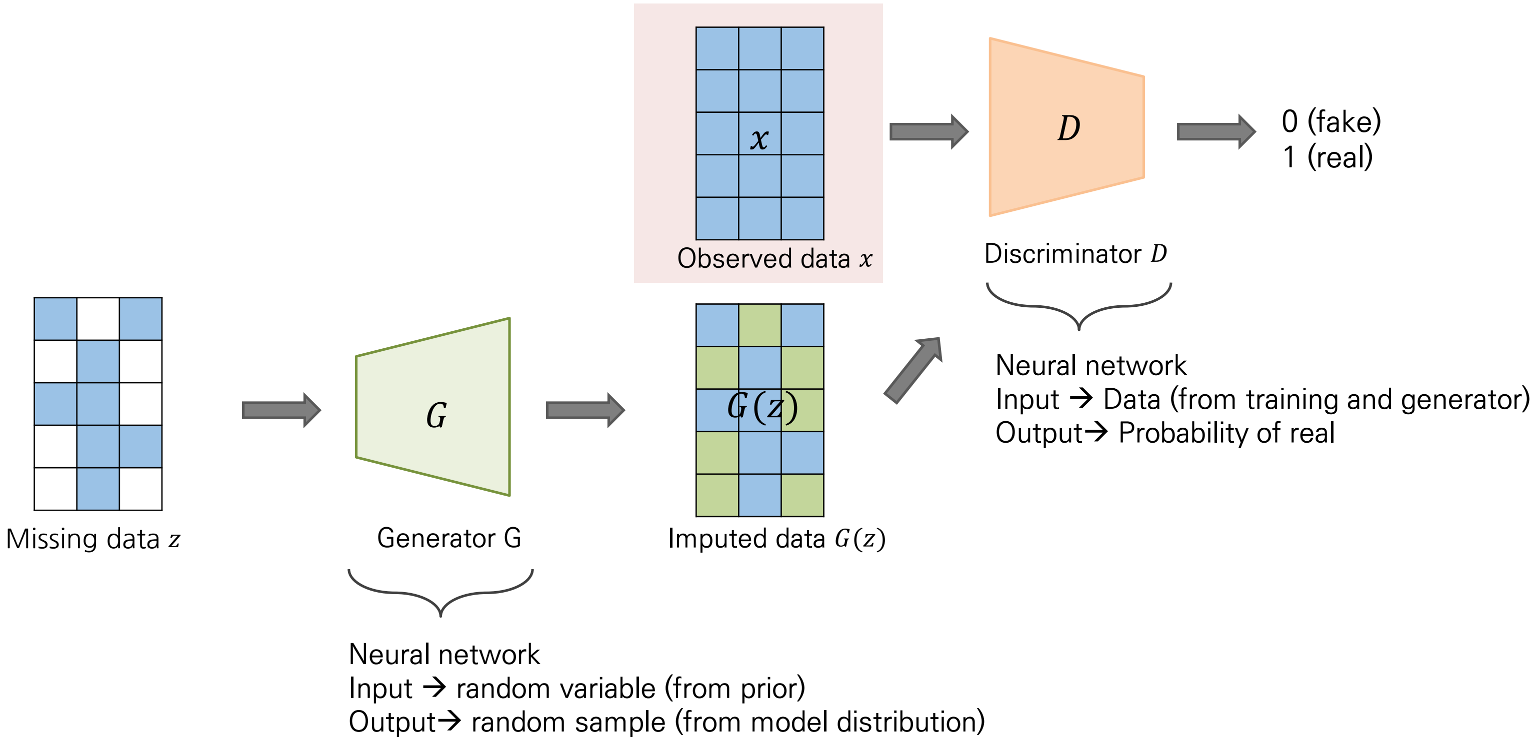
Literature Reviews

– Generative Adversarial Denoising Autoencoder for Face Completion

❖ GAN을 활용한 이미지 복구

➢ Generative model 기반, 대표적 결측치 대체 연구

➢ 한계점: 대다수 결측치 대체 모델은 학습시키는 과정에서 완전 데이터 필요

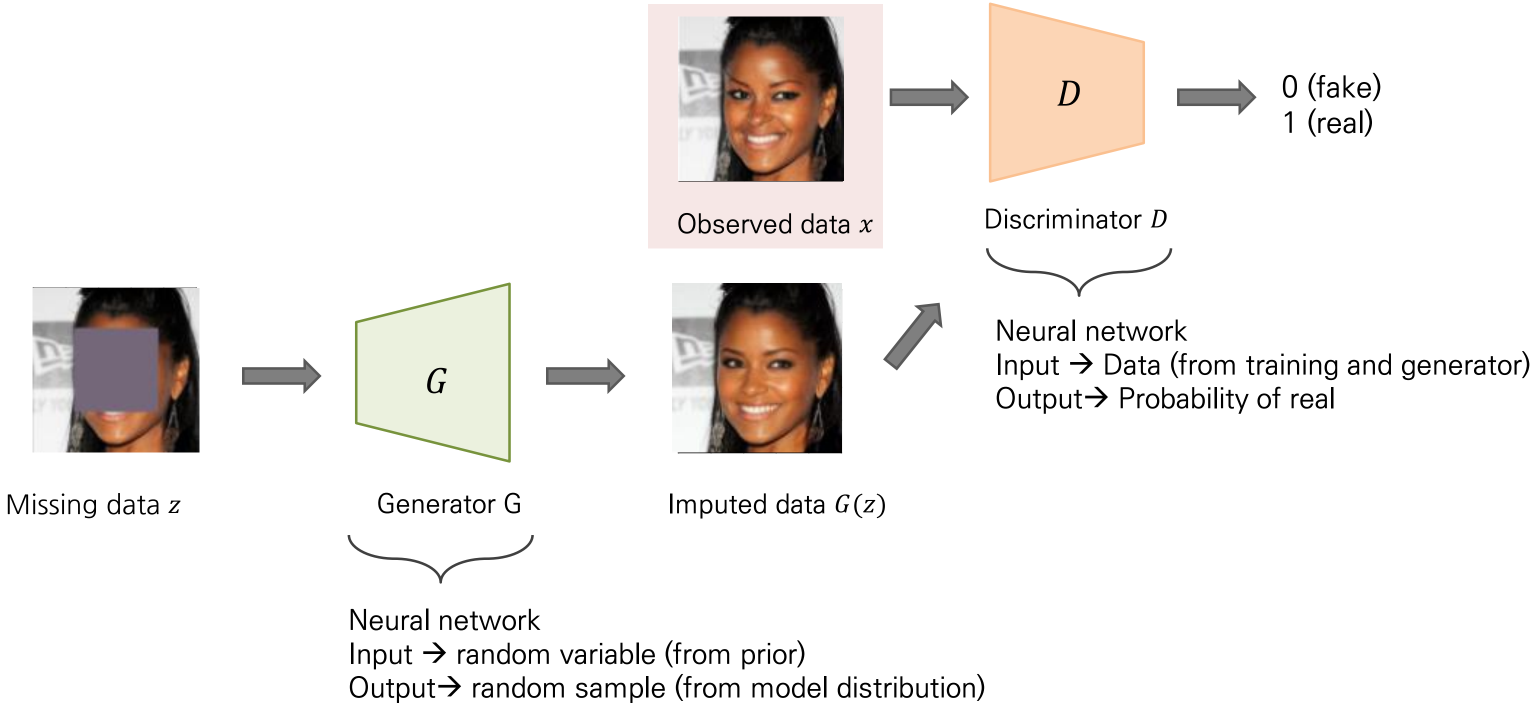


Literature Reviews

– Generative Adversarial Denoising Autoencoder for Face Completion

❖ GAN을 활용한 이미지 복구

- Generative model 기반, 대표적 결측치 대체 연구
- 한계점: 대다수 결측치 대체 모델은 학습시키는 과정에서 완전 데이터 필요



Literature Reviews

– Generative Adversarial Denoising Autoencoder for Face Completion

❖ GAN을 활용한 이미지 복구

- ➢ 계측 데이터에 대한 정보 다량 유실
- ➢ 실험 데이터의 결측치 패턴이 학습 데이터의 결측치 패턴과 유사하다는 가정 필요

STEP 1.

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
이지윤	F	26	48	166	1	1

〈완전 데이터〉

STEP 2.

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
이지윤	F	26	48	166	1	1

〈완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	NaN	183	1	1
이지윤	NaN	26	48	166	1	1

〈불완전 데이터〉

Train set

Test set

STEP 3. 결측치 대체

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	NaN	0	NaN
이지윤	F	26	48	166	1	1
민다빈	M	NaN	70	180	1	0
이민정	F	27	46	162	NaN	0

〈불완전 데이터〉

이름	성별	나이	무게	키	흡연 여부	당뇨병 여부
강현규	M	27	80	183	1	1
강현구	F	26	65	173	0	0
이지윤	F	26	48	166	1	1
민다빈	M	28	70	180	1	0
이민정	F	27	46	162	0	0

〈완전 데이터〉

https://www.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen/

Generative Adversarial Imputation Nets

ICML | 2018

Thirty-fifth International Conference on Machine Learning

Dates | Schedule | Calls | Stockholm | Students | Committees

- Year (2018)
- Help
- My Registrations
- Profile
- Sponsor Info
- Code of Conduct
- Future Meetings
- Board 2019

Program Highlights »

Thu Jul 12th 06:15 -- 09:00 PM @ Hall B #83 Poster

GAIN: Missing Data Imputation using Generative Adversarial Nets

Jinsung Yoon · James Jordon · Mihaela van der Schaar In Posters Thu

PDF »

Jinsung Yoon » James Jordon » Mihaela van der Schaar »

We propose a novel method for imputing missing data by adapting the well-known Generative Adversarial Nets (GAN) framework. Accordingly, we call our method Generative Adversarial Imputation Nets (GAIN). The generator (G) observes some components of a real data vector, imputes the missing components conditioned on what is actually observed, and outputs a completed vector. The discriminator (D) then takes a completed vector and attempts to determine which components were actually observed and which were imputed. To ensure that D forces G to learn the desired distribution, we provide D with some additional information in the form of a hint vector. The hint reveals to D partial information about the missingness of the original sample, which is used by D to focus its attention on the imputation quality of particular components. This hint ensures that G does in fact learn to generate according to the true data distribution. We tested our method on various datasets and found that GAIN significantly outperforms state-of-the-art imputation methods.

jsyoon0823 Update README.md		Latest commit d4110a7 on 23 Dec 2018
MNST_Code_Example.py	Update MNST_Code_Example.py	4 months ago
README.md	Update README.md	2 months ago

README.md

Generative Adversarial Imputation Networks (GAIN)

Title: GAIN: Missing Data Imputation using Generative Adversarial Nets

Authors: Jinsung Yoon, James Jordon, Mihaela van der Schaar

Reference: J. Yoon, J. Jordon, M. van der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets," International Conference on Machine Learning (ICML), 2018.

Paper Link: http://medianetlab.ee.ucla.edu/papers/ICML_GAIN.pdf

Appendix Link: http://medianetlab.ee.ucla.edu/papers/ICML_GAIN_Supp.pdf

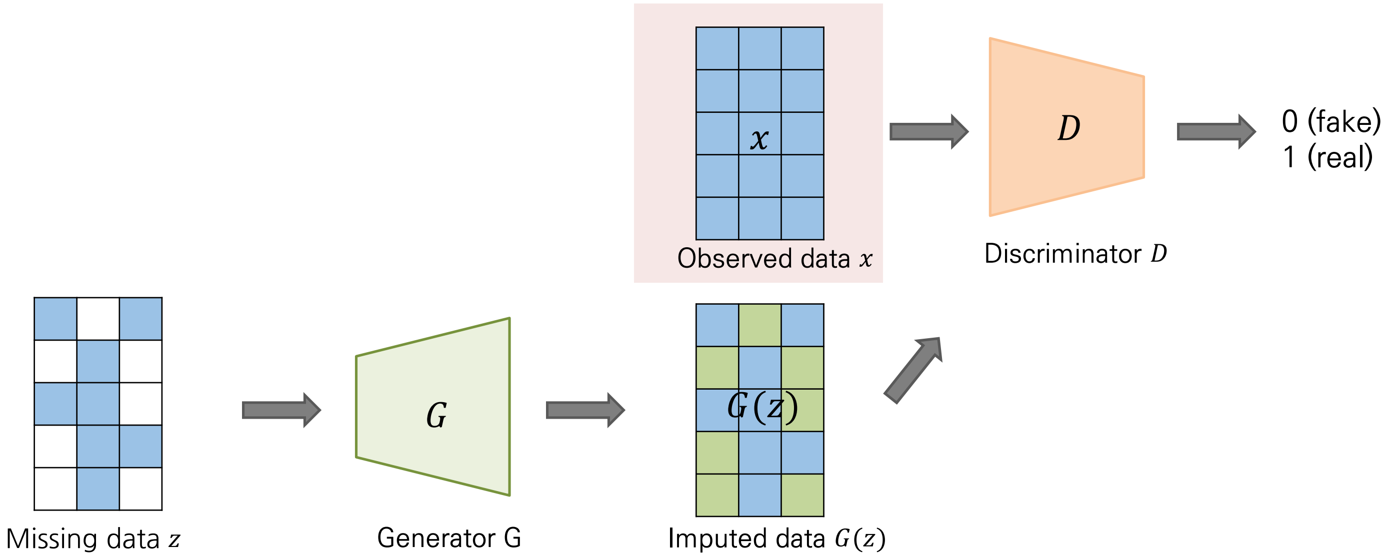
GAIN

– Prerequisite

❖ Standard GAN 기반 결측치 대체 기법

➢ 기존 방법론의 한계점 개선: 완전 데이터 불필요

➢ 결측치는 MCAR(missing completely at random) 가정



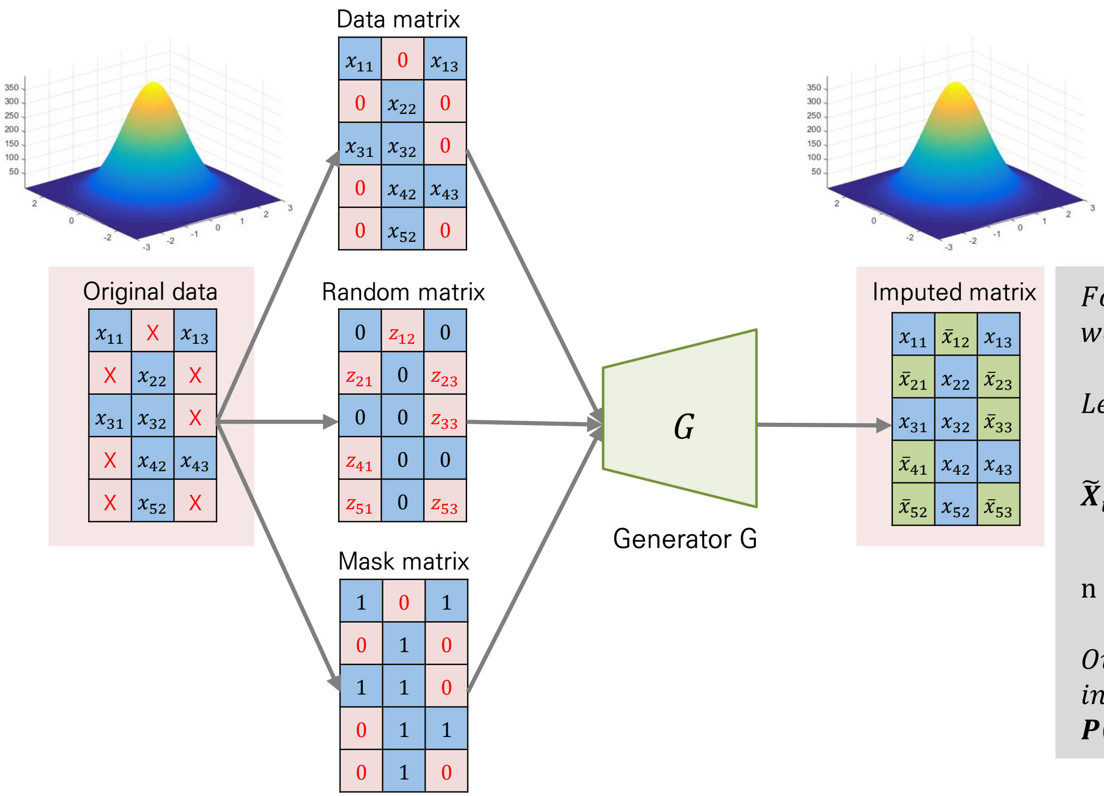
- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Problem Formulation

❖ Generative model의 특징을 접목

➢ 단일 기댓값이 아닌 계측된 데이터의 분포를 모델링하여, 대체 값의 불확실성을 보완하며, multiple imputation이 가능해짐



For each $i \in \{1, \ldots, d\}$,
we define a space $\tilde{\chi}_i = \chi_i \cup \{*\}$

Let $\tilde{\boldsymbol{X}} = (\tilde{X}_1, \cdots, \tilde{X}_d) \in \tilde{\chi}$

$$\tilde{\boldsymbol{X}}_i = \begin{cases} \boldsymbol{X}_i, \boldsymbol{if}\ \boldsymbol{M} = \boldsymbol{1} \\ *, \boldsymbol{otherwise} \end{cases}$$

n *i.i.d. copies of* $\tilde{X}$ *are realized, denoted* $\tilde{x}^1, \ldots, \tilde{x}^n$.

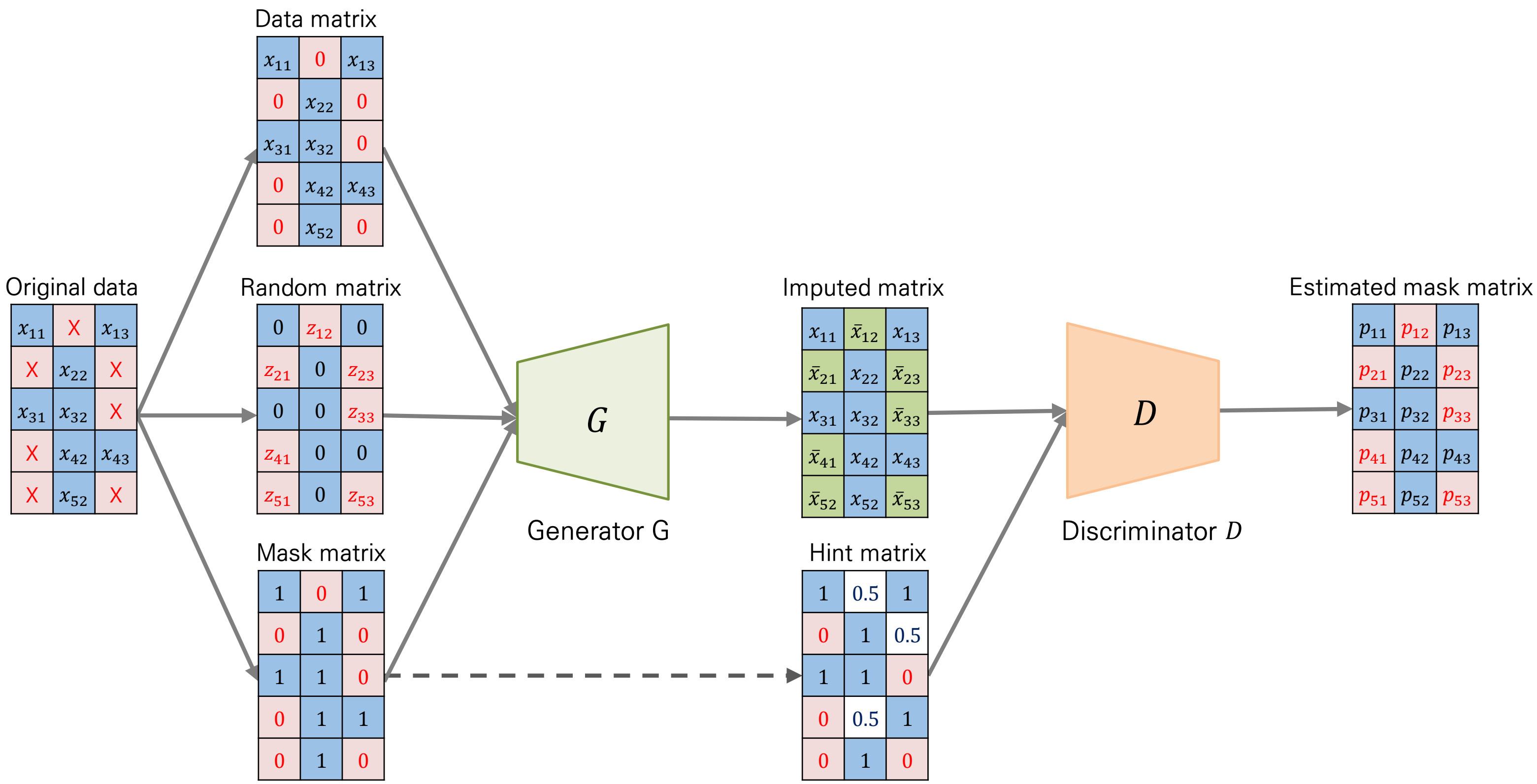
Our goal is to impute the unobserved values $\tilde{x}_i$
in each We want to generate samples according to
$\boldsymbol{P}(\boldsymbol{X}|\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}}^i)$

GAIN

– Problem Formulation

❖ GAIN은 두개의 네트워크로 구성

- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크



GAIN

– Problem Formulation

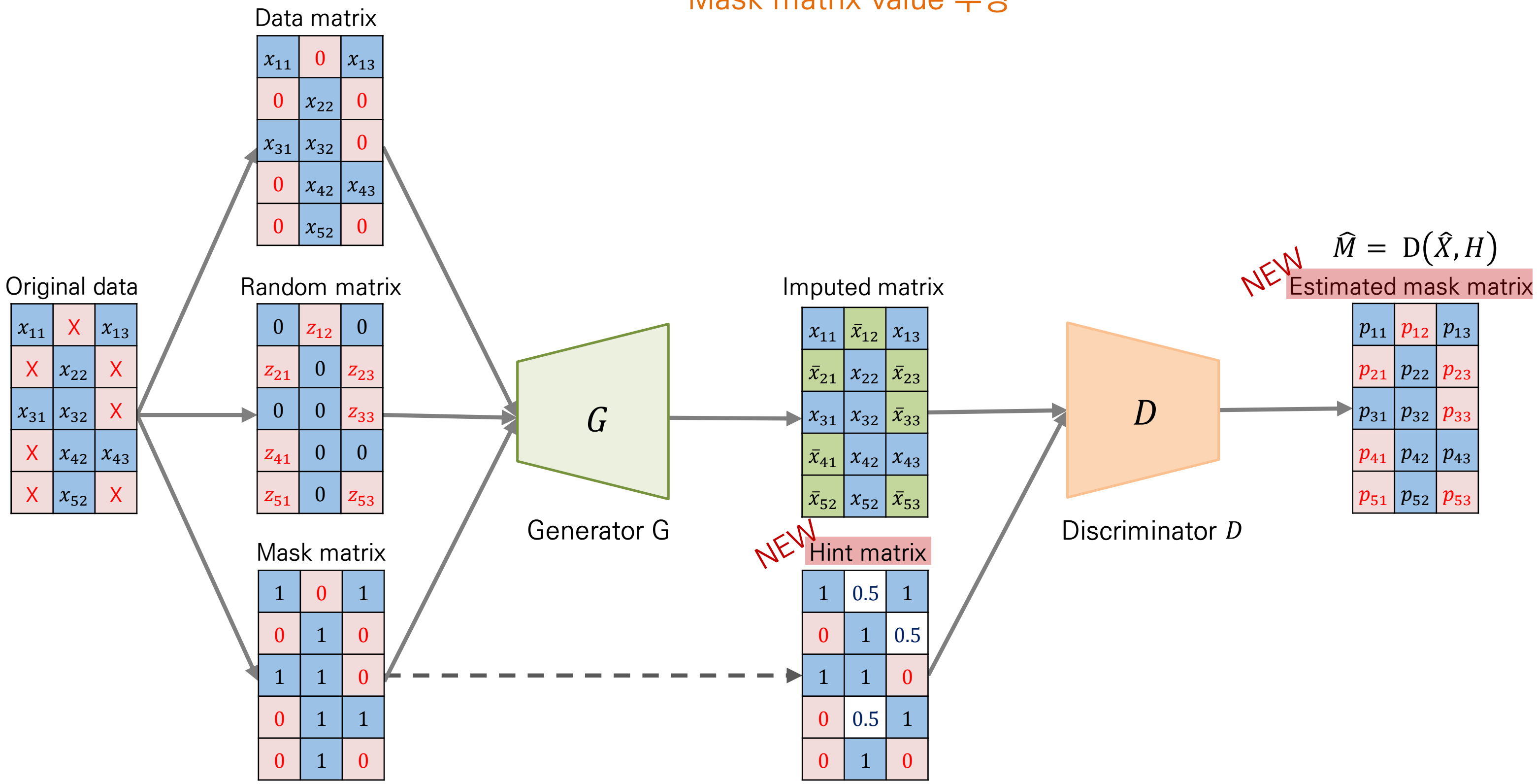
❖ GAIN은 두개의 네트워크로 구성

결측치 대체

➢ Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크

➢ Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

Mask matrix value 추정



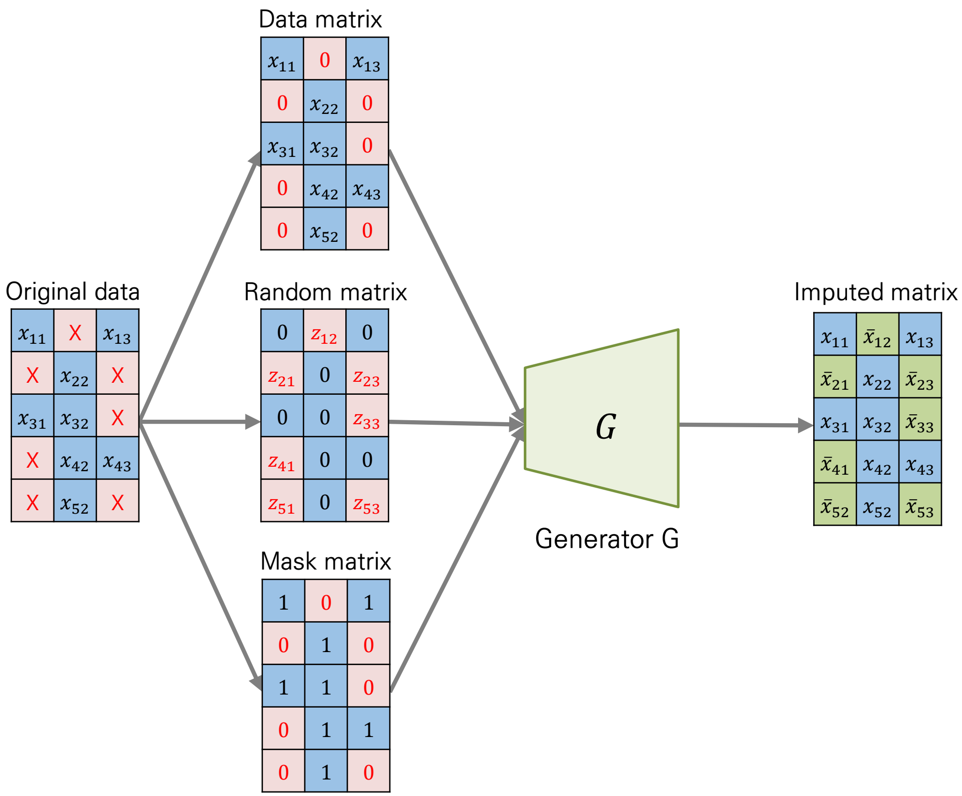
- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Generator G

❖ Input: $\tilde{X}(data\ matrix),\ Z(random\ matrix),\ M(mask\ matrix)$

❖ Output: $\hat{X}(imputed\ matrix)$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

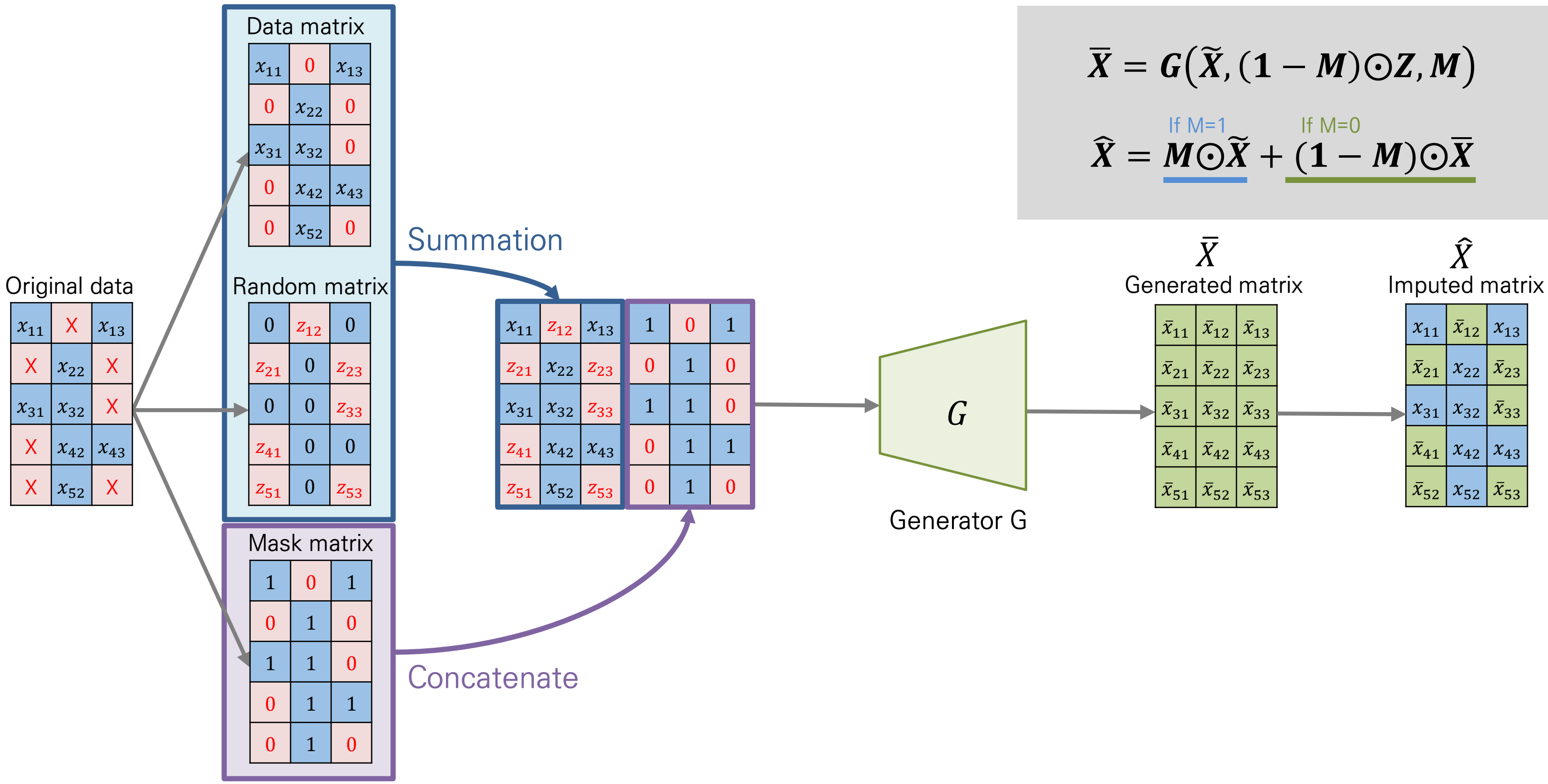
– Generator G

❖ Input: $\tilde{X}(data\ matrix), (1-M)\odot Z(random\ matrix),\ M(mask\ matrix)$

❖ Output: $\hat{X}(imputed\ matrix)$

$$\bar{\boldsymbol{X}} = \boldsymbol{G}(\tilde{\boldsymbol{X}}, (\boldsymbol{1}-\boldsymbol{M})\odot\boldsymbol{Z}, \boldsymbol{M})$$

$$\hat{\boldsymbol{X}} = \underbrace{\boldsymbol{M}\odot\tilde{\boldsymbol{X}}}_{\text{If M=1}} + \underbrace{(\boldsymbol{1}-\boldsymbol{M})\odot\bar{\boldsymbol{X}}}_{\text{If M=0}}$$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

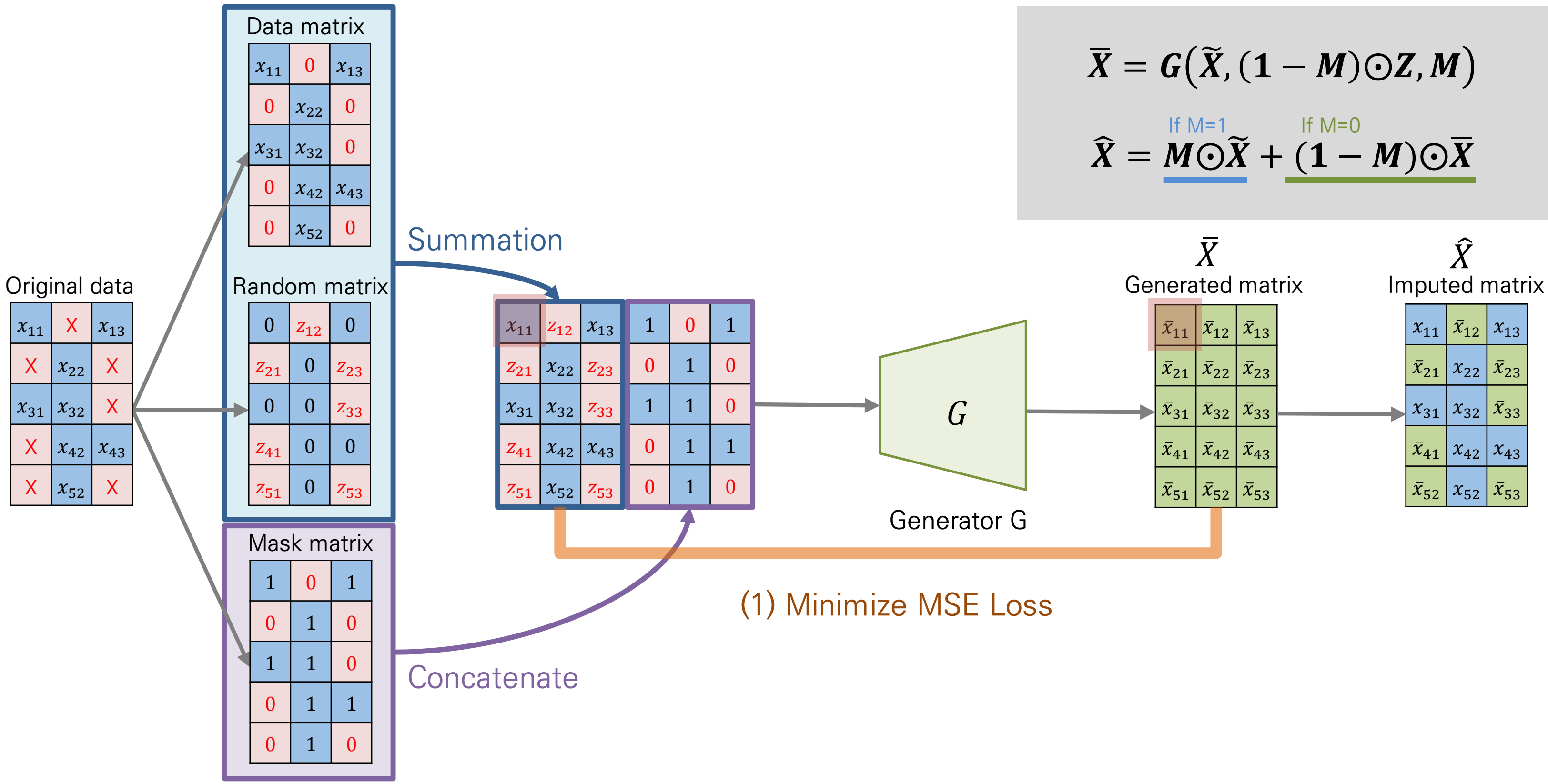
– Generator G

❖ 학습 목표

- ➢ M=1(계측된 값)에 위치한 값이 실제 계측된 값과 비슷하게 생성
- ➢ Discriminator가 실제 계측된 값으로 판별

$$\bar{X} = G(\tilde{X}, (1-M)\odot Z, M)$$

$$\hat{X} = \underbrace{M\odot\tilde{X}}_{\text{If M=1}} + \underbrace{(1-M)\odot\bar{X}}_{\text{If M=0}}$$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

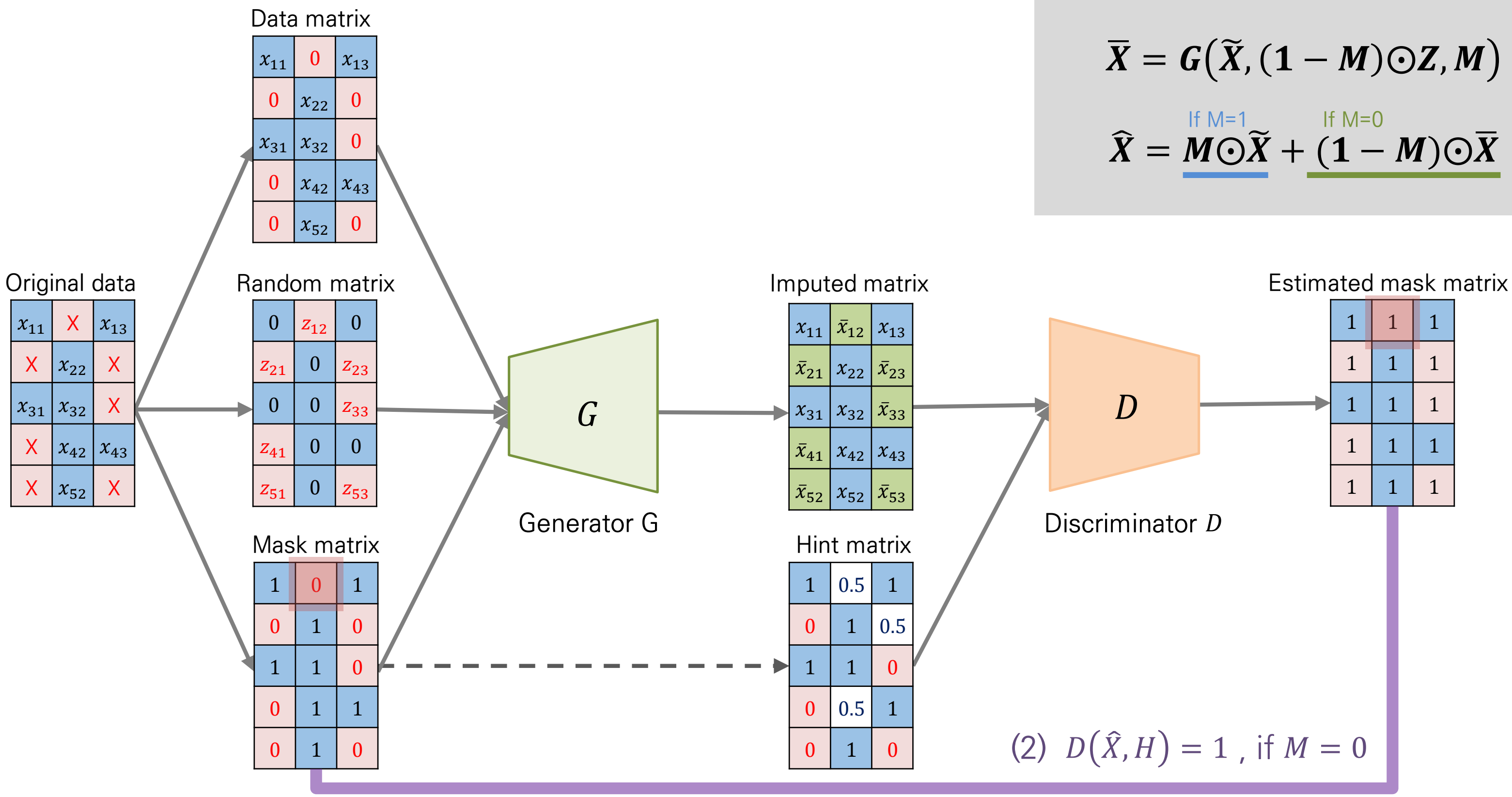
– Generator G

❖ 학습 목표

- ➢ M=1(계측된 값)에 위치한 값이 실제 계측된 값과 비슷하게 생성
- ➢ Discriminator가 실제 계측된 값으로 판별

$$\bar{X} = G(\tilde{X}, (1 - M) \odot Z, M)$$

$$\hat{X} = \underbrace{M \odot \tilde{X}}_{\text{If M=1}} + \underbrace{(1 - M) \odot \bar{X}}_{\text{If M=0}}$$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

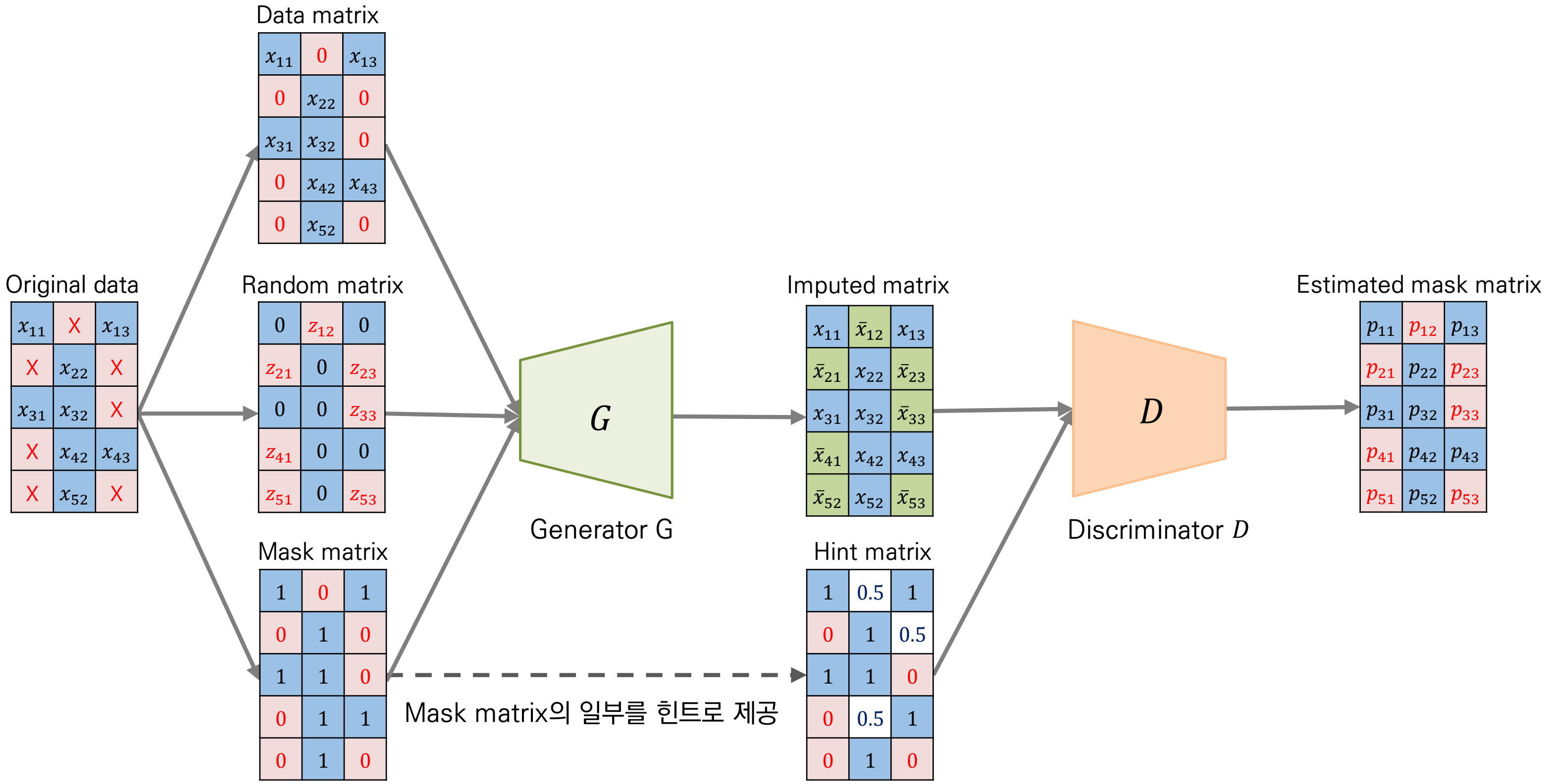
GAIN

– Discriminator D

❖ Input: $\hat{X}(imputed\ matrix)$, $H(hint\ matrix)$

❖ Output: $\widehat{M} = \mathrm{D}(\hat{X}, H)\ (estimated\ mask\ matrix)$

$\mathrm{D}(\hat{x}, h)$ 는 $\hat{X} = \hat{x}, H = h$일 때, 해당 값이 계측 값일 확률



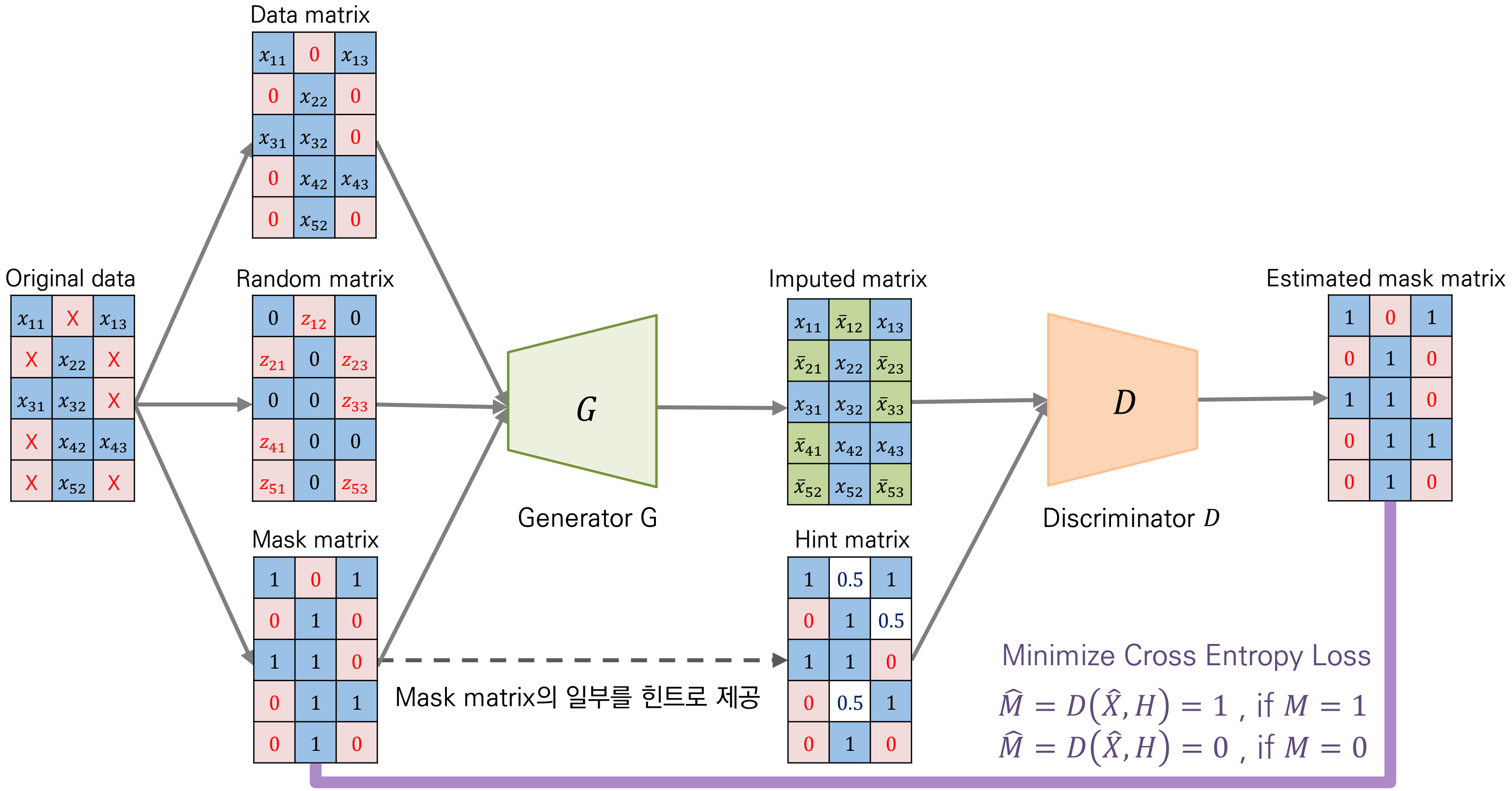
- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Discriminator D

❖ 학습 목표

- M=1(계측 값)에 위치한 값을 Discriminator가 계측 값으로 판별
- M=0(결측 값)에 위치한 값을 Discriminator가 결측 값으로 판별



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Hint H

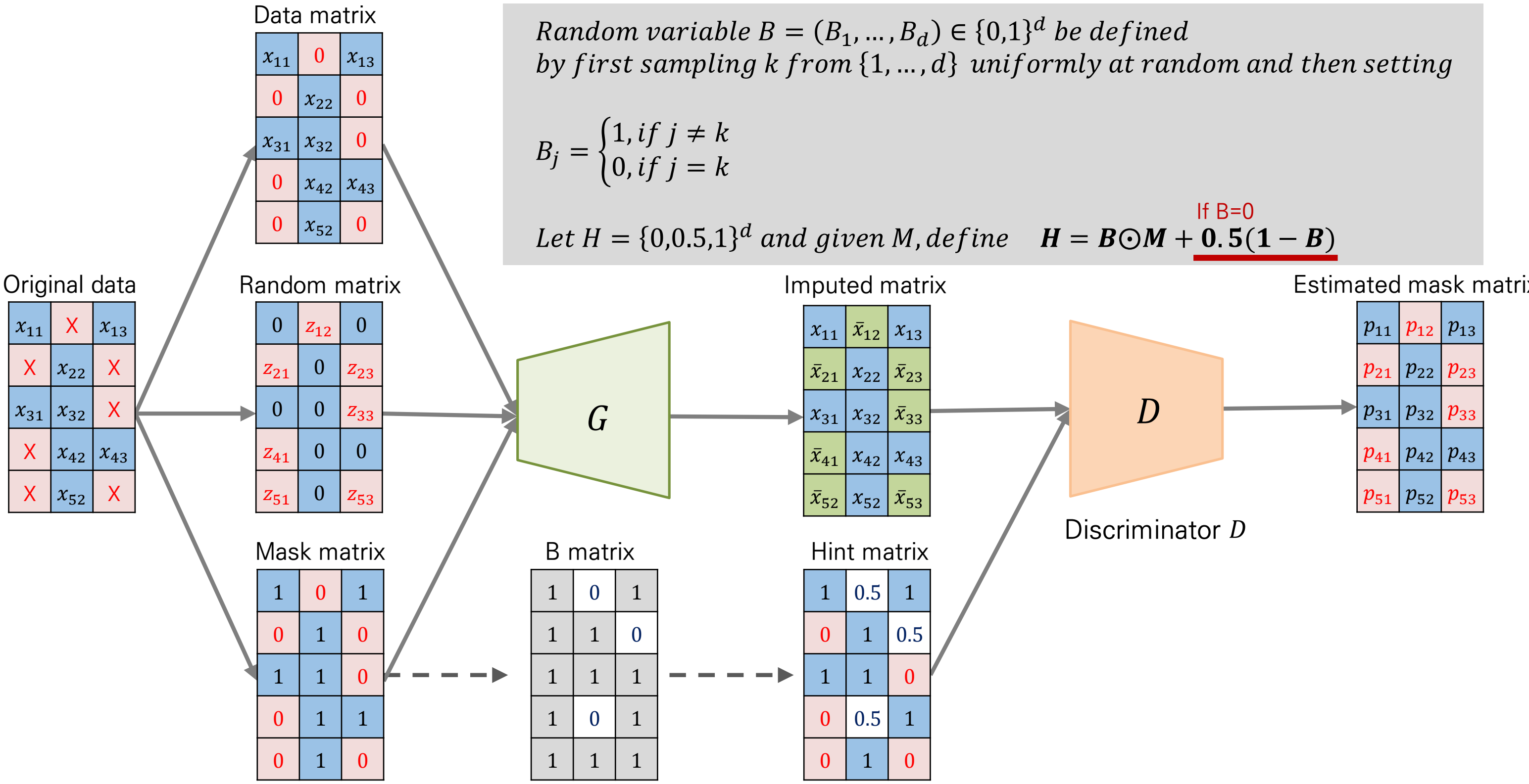
❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Random variable $B = (B_1, \dots, B_d) \in \{0,1\}^d$ *be defined by first sampling* k *from* $\{1, \dots, d\}$ *uniformly at random and then setting*

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

Let $H = \{0, 0.5, 1\}^d$ *and given* M*, define* $\boldsymbol{H} = \boldsymbol{B} \odot \boldsymbol{M} + \underbrace{\boldsymbol{0.5(1 - B)}}_{\text{If B=0}}$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

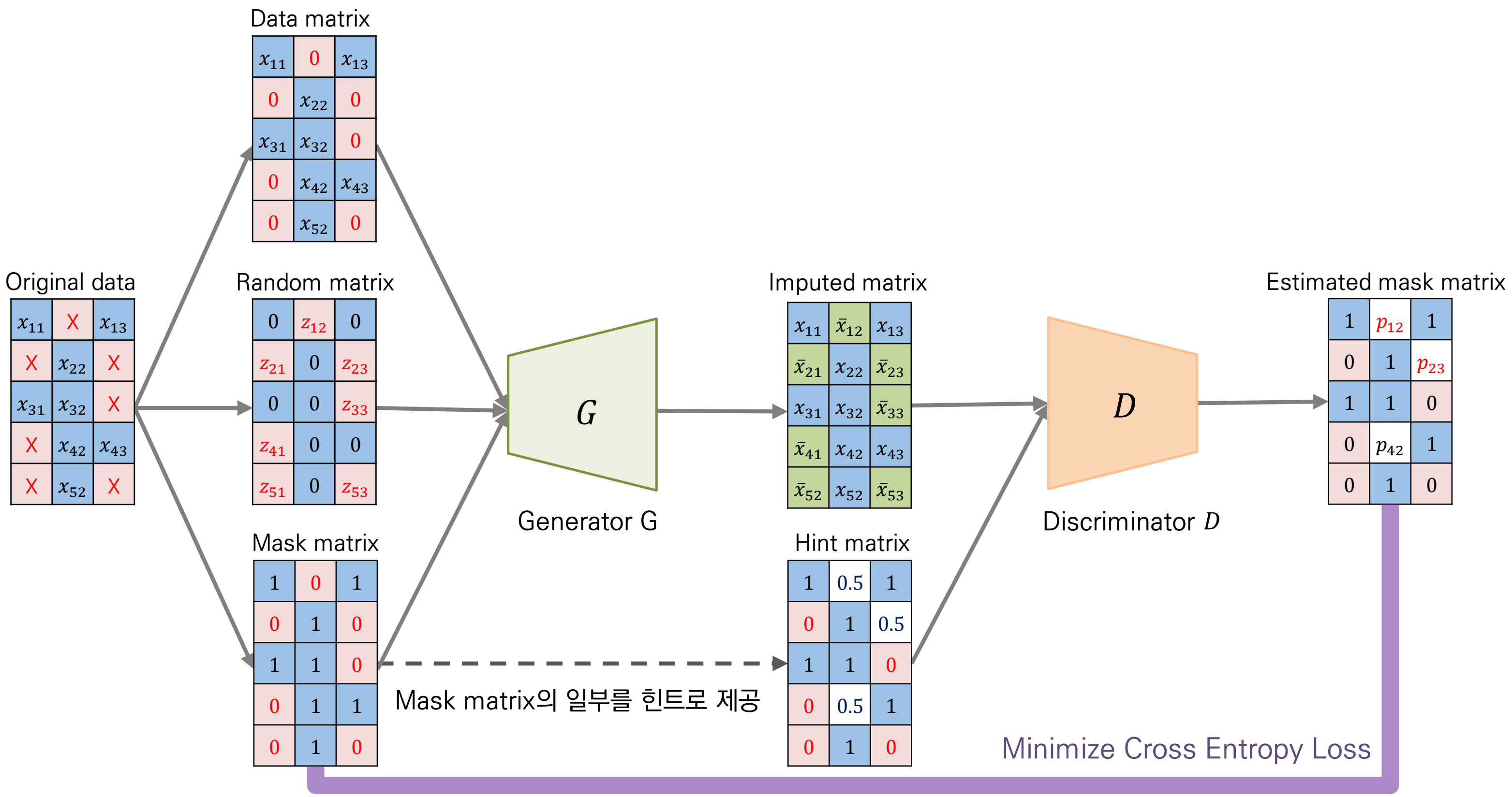
GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case1. 힌트 80%



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

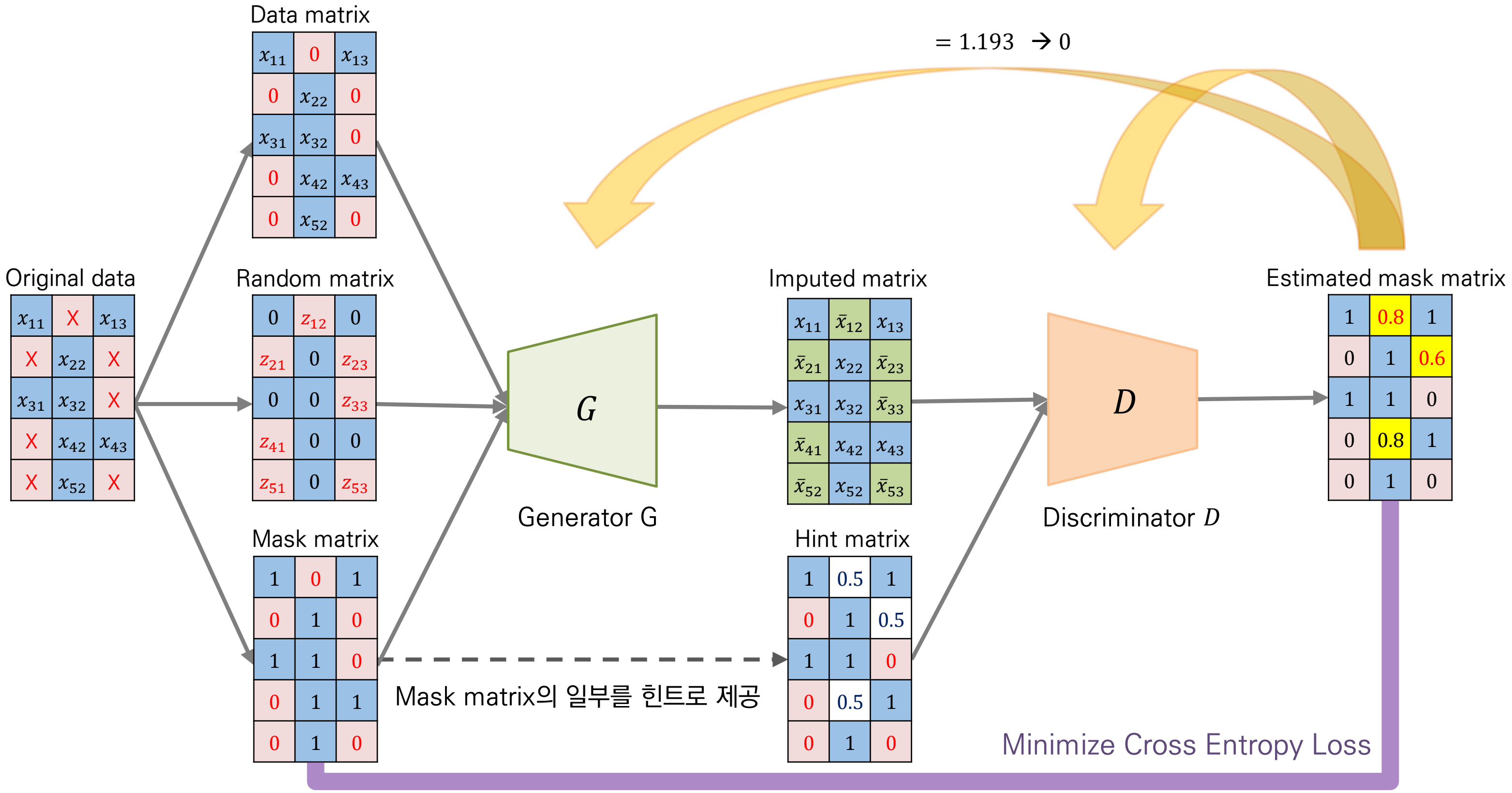
❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case1. 힌트 80%

$$-\left(\sum_{i:b_i=0} m_i \log(\widehat{m_i}) + (1-m_i)\log(1-\widehat{m_i})\right)$$

$$= -(1\log(0.8) + 1\log(0.2) + 1\log(0.4))$$

$$= 1.193 \rightarrow 0$$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

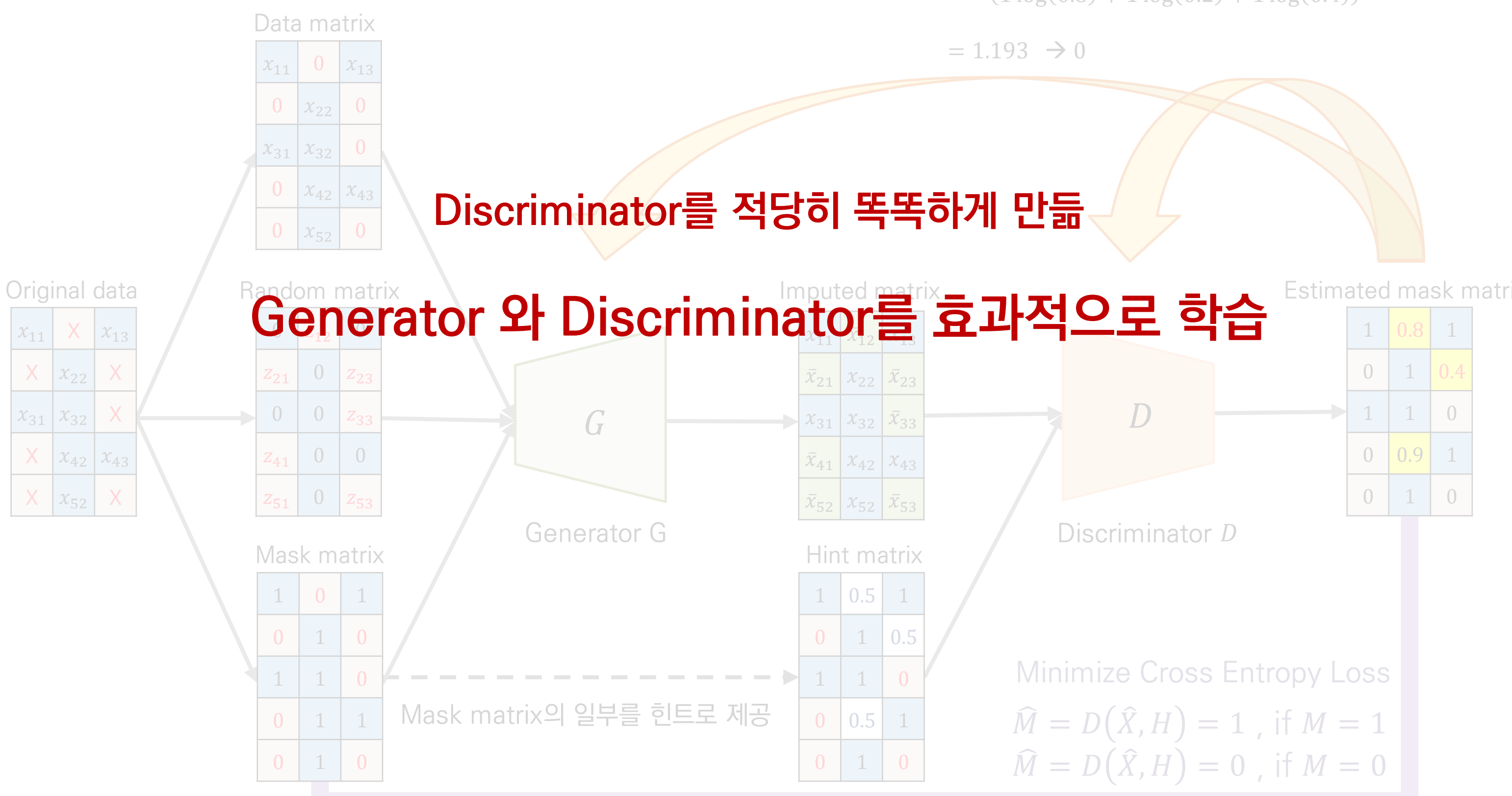
❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case1. 힌트 80%

$$-\left(\sum_{i:b_i=0} m_i \log(\widehat{m_i}) + (1-m_i)\log(1-\widehat{m_i})\right)$$
$$= -(1\log(0.8) + 1\log(0.2) + 1\log(0.4))$$
$$= 1.193 \rightarrow 0$$

Discriminator를 적당히 똑똑하게 만듦

Generator 와 Discriminator를 효과적으로 학습



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Hint H

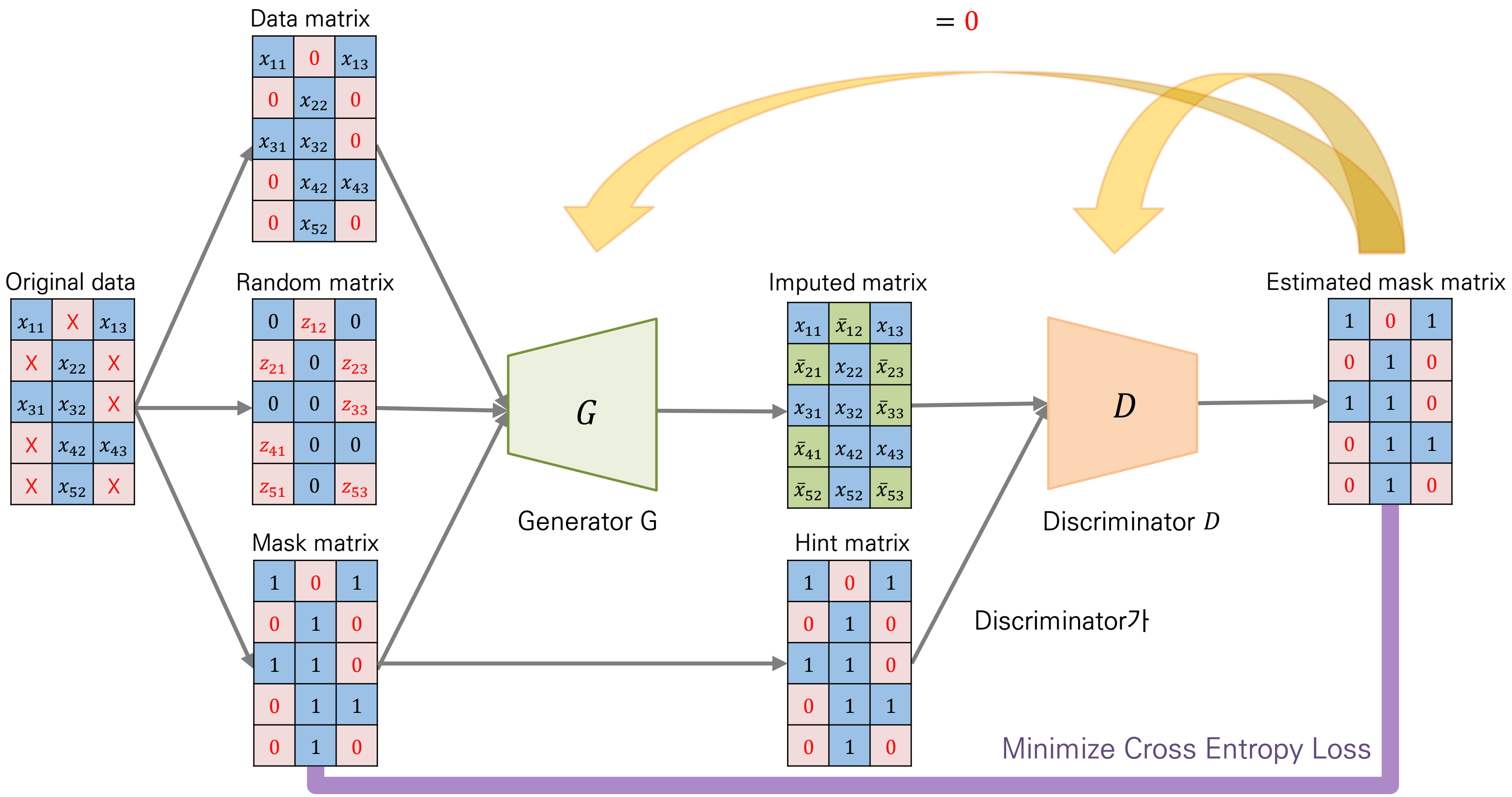
❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case2. 힌트 100%

$$-\left(\sum_{i:b_i=0} m_i \log(\widehat{m_i}) + (1-m_i)\log(1-\widehat{m_i})\right)$$

$$= 0$$



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

GAIN

– Hint H

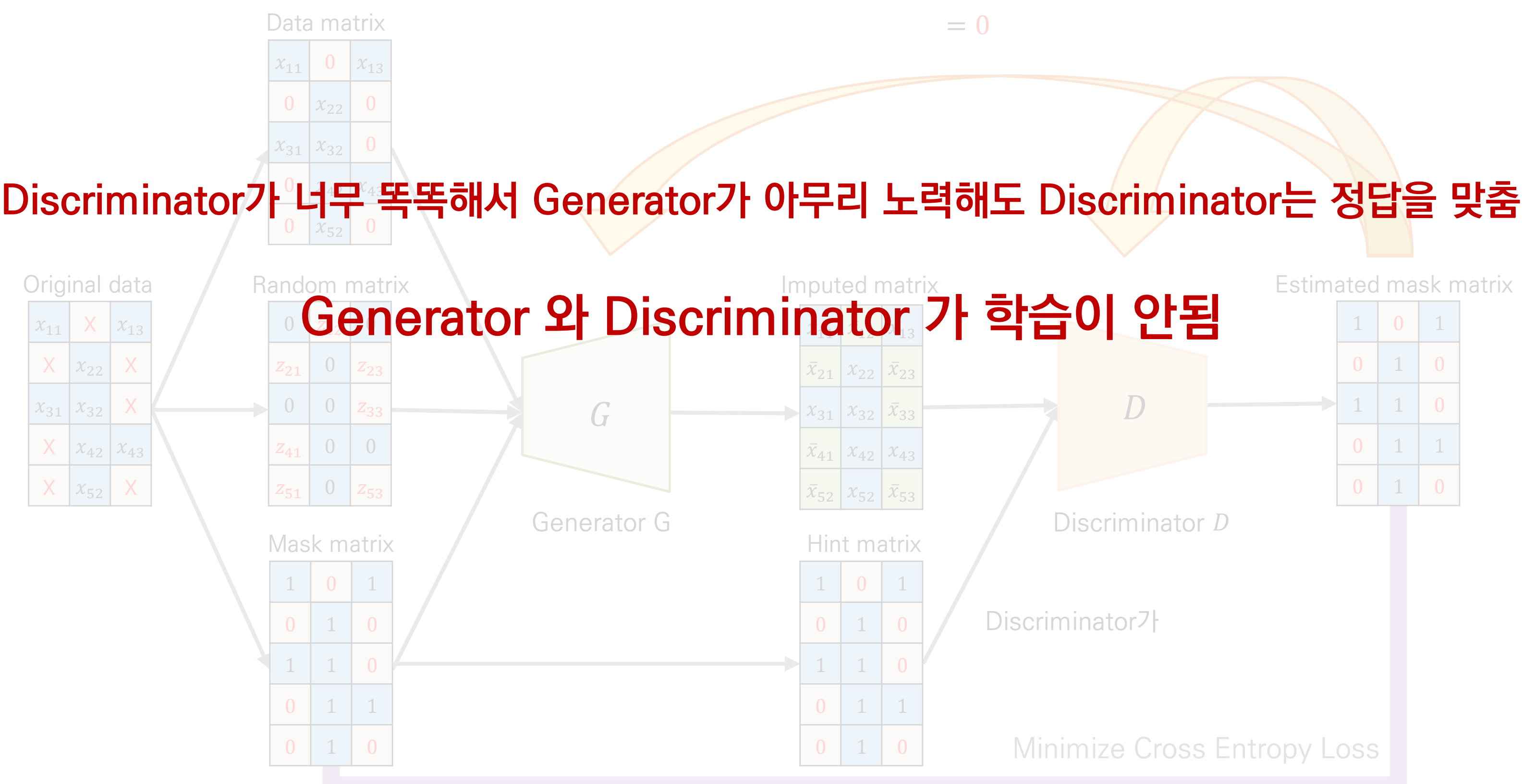
❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case2. 힌트 100%

$$-\left(\sum_{i:b_i=0} m_i \log(\widehat{m_i}) + (1-m_i)\log(1-\widehat{m_i})\right)$$

$$= 0$$



Discriminator가 너무 똑똑해서 Generator가 아무리 노력해도 Discriminator는 정답을 맞춤

Generator 와 Discriminator 가 학습이 안됨

- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

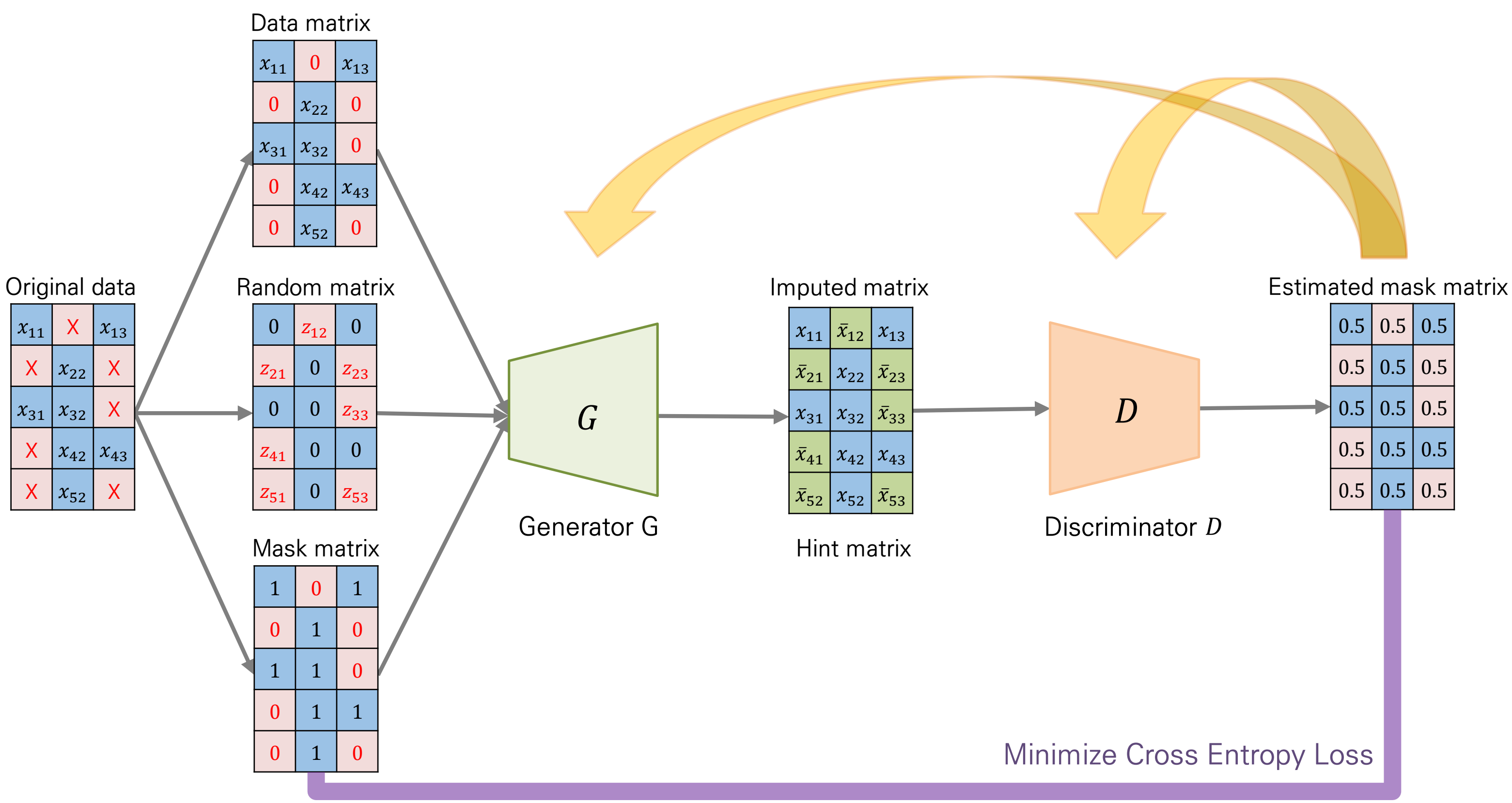
GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case3. 힌트 0%



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

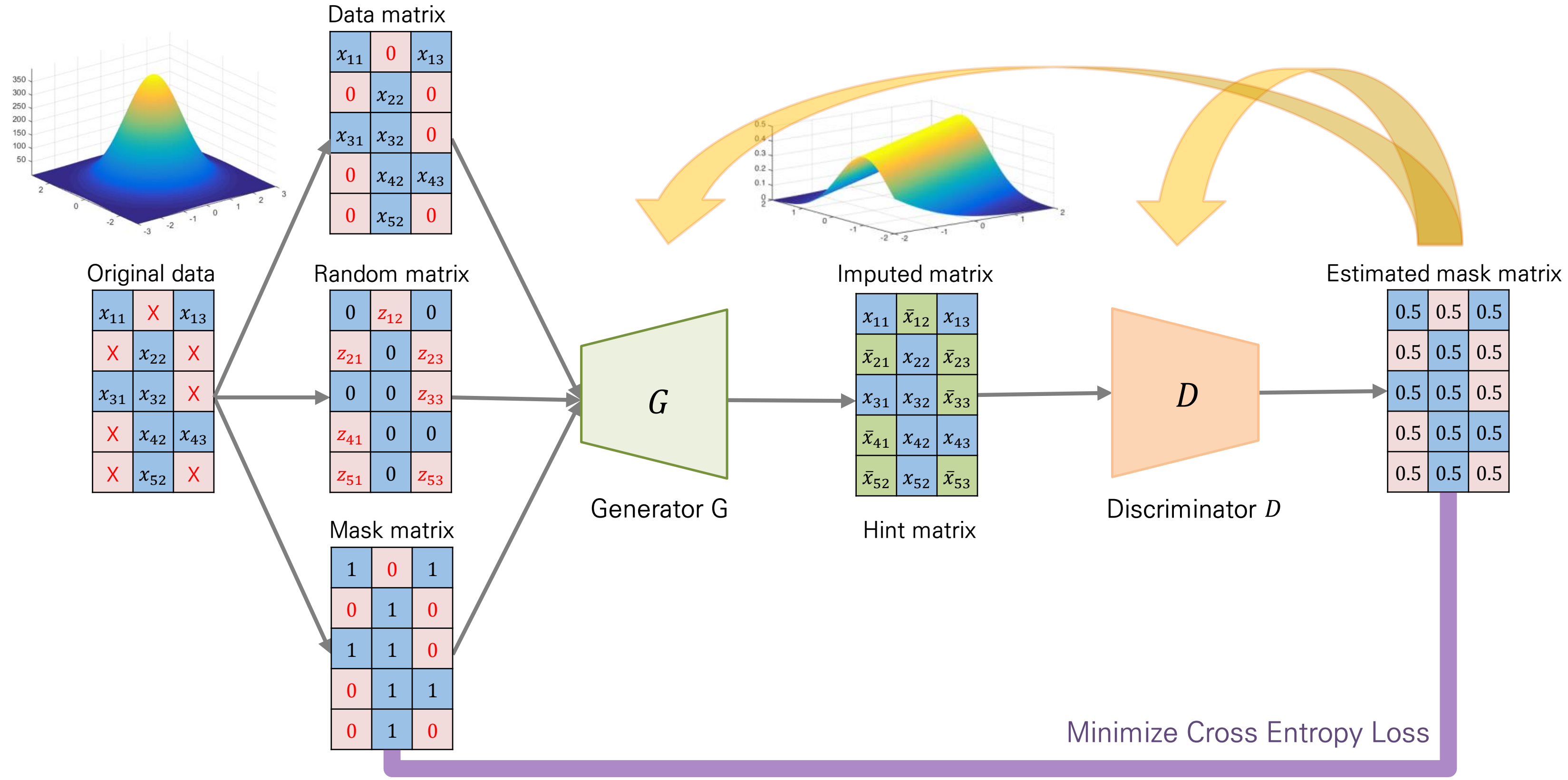
GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case3. 힌트 0%



- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크

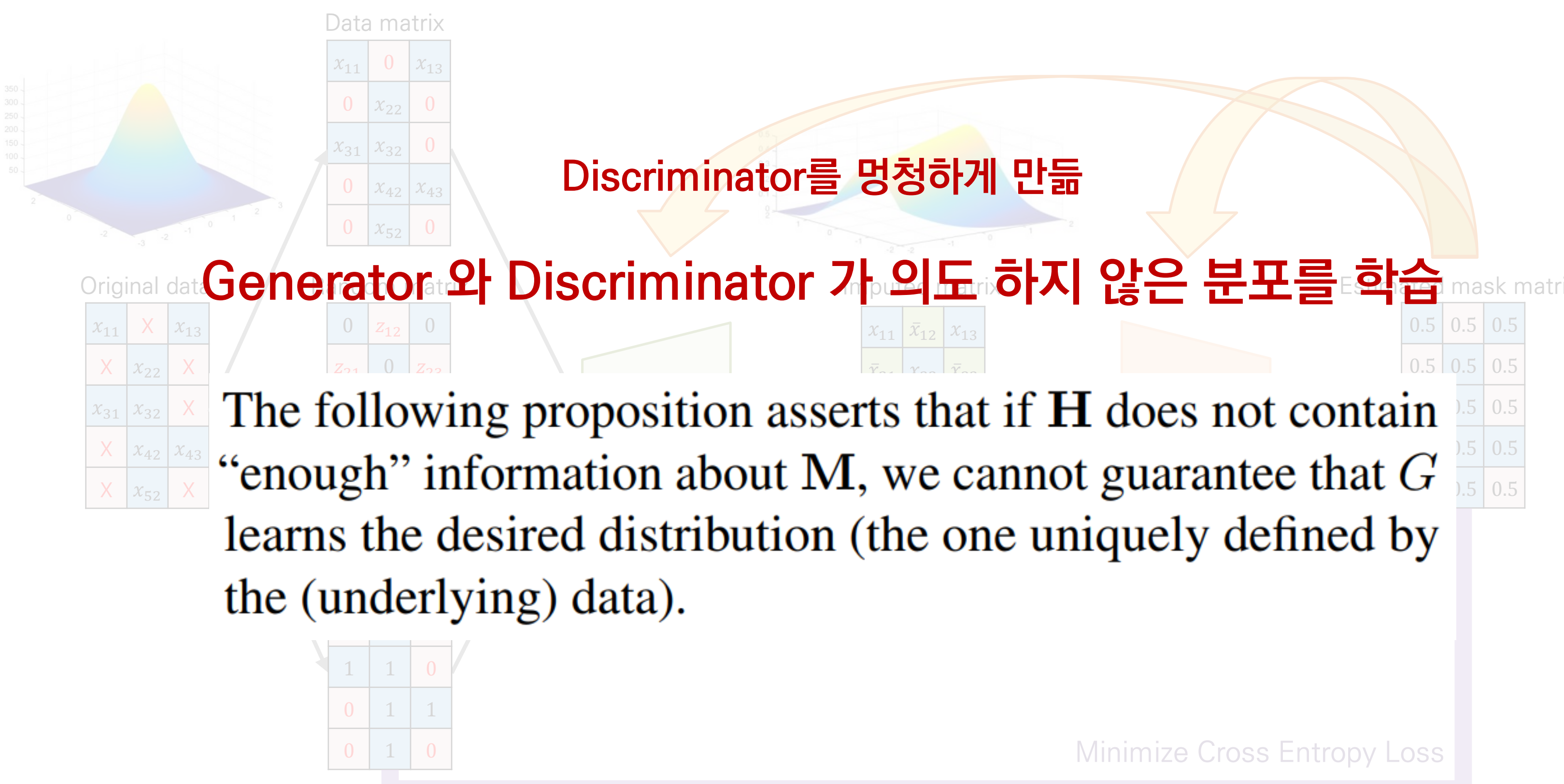
GAIN

– Hint H

❖ Mask matrix의 일부를 주입하여 discriminator학습에 가이드 제공

❖ H=0.5에 해당하는 값은 G로부터 생성된 값에 의존

Case3. 힌트 0%



Discriminator를 멍청하게 만듦

Generator 와 Discriminator 가 의도 하지 않은 분포를 학습

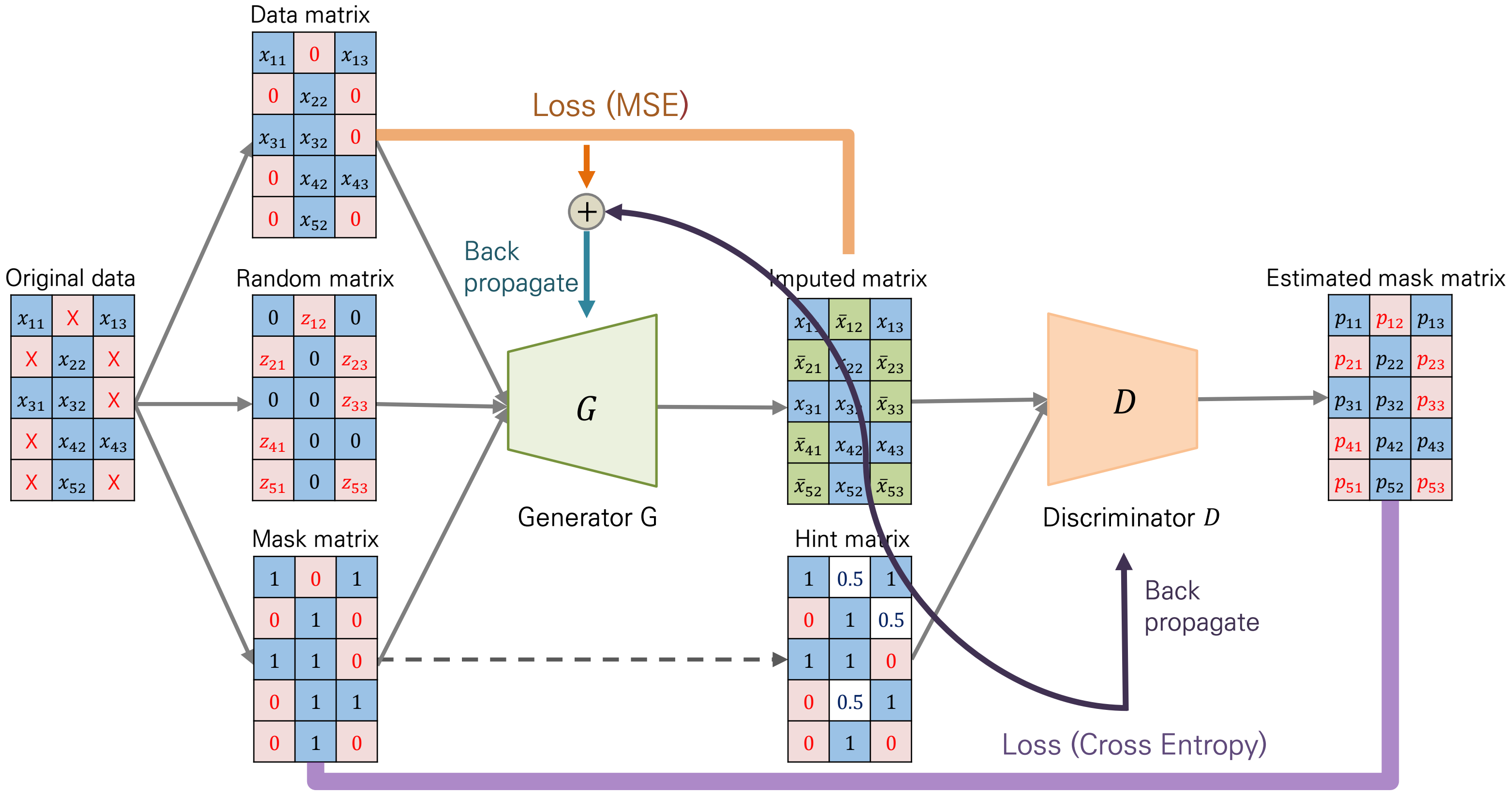
The following proposition asserts that if $\mathbf{H}$ does not contain "enough" information about $\mathbf{M}$, we cannot guarantee that G learns the desired distribution (the one uniquely defined by the (underlying) data).

GAIN

– GAIN algorithm

❖ GAIN objective function

$$\min_{G}\max_{D} V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log D(\hat{X},\mathrm{H}) + (1-\mathrm{M}^{\mathrm{T}})\log(1-D(\hat{X},\mathrm{H}))\right]$$

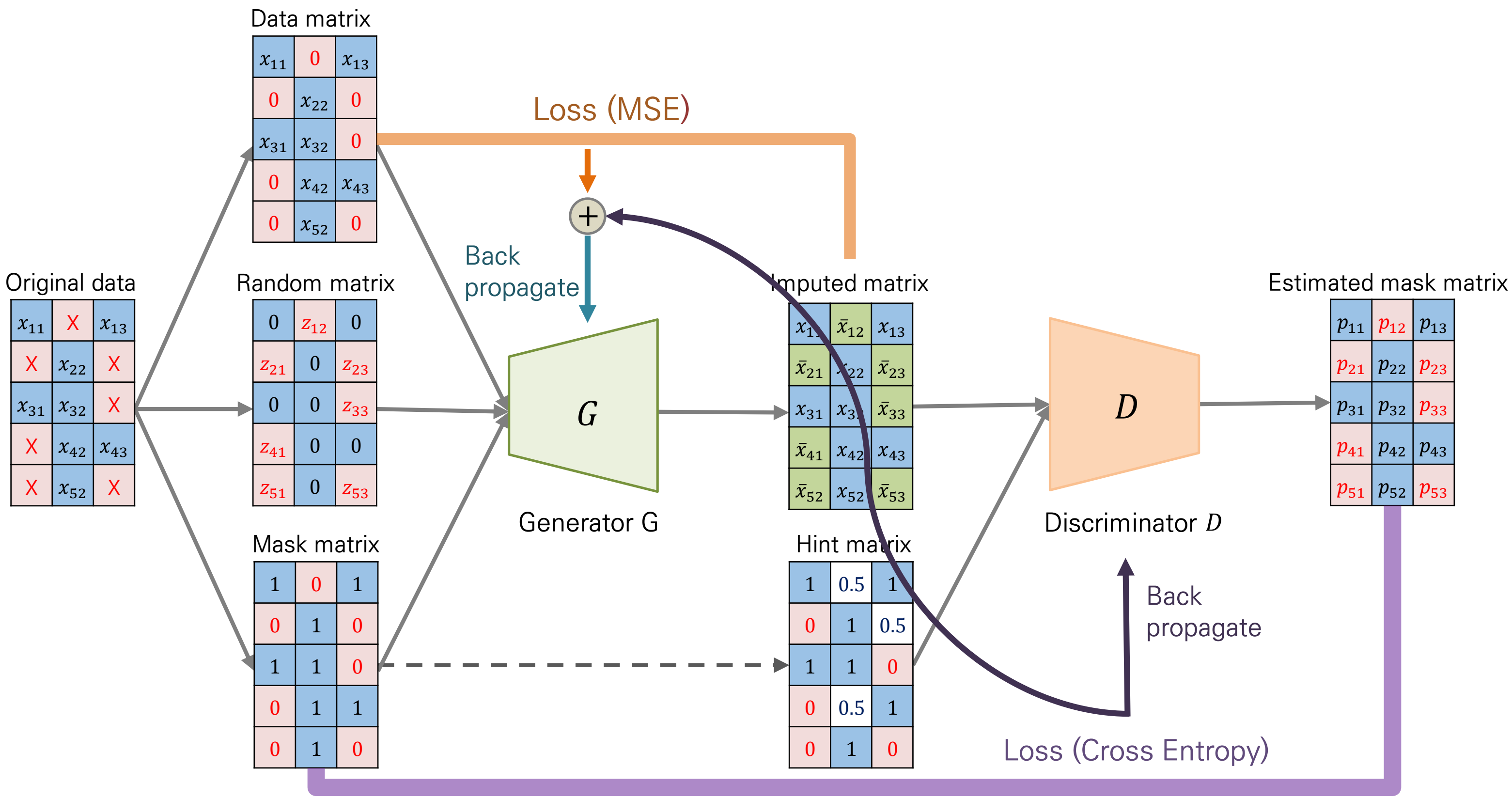


GAIN

– GAIN algorithm

❖ GAIN objective function

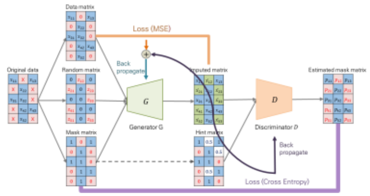
$$\min_{G}\max_{D} V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log\hat{M} + (1-\mathrm{M}^{\mathrm{T}})\log(1-\hat{M})\right]$$



GAIN

– GAIN algorithm

- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크



❖ GAIN objective function

$$\min_{G}\max_{D} V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log \widehat{M} + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log\left(1-\widehat{M}\right)\right]$$

Discriminator

- 목적: 대체 값(imputed data)와 계측 (observed)를 구별하는 네트워크
- $L_D(m,\widehat{m},\mathrm{b})$ (≤ 0)

$$= \sum_{i:b_i}\left(\underbrace{m_i \log(\widehat{m_i})}_{(1)} + \underbrace{(1-m_i)\log(1-\widehat{m_i})}_{(2)}\right)$$

(1) $m_i = 0$ (missing) 일 때, $\widehat{m_i} = 0$ 이도록
(2) $m_i = 1$ (observed) 일 때, $\widehat{m_i} = 1$ 이도록

➢ $\min_{D}\left[-\sum_{j=1}^{K_D} L_D\left(m(j),\widehat{m}(j),\mathrm{b}(j)\right)\right]$
➢ $\max_{D}\left[\sum_{j=1}^{K_D} L_D\left(m(j),\widehat{m}(j),\mathrm{b}(j)\right)\right]$

Generator

- 목적: 대체 값(imputed data)을 생성하는 네트워크
- $L_G(m,\widehat{m},\mathrm{b})$ (≥ 0)

$$= -\sum_{i:b_i}(1-m_i)\log(\widehat{m_i}))$$

$m_i = 0$ (missing) 일 때, $\widehat{m_i} = 1$ 이도록

➢ $\min_{G}\left[\sum_{j=1}^{K_D} L_G\left(m(j),\widehat{m}(j),\mathrm{b}(j)\right)\right]$

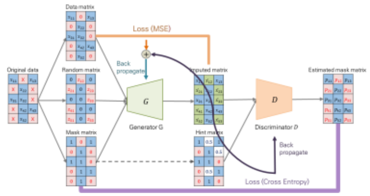
- $L_M(x_i,\hat{x}_i)$ (≥ 0)

$$= \begin{cases} m_i(x_i-\hat{x}_i)^2 & , if\ x_i\ is\ continuous \\ m_i(-x_i\log(\hat{x}_i)) & , if\ x_i\ is\ binary \end{cases}$$

$m_i = 1$ (observed) 일 때, 값 잘 예측하도록

➢ $\min_{G}\left[\sum_{j=1}^{K_D} L_M(\tilde{x}(j)-\hat{x}(j))\right]$

- Generator(G): 진짜 같은 대체 값(imputed data)을 생성하는 네트워크
- Discriminator(D): 대체 값(imputed data)과 계측 값(observed)를 구별하는 네트워크



GAIN

– GAIN algorithm

❖ GAIN objective function

$$\min_G \max_D V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log\widehat{M} + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log\left(1-\widehat{M}\right)\right]$$

Discriminator

- 목적: 대체 값(imputed data)와 계측 (observed)를 구별하는 네트워크
- $L_D(m, \widehat{m}, \mathrm{b})\ (\leq 0)$

  $= \sum_{i:b_i}(m_i \log(\widehat{m_i}) + (1-m_i)\log(1-\widehat{m_i}))$

  ➢ $\max_D[\sum_{j=1}^{K_D} L_D(m(j), \widehat{m}(j), \mathrm{b}(j))]$

Generator

- 목적: 대체 값(imputed data)을 생성하는 네트워크
- $L_G(m, \widehat{m}, \mathrm{b})\ (\geq 0)$

  $= -\sum_{i:b_i}(1-m_i)\log(\widehat{m_i}))$

  ➢ $\min_G[\sum_{j=1}^{K_D} L_G(m(j), \widehat{m}(j), \mathrm{b}(j))]$
- $L_M(x_i, \hat{x}_i)\ (\geq 0)$

  $= \begin{cases} m_i(x_i - \hat{x}_i)^2 & , if\ x_i\ is\ continuous \\ m_i(-x_i \log(\hat{x}_i)) & , if\ x_i\ is\ binary \end{cases}$

  ➢ $\min_G[\sum_{j=1}^{K_D} L_M(\tilde{x}(j) - \hat{x}(j))]$
- $\min_G[\sum_{j=1}^{K_D} L_G(m(j), \widehat{m}(j), \mathrm{b}(j)) + \alpha\ L_M(\tilde{x}(j) - \hat{x}(j))]$

GAIN

- GAIN Pseudo-code

Algorithm 1 Pseudo-code of GAIN

while training loss has not converged **do**

(1) Discriminator optimization

Draw k_D samples from the dataset $\{(\tilde{\mathbf{x}}(j), \mathbf{m}(j))\}_{j=1}^{k_D}$

Draw k_D i.i.d. samples, $\{\mathbf{z}(j)\}_{j=1}^{k_D}$, of $\mathbf{Z}$

Draw k_D i.i.d. samples, $\{\mathbf{b}(j)\}_{j=1}^{k_D}$, of $\mathbf{B}$

for $j = 1, ..., k_D$ **do**

$\bar{\mathbf{x}}(j) \leftarrow G(\tilde{\mathbf{x}}(j), \mathbf{m}(j), \mathbf{z}(j))$

$\hat{\mathbf{x}}(j) \leftarrow \mathbf{m}(j) \odot \tilde{\mathbf{x}}(j) + (\mathbf{1} - \mathbf{m}(j)) \odot \bar{\mathbf{x}}(j)$

$\mathbf{h}(j) = \mathbf{b}(j) \odot \mathbf{m}(j) + 0.5(\mathbf{1} - \mathbf{b}(j))$

end for

Update D using stochastic gradient descent (SGD)

$$\nabla_D - \sum_{j=1}^{k_D} \mathcal{L}_D(\mathbf{m}(j), D(\hat{\mathbf{x}}(j), \mathbf{h}(j)), \mathbf{b}(j))$$

(2) Generator optimization

Draw k_G samples from the dataset $\{(\tilde{\mathbf{x}}(j), \mathbf{m}(j))\}_{j=1}^{k_G}$

Draw k_G i.i.d. samples, $\{\mathbf{z}(j)\}_{j=1}^{k_G}$ of $\mathbf{Z}$

Draw k_G i.i.d. samples, $\{\mathbf{b}(j)\}_{j=1}$ of $\mathbf{B}$

for $j = 1, ..., k_G$ **do**

$\mathbf{h}(j) = \mathbf{b}(j) \odot \mathbf{m}(j) + 0.5(\mathbf{1} - \mathbf{b}(j))$

end for

Update G using SGD (for fixed D)

$$\nabla_G \sum_{j=1}^{k_G} \mathcal{L}_G(\mathbf{m}(j), \hat{\mathbf{m}}(j), \mathbf{b}(j)) + \alpha \mathcal{L}_M(\mathbf{x}(j), \tilde{\mathbf{x}}(j))$$

end while

GAIN

– Lemma1.

❖ GAIN objective function

GAIN

$$\min_G \max_D V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} logD(\hat{X},\mathrm{H}) + (1-\mathrm{M}^{\mathrm{T}})\log(1-D(\hat{X},\mathrm{H})\right]$$

$$V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} logD(\hat{X},\mathrm{H}) + (1-\mathrm{M}^{\mathrm{T}})\log(1-D(\hat{X},\mathrm{H})\right]$$

for G fixed, the optimal discriminator D is maximized when

$$D_G^*(x,h)_i = \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)} = p_m(m_i = 1|x,h) \;\; for\ each\ i \in \{1,\dots,d\}.$$

$$h_i = 0, D^*(x,h)_i = 0$$
$$h_i = 1, D^*(x,h)_i = 1 \quad , for\ all\ x \in X, i \in \{1,\dots,d\}$$

GAN

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[logD(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D(G(z))\right)\right]$$

$$\begin{aligned} V(D,G) &= E_{x\sim p_{data}(x)}\left[logD(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D(G(z))\right)\right] \\ &= E_{x\sim p_{data}(x)}[logD(x)] + E_{x\sim p_g(x)}[\log(1-D(x))] \\ &= \int_x P_{data}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx \end{aligned}$$

for G fixed, the optimal discriminator D is maximized when

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$$

GAIN

– Lemma1.

❖ GAIN objective function

GAIN

$$\min_G \max_D V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log D(\hat{X},\mathrm{H}) + (1-\mathrm{M}^{\mathrm{T}})\log(1-D(\hat{X},\mathrm{H})\right]$$

$$\begin{aligned}
C(G) &= V(D^*,G) \\
&= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log D^*(\hat{X},\mathrm{H}) + (1-\mathrm{M}^{\mathrm{T}})\log(1-D^*(\hat{X},\mathrm{H})\right] \\
&= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)} + (1-\mathrm{M}^{\mathrm{T}})\log(1-\frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)})\right] \\
&= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)} + (1-\mathrm{M}^{\mathrm{T}})\log\frac{p(x,h,m_i=0)}{p(x,h,m_i=1)+p(x,h,m_i=0)}\right] \\
&= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log(p_m(m_i=1|\hat{X},\mathrm{H})) + (1-\mathrm{M}^{\mathrm{T}})\log(p_m(m_i=0|\hat{X},\mathrm{H}))\right] \\
&= E_{\hat{X},M,G}\left[\textstyle\sum_{i:M_i=1} log(p_m(m_i=1|\hat{X},\mathrm{H})) + \sum_{i:M_i=0} log(p_m(m_i=0|\hat{X},\mathrm{H}))\right]
\end{aligned}$$

The global minimum of the virtual training criterion $C(G)$ is achieved if and ony if $\hat{p}(x|h,m_i=t) = \hat{p}(x|h)$ at that point.

Then the solution above is unique and satisfies $\hat{p}(x|m_1) = \hat{p}(x|m_2)$

GAN

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[log D(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D(G(z))\right)\right]$$

$$\begin{aligned}
C(G) &= V(D^*,G) \\
&= E_{x\sim p_{data}(x)}[log D^*(x)] + E_{x\sim P_g(x)}[\log(1-D^*(x))] \\
&= E_{x\sim p_{data}(x)}\left[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim P_g(x)}\left[\log(1-\frac{p_{data}(x)}{p_{data}(x)+p_g(x)})\right] \\
&= E_{x\sim p_{data}(x)}\left[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim P_g(x)}\left[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right] \\
&= -\log(4) + KL(p_{data} \parallel \frac{p_{data}(x)+p_g(x)}{2}) + KL(p_g \parallel \frac{p_{data}(x)+p_g(x)}{2}) \\
&= -\log(4) + 2 \times JSD(p_{data} \parallel p_g)
\end{aligned}$$

The global minimum of the virtual training criterion $C(G)$ is achieved if and ony if $p_g = p_{data}$. at that point, $C(G)$ achieves the value $-\log(4)$

Experiments

❖ Source of gains in GAIN algorithm (Mean ± Std of RMSE (Gain %))

- 목적: GAIN 손실 함수 변경에 따라 성능 차이 비교
- 실험 방식: 완전 데이터에 일부 결측을 발생시킨 뒤, 대체된 값과 실제 값 사이 RMSE확인
- 사용 데이터: UCI 데이터(완전 데이터)

Algorithm	Breast	Spam	Letter	Credit	News
GAIN	**.0546 ± .0006**	**.0513± .0016**	**.1198± .0005**	**.1858 ± .0010**	**.1441 ± .0007**
GAIN w/o $\mathcal{L}_G$	.0701 ± .0021 **(22.1%)**	.0676 ± .0029 **(24.1%)**	.1344 ± .0012 **(10.9%)**	.2436 ± .0012 **(23.7%)**	.1612 ± .0024 **(10.6%)**
GAIN w/o $\mathcal{L}_M$	.0767 ± .0015 **(28.9%)**	.0672 ± .0036 **(23.7%)**	.1586 ± .0024 **(24.4%)**	.2533 ± .0048 **(26.7%)**	.2522 ± .0042 **(42.9%)**
GAIN w/o Hint	.0639 ± .0018 **(14.6%)**	.0582 ± .0008 **(11.9%)**	.1249 ± .0011 **(4.1%)**	.2173 ± .0052 **(14.5%)**	.1521 ± .0008 **(5.3%)**
GAIN w/o Hint & $\mathcal{L}_M$	.0782 ± .0016 **(30.1%)**	.0700 ± .0064 **(26.7%)**	.1671 ± .0052 **(28.3%)**	.2789 ± .0071 **(33.4%)**	.2527 ± .0052 **(43.0%)**

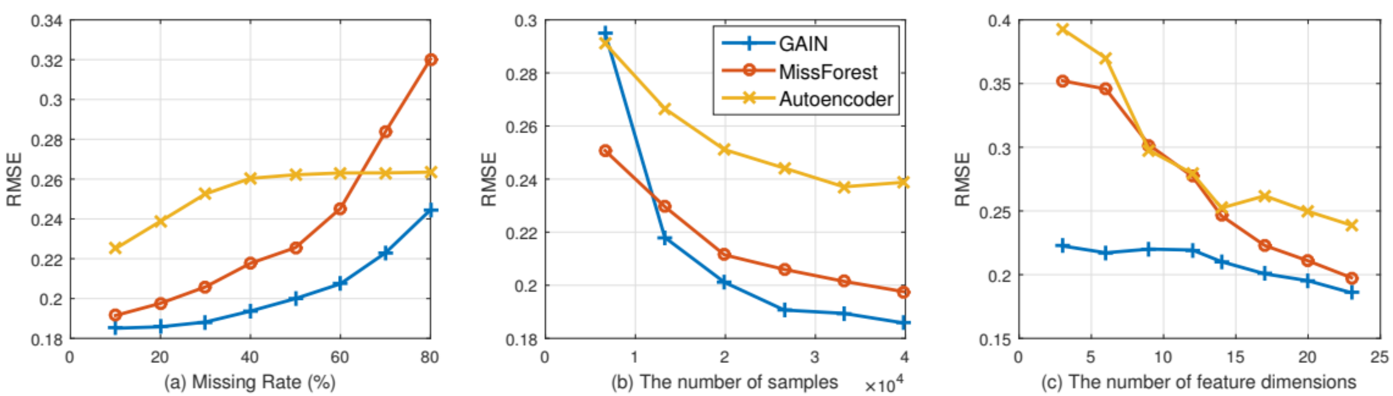
Experiments

❖ Quantitative analysis of GAIN

➢ 목적: GAIN과 결측치 대체 알고리즘 성능 차이 비교

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



Experiments

❖ Prediction Performance

- ➢ 목적: 대체된 결측치로 모델 성능 개선 정도 비교
- ➢ 실험 방식: 대체된 값을 통해 완전 데이터 형성 후, 분류 모델의 성능(AUROC*)확인
- ➢ 사용 데이터: UCI 데이터(완전 데이터)
- ➢ 사용 분류 모델: logistic regression (binary classification)

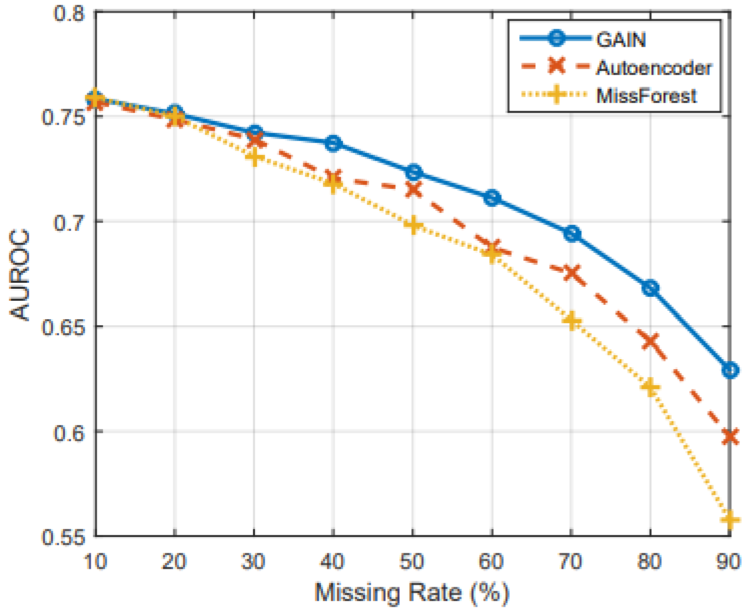
Algorithm	AUROC (Average ± Std)			
	Breast	Spam	Credit	News
GAIN	**.9930 ± .0073**	**.9529 ± .0023**	**.7527 ± .0031**	**.9711 ± .0027**
MICE	.9914 ± .0034	.9495 ± .0031	.7427 ± .0026	.9451 ± .0037
MissForest	.9860 ± .0112	.9520 ± .0061	.7498 ± .0047	.9597 ± .0043
Matrix	.9897 ± .0042	.8639 ± .0055	.7059 ± .0150	.8578 ± .0125
Auto-encoder	.9916 ± .0059	.9403 ± .0051	.7485 ± .0031	.9321 ± .0058
EM	.9899 ± .0147	.9217 ± .0093	.7390 ± .0079	.8987 ± .0157

AUROC(the Area Under a ROC Curve)*: ROC 커브의 밑 면적 값, 1에 가까울수록 민감도 특이도 모두 만족

Experiments

❖ Prediction Performance

- 목적: 대체된 결측치로 모델 성능 개선 정도 비교
- 실험 방식: 대체된 값을 통해 완전 데이터 형성 후, 분류 모델의 성능(AUROC*)확인
- 사용 데이터: UCI 데이터(완전 데이터)
- 사용 분류 모델: logistic regression (binary classification)



Experiments

❖ Prediction Performance

- 목적: 구축된 모델의 파라미터 차이 확인
- 실험 방식: 완전한 데이터를 기반으로 형성한 분류모델의 파라미터(w), 임의로 결측 생성 이후 대체된 값을 통해 완전 데이터 형성 후, 분류 모델의 파라미터($\hat{w}$)차이 비교
- 사용 데이터: UCI 데이터(완전 데이터)
- 사용 분류 모델: logistic regression (binary classification)

Algorithm	**Mean Bias** ($\|\|\mathbf{w} - \hat{\mathbf{w}}\|\|_1$)	**MSE** ($\|\|\mathbf{w} - \hat{\mathbf{w}}\|\|_2$)
GAIN	**0.3163± 0.0887**	**0.5078± 0.1137**
MICE	0.8315 ± 0.2293	0.9467 ± 0.2083
MissForest	0.6730 ± 0.1937	0.7081 ± 0.1625
Matrix	1.5321 ± 0.0017	1.6660 ± 0.0015
Auto-encoder	0.3500 ± 0.1503	0.5608 ±0.1697
EM	0.8418 ± 0.2675	0.9369 ± 0.2296

Conclusion

❖ Compare with simply GAN (GAN의 특징을 목적에 맞게 접목)

➢ Generator G

- Target distribution $P(X|\tilde{X} = \tilde{x}^i)$, need $\|1 - M\|_1$–dimensional random variable z
- 추가적으로 Mask vector를 주입

➢ Discriminator D

- 데이터 전체에 대한 (real/fake)판별이 아닌 각 원소(components)가 (real/fake)인지 판별

❖ generative model기반 결측치 대체를 제안하여 범용성 확보

❖ 다양한 실험을 통해 해당 알고리즘의 우수성 확인

Thank you

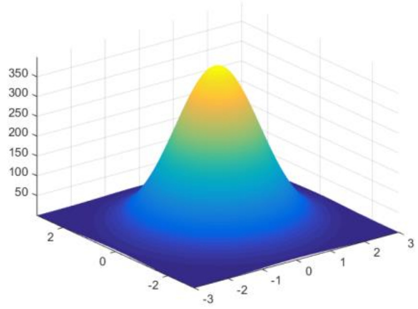
Appendix

Introduction

– Discriminative models vs Generative models

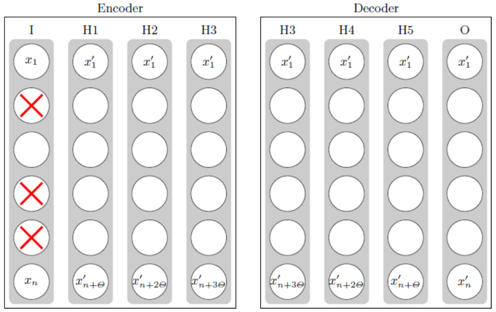
❖ Expectation-Maximization Multiple Imputation

- 데이터가 정규분포로부터 생성되었다고 가정
- EM 알고리즘을 이용하여 결측치를 포함하고 있는 데이터의 incomplete likelihood function을 최대화하는 MLE 계산
- 추정된 MLE를 기반으로 결측치의 기댓값을 도출하여 결측치 대체



❖ MIDA – Multiple Imputation using Denoising Autoencoders

- Denoising Autoencoder(DAE) 구조를 활용하여 결측치 대체 방법론 제안
- DAE를 통해 원본 데이터를 재구축(reconstruct)하여 그 차이를 최소화하도록 학습

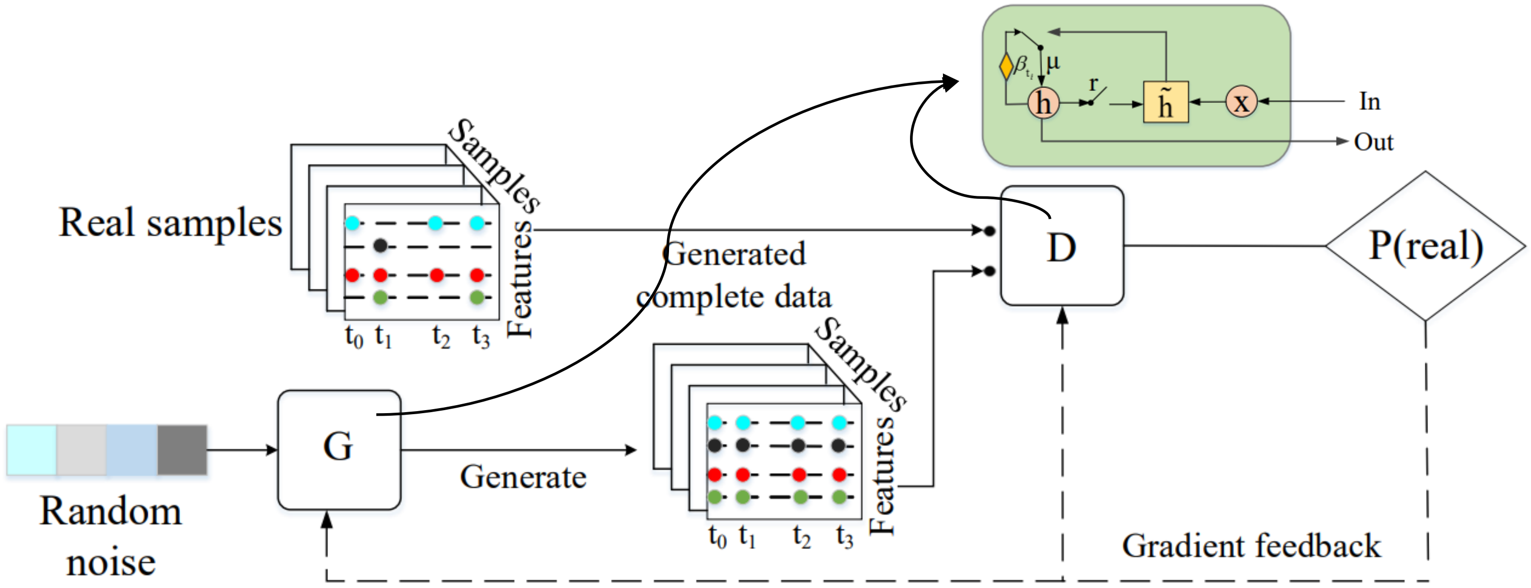


Related Works

– Multivariate Time Series Imputation with Generative Adversarial Networks

❖ GAN구조를 변형하여 시계열 데이터의 특성을 반영

- ➢ 모델의 구조를 GRU를 기반으로 구축 (GRUI cell제안)
- ➢ GAIN과 마찬가지로 mask vecto사용
- ➢ Discriminator가 실제 값들의 차이를 줄이도록 학습



Luo, Y., Cai, X., Zhang, Y., & Xu, J. (2018). Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems* (pp. 1603-1614).

Related Works

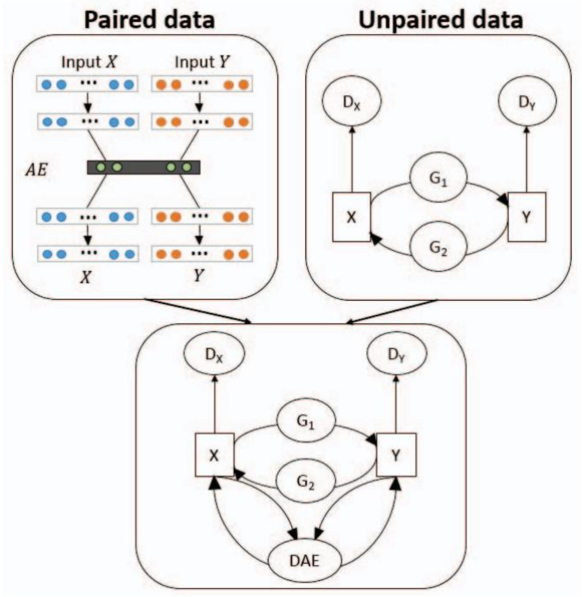
– VIGAN: Missing view imputation with generative adversarial networks

❖ GAN구조를 변형하여 multi-view 데이터의 특성을 반영

➢ Multi-view 데이터의 특성을 각 다른 도메인으로 인지

➢ Unpaired data set의 이미지 합성 연구 cycle GAN에서 제안한 cycle consistency loss를 반영

➢ Paired data의 shared representation을 도출하기 위해 DAE를 사용



Shang, C., Palmer, A., Sun, J., Chen, K. S., Lu, J., & Bi, J. (2017, December). VIGAN: Missing view imputation with generative adversarial networks. In *Big Data (Big Data), 2017 IEEE International Conference on* (pp. 766-775). IEEE.

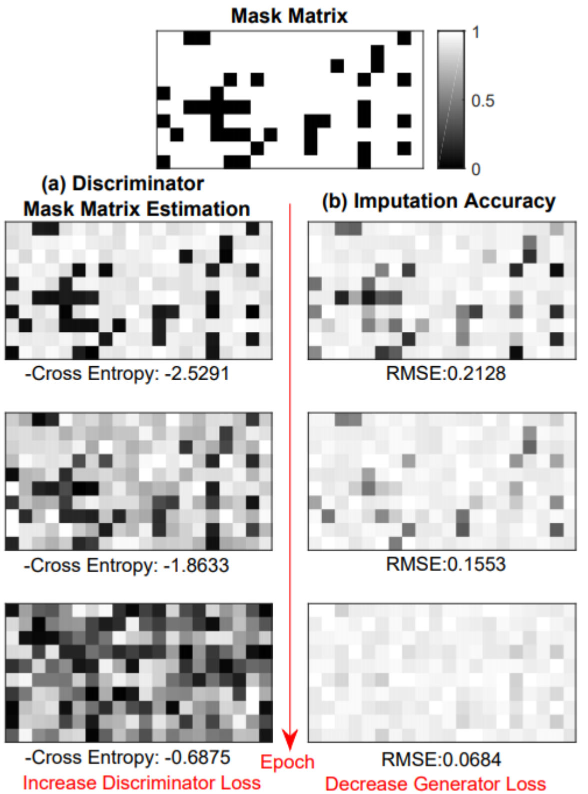
Generative Adversarial Imputation Nets

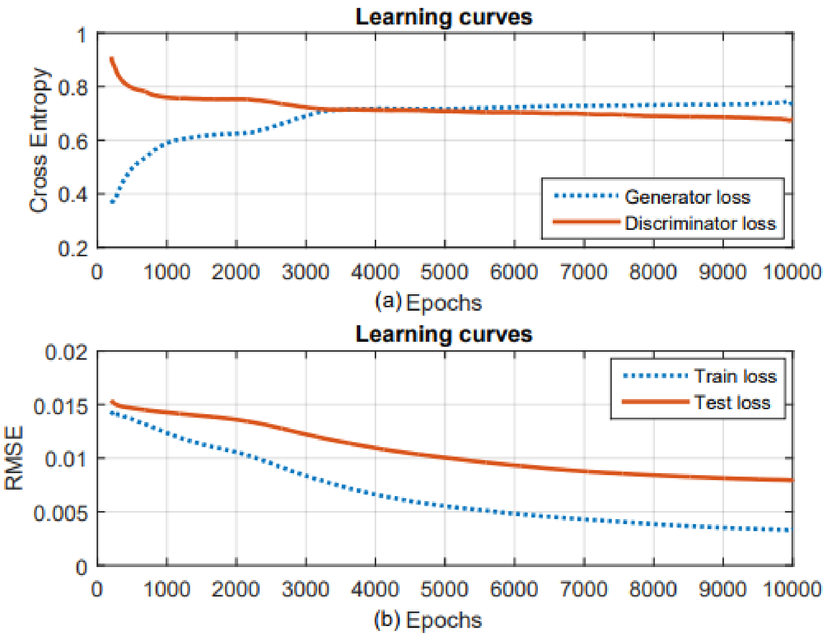
– Experiments

❖ Visualization of the convergence of GAIN

➢ (a) Discriminator output (estimated mask matrix)

➢ (b) imputation accuracy of the generator사용 분류 모델: logistic regression





GAIN

– Lemma1. (supplementary materials)

❖ GAIN objective function

1.1. Proof of Lemma 1

For $t \in \{0,1\}$ define the set $M_t^i = \{\mathbf{m} \in \{0,1\}^d : m_i = t\}$.

$$\begin{aligned}
V(D,G) &= \mathbb{E}_{\tilde{\mathbf{X}},\mathbf{z},\mathbf{M},\mathbf{H}}\left[\mathbf{M}^T \log D(G(\tilde{\mathbf{X}},\mathbf{M}),\mathbf{H}) + (\mathbf{1}-\mathbf{M})^T \log\left(\mathbf{1} - D(G(\tilde{\mathbf{X}},\mathbf{M}),\mathbf{H})\right)\right] \\
&= \mathbb{E}_{\hat{\mathbf{X}},\mathbf{M},\mathbf{H}}\left[\mathbf{M}^T \log D(\hat{\mathbf{X}},\mathbf{H}) + (\mathbf{1}-\mathbf{M})^T \log\left(\mathbf{1} - D(\hat{\mathbf{X}},\mathbf{H})\right)\right] \\
&= \int_{\mathcal{X}} \sum_{\mathbf{m}\in\{0,1\}^d} \int_{\mathcal{H}} \left(\mathbf{m}^T \log D(\mathbf{x},\mathbf{h}) + (\mathbf{1}-\mathbf{m})^T \log(\mathbf{1} - D(\mathbf{x},\mathbf{h}))\right) p(\mathbf{x},\mathbf{m},\mathbf{h}) d\mathbf{h} d\mathbf{x} \\
&= \int_{\mathcal{X}} \int_{\mathcal{H}} \sum_{\mathbf{m}\in\{0,1\}^d} \left(\sum_{i:m_i=1} \log D(\mathbf{x},\mathbf{h})_i + \sum_{i:m_i=0} \log(\mathbf{1} - D(\mathbf{x},\mathbf{h})_i)\right) p(\mathbf{x},\mathbf{m},\mathbf{h}) d\mathbf{h} d\mathbf{x} \\
&\overset{(1)}{=} \int_{\mathcal{X}} \int_{\mathcal{H}} \sum_{i=1}^d \left(\sum_{\mathbf{m}\in M_1^i} \log D(\mathbf{x},\mathbf{h})_i + \sum_{\mathbf{m}\in M_0^i} \log(\mathbf{1} - D(\mathbf{x},\mathbf{h})_i)\right) p(\mathbf{x},\mathbf{m},\mathbf{h}) d\mathbf{h} d\mathbf{x} \\
&= \int_{\mathcal{X}} \int_{\mathcal{H}} \sum_{i=1}^d \left(\log D(\mathbf{x},\mathbf{h})_i \sum_{\mathbf{m}\in M_1^i} p(\mathbf{x},\mathbf{m},\mathbf{h})\right) + \left(\log(\mathbf{1} - D(\mathbf{x},\mathbf{h})_i) \sum_{\mathbf{m}\in M_0^i} p(\mathbf{x},\mathbf{m},\mathbf{h})\right) d\mathbf{h} d\mathbf{x} \\
&= \int_{\mathcal{X}} \int_{\mathcal{H}} \sum_{i=1}^d \log D(\mathbf{x},\mathbf{h})_i p(\mathbf{x},\mathbf{h},m_i=1) + \log(\mathbf{1} - D(\mathbf{x},\mathbf{h})_i) p(\mathbf{x},\mathbf{h},m_i=0) d\mathbf{h} d\mathbf{x}
\end{aligned}$$

where (1) follows from switching the order of summation. We then note that $y \mapsto a \log y + b \log(1-y)$ achieves its maximum in $[0,1]$ at $\frac{a}{a+b}$ and so $V(D,G)$ is maximized (for fixed G) when

$$D(\mathbf{x},\mathbf{h})_i = \frac{p(\mathbf{x},\mathbf{h},m_i=1)}{p(\mathbf{x},\mathbf{h},m_i=0) + p(\mathbf{x},\mathbf{h},m_i=1)} \tag{1}$$

GAIN

– Lemma1.

❖ GAIN objective function

$$\min_G \max_D V(D,G) = E_{\hat{X},M,G}\left[\, \mathrm{M}^{\mathrm{T}} log D\left(\hat{X},\mathrm{H}\right) + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(1-D\left(\hat{X},\mathrm{H}\right)\right]$$

$$V(D,G) \ = E_{\hat{X},M,G}\left[\, \mathrm{M}^{\mathrm{T}} log D\left(\hat{X},\mathrm{H}\right) + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(1-D\left(\hat{X},\mathrm{H}\right)\right]$$

$for\ G\ fixed, the\ optimal\ discriminator\ D\ is\ maximized\ when$

$$D_G^*(x,h)_i = \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)}$$
$$= p_m(m_i = 1|x,h)\ \ for\ each\ i\ \in \{1,\dots,d\}.$$

$$h_i = 0, D^*(x,h)_i = 0$$
$$h_i = 1, D^*(x,h)_i = 1 \quad , for\ all\ x \in X, i \in \{1,\dots,d\}$$

GAIN

– Lemma1.

❖ GAN objective function

$$\min_{G}\max_{D} V(D,G) = E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log D\left(\hat{X},\mathrm{H}\right) + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(1-D\left(\hat{X},\mathrm{H}\right)\right]$$

$$C(G) = V(D^{*},G)$$

$$= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log D^{*}\left(\hat{X},\mathrm{H}\right) + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(1-D^{*}\left(\hat{X},\mathrm{H}\right)\right]$$

$$= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)} + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(1-\frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)})\right]$$

$$= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log \frac{p(x,h,m_i=1)}{p(x,h,m_i=1)+p(x,h,m_i=0)} + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log\frac{p(x,h,m_i=0)}{p(x,h,m_i=1)+p(x,h,m_i=0)}\right]$$

$$= E_{\hat{X},M,G}\left[\mathrm{M}^{\mathrm{T}} log(p_m\left(m_i = 1\middle|\hat{X},\mathrm{H}\right)) + \left(1-\mathrm{M}^{\mathrm{T}}\right)\log(p_m\left(m_i = 0\middle|\hat{X},\mathrm{H}\right))\right]$$

$$= E_{\hat{X},M,G}\left[\sum_{i:M_i=1} log(p_m\left(m_i = 1\middle|\hat{X},\mathrm{H}\right)) + \sum_{i:M_i=0} log(p_m\left(m_i = 0\middle|\hat{X},\mathrm{H}\right))\right]$$

The global minimum of the virtual training criterion $C(G)$ *is achieved if and ony if* $\hat{p}(x|h, m_i = t) = \hat{p}(x|h)$ *at that point.*

Then the solution above is unique and satisfies $\hat{p}(x|m_1) = \hat{p}(x|m_2)$