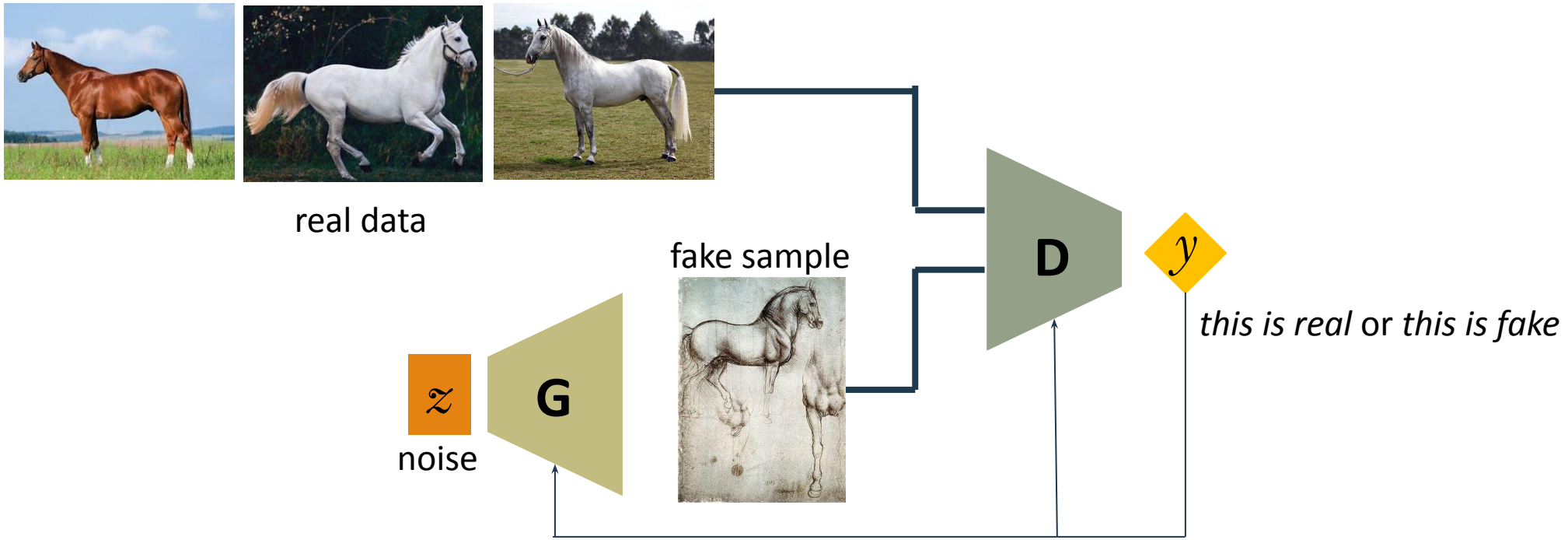
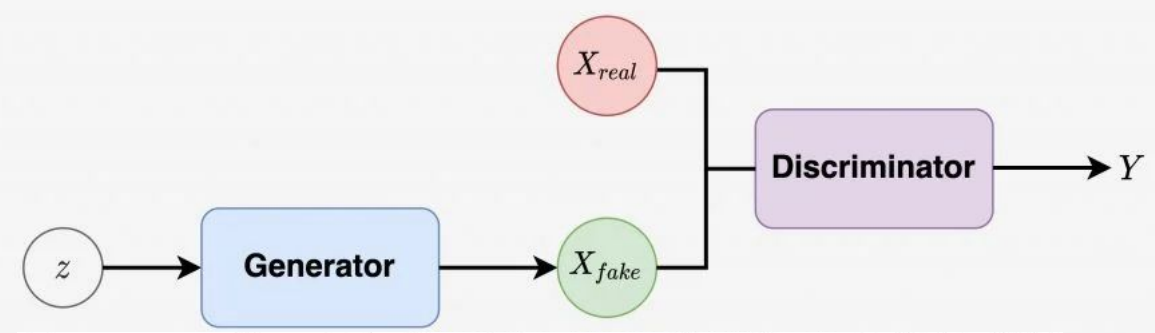
Generating Synthetic Tabular Data with GANs

JAMAL TOUTOUH
jamal@lcc.uma.es
jamal.es
@jamtou

Review basic ideas about GANs



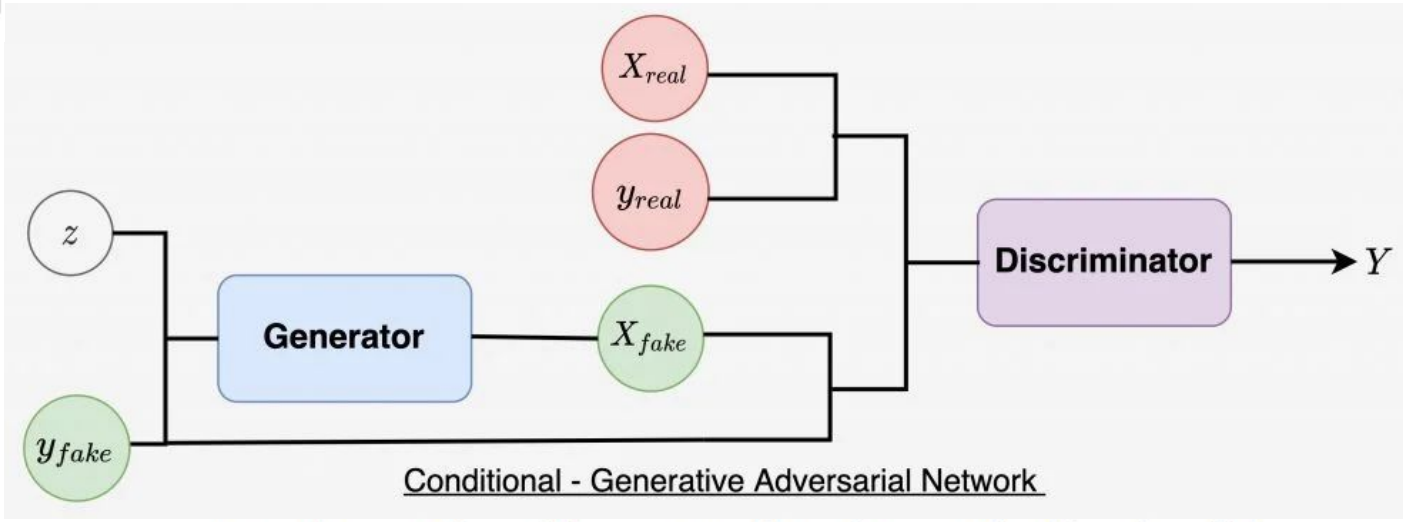
GANs vs cGAN



GANs

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

cGAN



$$\mathop{\mathbb{E}}_{x\sim p_{data}(x)}[\log D(x,y)] + \mathop{\mathbb{E}}_{z\sim p_z(z)}[\log(1 - D(G(z,y),y))]$$

Why synthetic data?

- **Data Privacy:** Synthetic data is a great way to ensure data privacy while being able to share microdata, allowing organizations to share sensitive and personal (synthetic) data without concerns with privacy regulations
- **Prototype Development**: Collecting and modeling tremendous amounts of real data is a complicated and tedious process. Generating synthetic data makes data available sooner. Besides that, it can help in faster iteration through the data collections development for ML initiatives
- **Edge-case Simulation**: It is often seen that the collected data do not contain every possible scenario which affects the model performance negatively. In such cases, we can include those rare scenarios by artificially generating them

https://news.mit.edu/2020/real-promise-synthetic-data-1016

Challenges

- **Mixed data types:** Numerical data, Categorical data (ordinal, low cardinality, etc.) , Text, Boolean
- **Sparse data**
- **Unbalanced data**
- ...

How to deal with tabular data

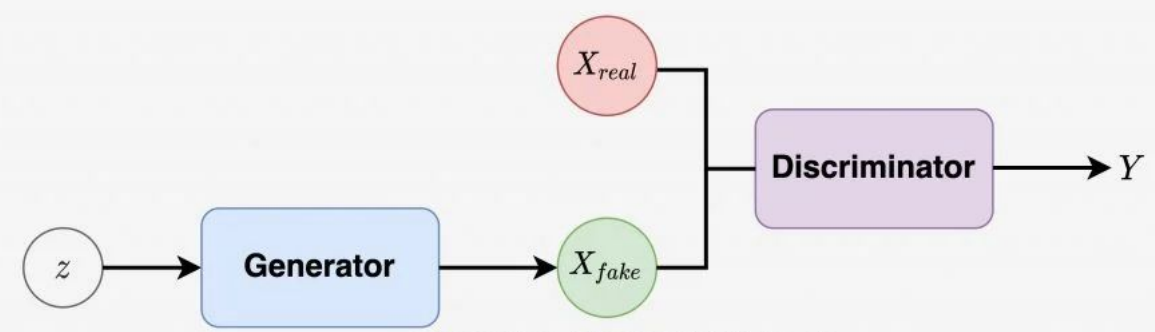
- Rows are treated as data samples
  - **One row** is **one data sample**
- **GANs** (unsupervised learning) used to **randomly** create samples (data is generated randomly)
- **cGANs** (supervised learning) used to create samples by **selecting a given category**
  - One categorical column is used as the label

How to deal with tabular data

- Every columns should be defined as a **numerical (float) value**
  - It is important to find the right distribution/transformation that fits the data
- **sklearn.preprocessing** → Preprocessing and Normalization
  - **MinMaxScaler**
  - **Normalizer**
  - **OneHotEncoder**
  - **PowerTransformer**

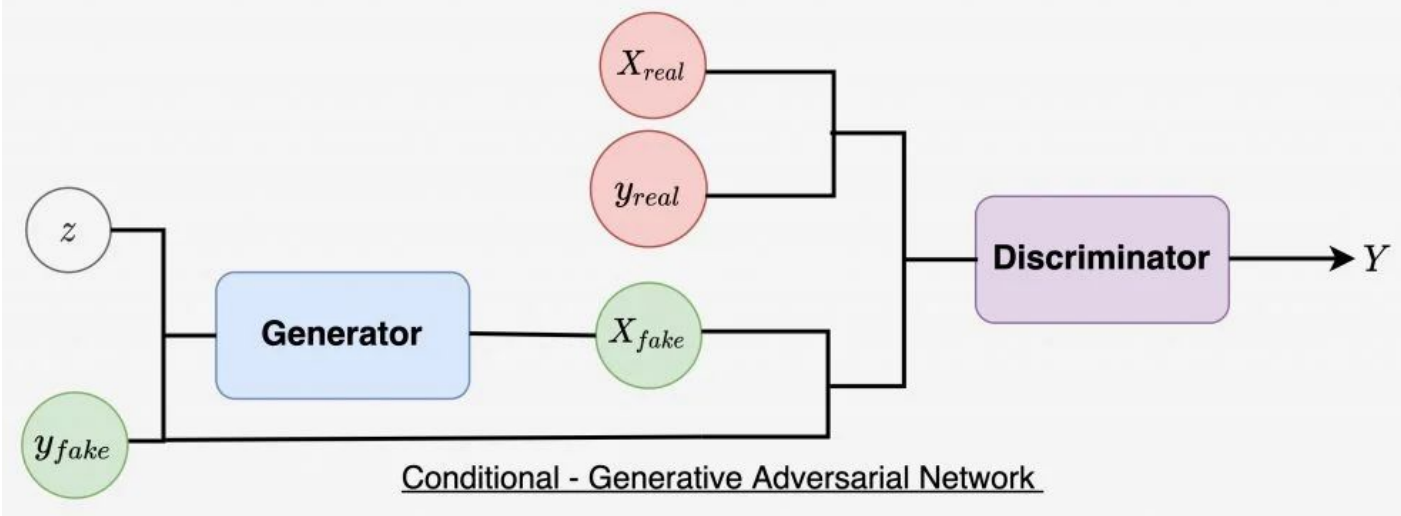
How to deal with tabular data

- Deal with the problem as we have already seen



GANs

cGAN



Example

- Medical data ⟶ **GAN**

  https://colab.research.google.com/drive/1q-IQlHWV3Izp2JBzBiLJ2uf2cauAl1wE

CTGAN

- CTGAN is specifically created for tabular data
  - Mixed data types
  - Non-Gaussian distributions
  - Multimodal distributions
  - Learning from sparse one-hot-encoded vectors
  - Highly imbalanced categorical columns

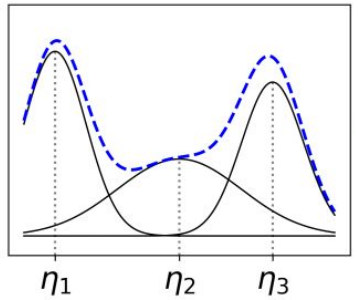
CTGAN

- **Mode-specific Normalization**

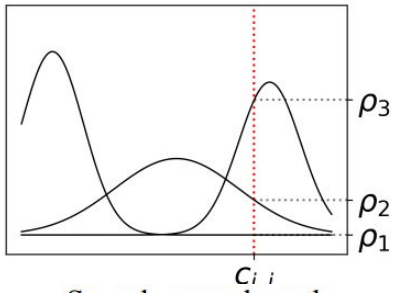
  Previously **min-max normalization** to normalize continuous values

  Three main steps:

Model the distribution of a continuous column with VGM.



For each value, compute the probability of each mode.



Sample a mode and normalize the value.

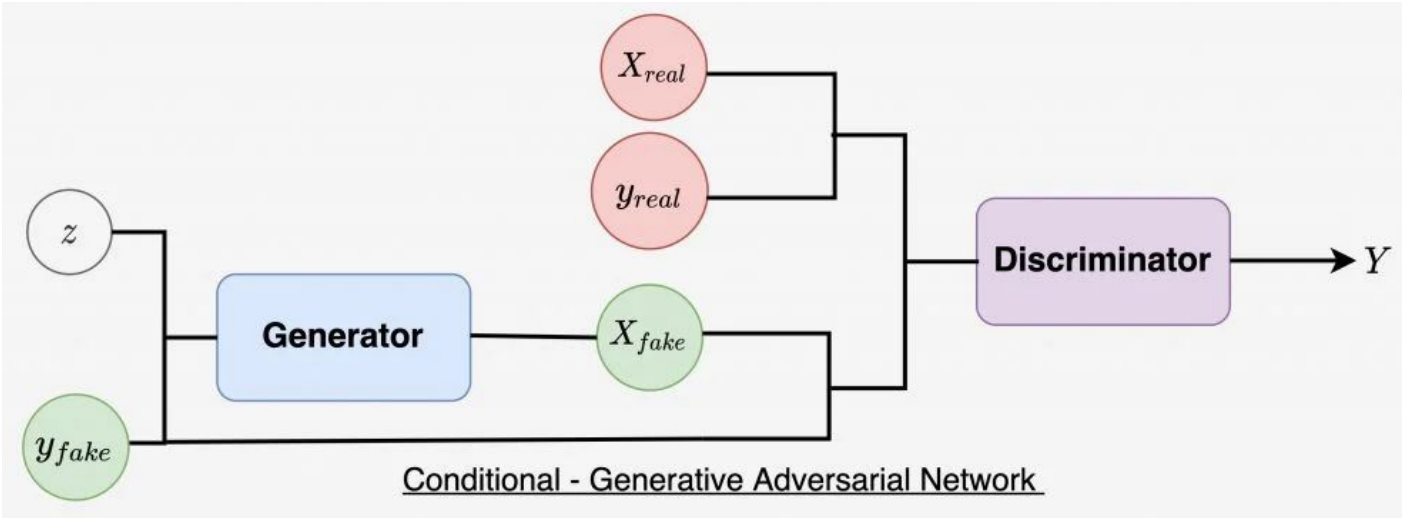
$$\alpha_{i,j} = \frac{c_{i,j} - \eta_3}{4\phi_3}$$

$$\beta_{i,j} = [0, 0, 1]$$

CTGAN

- CTGAN is a **conditional GAN**
  - What is the condition?

cGAN



$$\mathop{\mathbb{E}}_{x \sim p_{data}(x)} [\log D(x, y)] + \mathop{\mathbb{E}}_{z \sim p_z(z)} [\log(1 - D(G(z, y), y))]$$

CTGAN

- **Conditional vectors**
  - Concat all binary vectors that represent one hot representation of discrete columns to fairly sample all the classes

    For example, we have two discrete columns D_1 and D_2. Where D_1 has three classes and D_2 has two, i.e., D_1 = {1, 2, 3} and D_2 = {1, 2}. One hot vectors needed to represent them are of size 3 and size 2, respectively.

    E.g., D_1 -> [0,1,0] represents class 2 and D_2 -> [1,0] represents class 1

    So, if concat D_1 and D_2, we get [0, 1, 0, 1, 0]

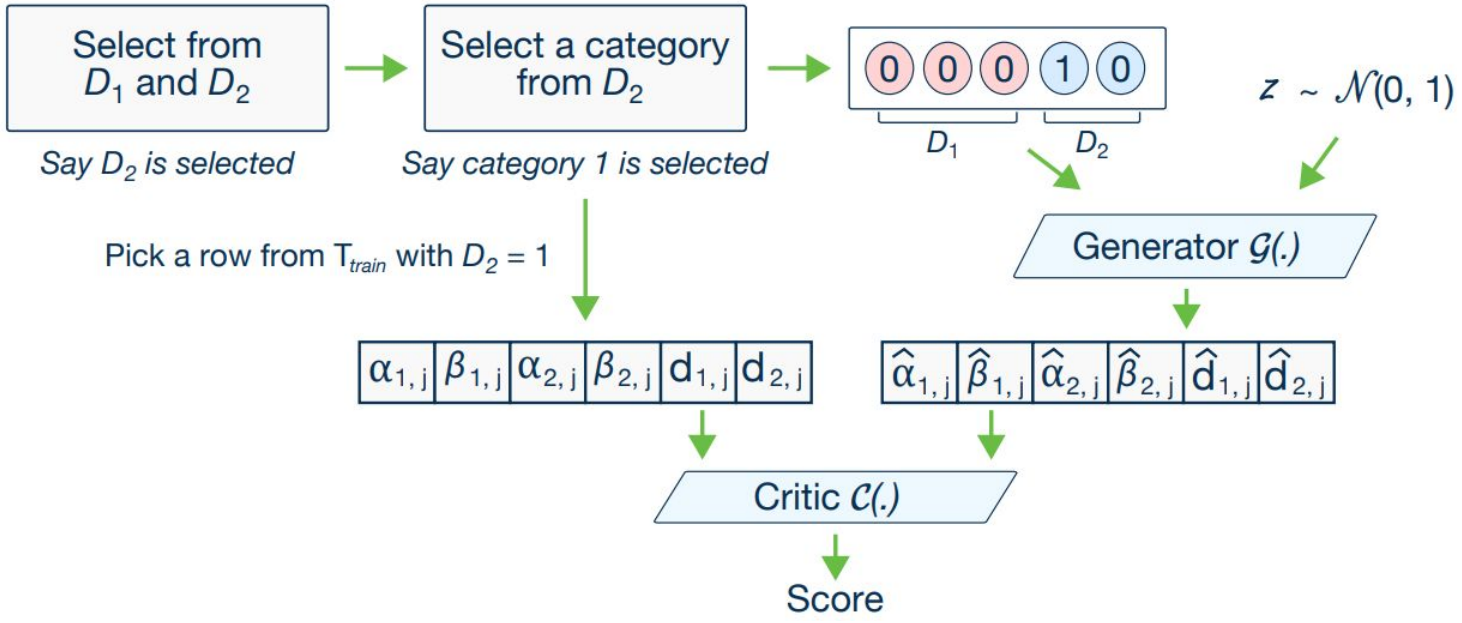
    For each training sample, just one of the columns is masked to select one element of that column. For example, to select the element in column D_2we use the mask [0, 0, 0, 1, 1], so the conditional vector is [0, 0, 0, 1, 0]

    With this condition the real data selected for the training should be any one that has the class 1 in the column D_2.

CTGAN

- **Generator loss**

  It applies a mechanism to enforce the conditional generator to produce the class in the discrete column defined by the conditional vector. It penalizes by adding the cross entropy loss between the generated class in the synthetic sample and the class defined in the conditional vector.



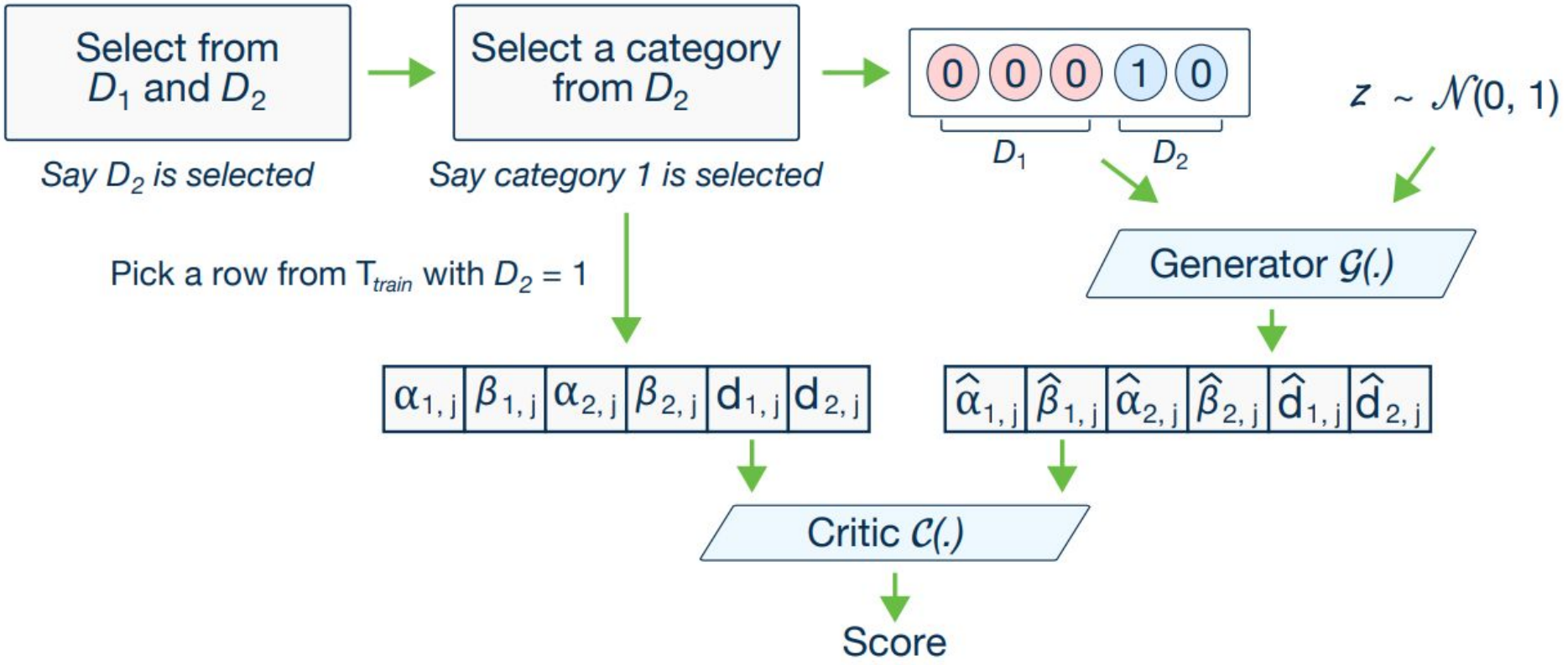
CTGAN

- **Training-by-sampling**

  For a fairly training, it samples training data for each column and each class of the column. The idea is to select training data that represent the whole training distribution.

  - All the columns have the same probability to be sampled.
  - All the classes in the column have a probability proportional to the frequency in the class.

CTGAN


Select from D_1 and D_2
Say D_2 is selected
Select a category from D_2
Say category 1 is selected
0 0 0 1 0
D_1
D_2
$z \sim \mathcal{N}(0, 1)$
Pick a row from T_{train} with $D_2 = 1$
Generator $\mathcal{G}(.)$
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Critic $\mathcal{C}(.)$
Score

Example

- Medical data ⟶ **CTGAN**

  https://colab.research.google.com/drive/19TfjXm2E_mdKDTVL0KwfDHslJ019ViMx

Massachusetts Institute of Technology

ALFA
ANYSCALE LEARNING FOR ALL

Thanks!

Comments?

JAMAL TOUTOUH
toutouh@mit.edu
jamal.es
necol.net
@jamtou