REMASKER : Imputation de données tabulaires avec Auto-Encodage Masqué[1]

Marcelin Seble - Nadia Garnou - Oussama Sahli

Introduction

Les valeurs manquantes sont omniprésentes dans les données tabulaires du monde réel en raison de diverses raisons lors de la collecte, du traitement, ou de la transmission de données. Bien que plusieurs méthodes d'imputation aient été proposées, elles sont souvent limitées en raison de leur nature discriminante ou généralisatrice.
Cet article présente une nouvelle méthode appelée REMASKER qui est basée sur un Transformer dont le méchanisme d'attention permet de capturer la corrélation complexe entre les données et qui est applicable à divers scénarios même avec un ratio de données manquantes élevé(ex: 0.7).

Formalisation

- **Données incomplètes** : Soit X un ensemble de variables, où chaque donnée $\mathbf{x} \in \mathcal{X}$ est représentée sous d dimensions:

$$\mathbf{x} \triangleq (\mathrm{x}_1, ..., \mathrm{x}_d) \subset \mathcal{X}_1 \text{ x...x } \mathcal{X}_d$$

On associe à chaque donnée $\mathbf{x}$ une variable $\mathbf{m}$ de dimension d indiquant les positions des valeurs manquantes :

$$\mathbf{m} \triangleq (\mathbf{m}_1, ..., \mathbf{m}_d) \in \{0, 1\}^d$$

Ainsi $\mathbf{x}_i$ n'est accessible que si $\mathbf{m}_i = 1$. Nous observons $\mathbf{x}$ sous sa forme incomplète :

$$\tilde{\mathbf{x}} \triangleq (\tilde{\mathrm{x}}_1, ..., \tilde{\mathrm{x}}_d)$$

$$\tilde{\mathrm{x}}_i \triangleq \begin{cases} \mathrm{x}_i \; si \; \mathrm{m}_i \; = 1 \\ * \; si \; \mathrm{m}_i \; = 0 \end{cases} (\mathrm{i} \in \{1, ..., d\})$$

Où * représente la valeur non observée.

- **Imputation** : Étant donné un ensemble de données incomplètes $\mathcal{D}$ tel que:

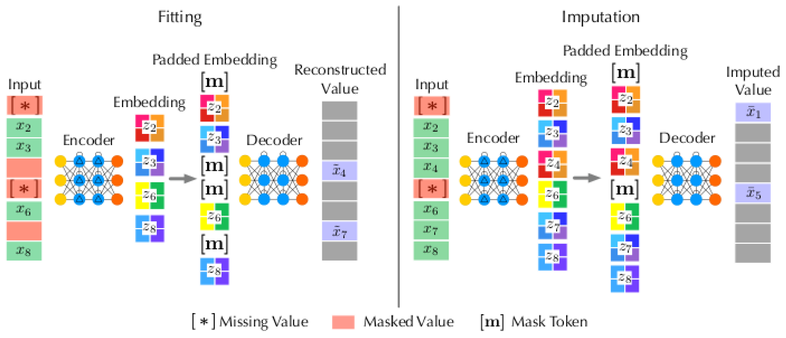
$$\mathcal{D} = \{\tilde{\mathbf{x}}^{(i)}, \mathbf{m}^{(i)}\}_{i=1}^{n}$$

Le but est de récupérer les valeurs manquantes de chaque entrée en générant une version imputée

$$\hat{\mathbf{x}} = (\hat{\mathrm{x}}_1, ..., \hat{\mathrm{x}}_d)$$

$$\text{Tel que : } \hat{\mathbf{x}} \triangleq \begin{cases} \tilde{\mathrm{x}}_i \; si \; \mathrm{m}_i \; = 1 \\ \bar{\mathrm{x}}_i \; si \; \mathrm{m}_i \; = 0 \end{cases} (\mathrm{i} \in \{1, ..., d\})$$

avec $\bar{\mathrm{x}}_i$ la valeur imputée



Architecture de REMASKER

Encodeur :

- **Embedding** :
  - Représentation par un vecteur de nombres réels
  - But : capturer des corrélations fines.
- **Positional Encoding** :
  - Mécanisme de récurrence pas pris en compte
  - Importance de la position et l'ordre des valeurs
- **Masquage aléatoire** :
  - Ratio de valeurs à remasquer.
  - Apprendre des représentations invariantes par rapport aux données manquantes
- **Encodage** :
  - **Standardisation** : Réduire la variance des données / Ordre de grandeurs différent
  - **MultiHeadSelfAttention** : Donner un poids selon la pertinence des états précédents par rapport au jeton courant / vanishing gradient
  - **MLP Layer**: Représentation latente / Couche linéaire, ReLU, Dropout, couhe linéaire

Décodeur :

- Padding sur les données remasquées
- Même schéma que l'encodage (standardisation + Attention + MLP)
- Prédiction finale: couche linéaire avec fonction d'activation sur les données.

Loss :

- RMSE : entre les données d'entrées et données de sorties du REMASKER.

Fonctionnement de REMASKER

Algorithme 1 REMASKER

Entrées: $\mathcal{D} = \{\tilde{\mathbf{x}}^{(i)}, \mathbf{m}^{(i)}\}_{i=1}^{n}$, MAX_EPOCH, Γ:fonction de remasquage, f_θ:encodeur, g_ϑ:decodeur,L: fonction de coût

Sorties: $\hat{\mathcal{D}} = \{\hat{\mathbf{x}}^{(i)}\}_{i=1}^{n}$: Données imputées

// Phase d'apprentissage (fitting)

tant que MAX_EPOCH *n'est pas atteint* **faire**

  pour tout $(\tilde{\mathbf{x}}, \mathbf{m}) \in \mathcal{D}$ **faire**

    $\tilde{\mathbf{x}}_{\mathbf{m} \wedge \bar{\mathbf{m}}'}, \tilde{\mathbf{x}}_{\mathbf{m} \wedge \mathbf{m}'} \leftarrow \Gamma(\tilde{\mathbf{x}}, \mathbf{m})$; // Remasquage

    $\mathbf{z} \leftarrow f_\theta(\tilde{\mathbf{x}}_{\mathbf{m} \wedge \mathbf{m}'})$; // Encodage des valeurs non masquées

    Rajouter les tokens des masques à z;

  fin pour tout

  // minimiser l'erreur de reconstruction

  Mettre à jour θ et ϑ avec $\nabla \mathsf{L}(g_\vartheta(\{\mathbf{z}\}), \{\tilde{\mathbf{x}}_{\mathbf{m} \wedge \bar{\mathbf{m}}'}\})$;

fin tant que

// Phase d'imputation

pour tout $(\tilde{\mathbf{x}}, \mathbf{m}) \in D$ **faire**

  $\mathbf{z} \leftarrow f_\theta(\tilde{\mathbf{x}}_{\mathbf{m}})$; // encodage des valeurs non manquantes

  Rajouter les tokens des masques à z;

  $\bar{\mathbf{x}} \leftarrow g_\vartheta(\mathbf{z})$; // prédiction des valeurs manquantes

  $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}_{\mathbf{m}} \cup \bar{\mathbf{x}}_{\bar{\mathbf{m}}}$;

fin pour tout

retourne $\hat{\mathcal{D}} = \{\hat{\mathbf{x}};\}$

$\tilde{\mathbf{x}}_{\bar{\mathbf{m}}}$, $\tilde{\mathbf{x}}_{\mathbf{m} \wedge \bar{\mathbf{m}}'}$ et $\tilde{\mathbf{x}}_{\mathbf{m} \wedge \mathbf{m}'}$ représentent respectivement les données masquée, remasquée et non masquée.

Résultats

Jeux de données utilisés :

1. **Iris** : Caractéristiques de fleurs, shape = (150,4)
2. **Obesity** : Caractéristiques physiques et comportementales d'un ensemble de personnes, shape = (2098,16)
3. **Climate** : Données climatiques et météorologiques, shape = (527,20)
4. **bike** : Données sur la location de vélos avec leurs caractéristiques, shape = (8747,12)

Méthode	MAR	MCAR	MNAR
Remasker	0.22	0.27	0.25
Lstm	0.48	0.53	0.54
MissForest	0.23	0.26	0.25
MostFrequent	0.34	0.36	0.36
Mean	0.25	0.28	0.27

Table 1:RMSE - Dataset Obesity (missing ratio=0.3)

Temps	Iris	Climate	Obesity	Bike
Remasker	2	5	15	45
Total	20	120	240	1200

Table 2:Temps d'exécution en min

Conclusion

- Article assez compréhensible.
- Première méthode d'imputation qui étend les MAE.
- Résultats qui ne surpassent pas l'état de l'art.

Références

[1] Tianyu Du, Luca Melis, and Ting Wang. "Have Missing Data? Make It Miss More! Imputing Tabular Data with Masked Autoencoding". In: (2023). URL: https://openreview.net/forum?id=yzE6LtZSHo.