

Algoritmo multi-fuente de imputación de datos faltantes basado en algoritmo EM y vecinos recomendados

19 de junio de 2025

Productos publicados en el marco de esta trabajo

- ① Campos, S., Pizarro, L., Valle, C., Gray, K. R., Rueckert, D., and Allende, H. (2015). Evaluating imputation techniques for missing data in ADNI: A patient classification study. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 3–10. Springer International Publishing.
- ② Campos, S., Veloz, A., and Allende, H. (2018). An out of sample version of the EM algorithm for imputing missing values in classification. volume 11401 of *Lecture Notes in Computer Science*, pages 194–202. Springer.
- ③ Campos, S., Zamora, J., and Allende, H. (2024). Block-wise imputation EM algorithm in multi-source scenario: Adni case. *Pattern Anal. Appl.*, 27(2).

Table of Contents

- ① El problema
- ② Estado del arte
- ③ Marco teórico
- ④ Propuesta
- ⑤ Experimentos
- ⑥ Conclusiones y trabajo futuro

Tabla de contenidos

1 El problema

2 Estado del arte

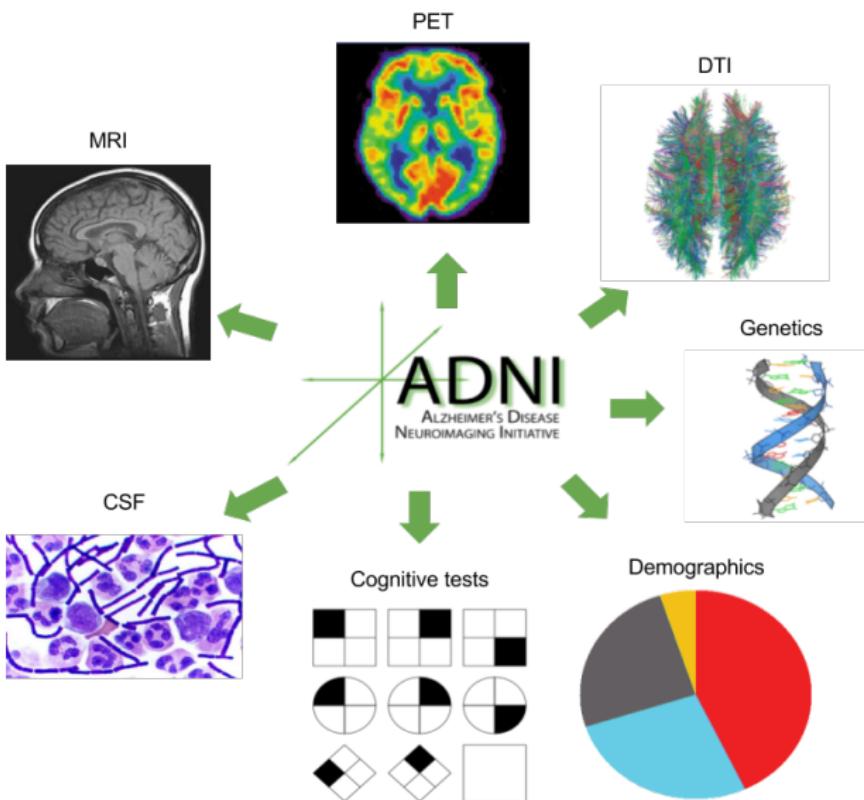
3 Marco teórico

4 Propuesta

5 Experimentos

6 Conclusiones y trabajo futuro

Motivación: problema inicial



Motivación: problema inicial

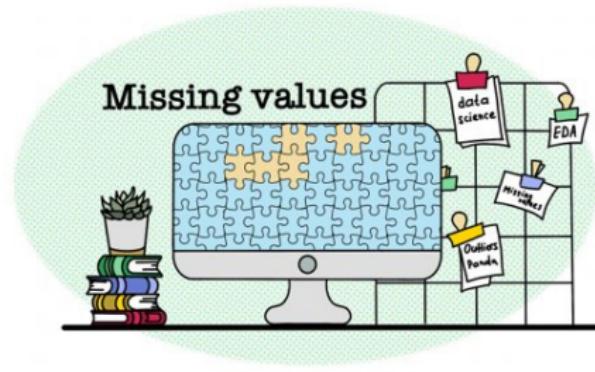
ADNI (Alzheimers Disease Neuroimaging Initiative)¹.

	Fuente 1		Fuente r		Fuente s	
	variable 1	variable 2	variable j	variable d-1	variable d	
Ejemplo 1	[Red]	[Red]		[Blue]	[Blue]	[?]
Ejemplo 2	[?]	[?]		[?]	[?]	[Blue]
Ejemplo 3	[Red]	[Red]		[Blue]	[Blue]	[Blue]
⋮						
Ejemplo i	[Red]	[Red]		[?]	[?]	[?]
⋮						
Ejemplo n-1	[Red]	[Red]		[Blue]	[Blue]	[Blue]
Ejemplo n	[?]	[?]		[Blue]	[Blue]	[Blue]

¹<http://www.adni-info.org>

Motivación: problema general

- Datos faltantes (Missing Values) se refiere a la ausencia de valores dentro de un conjunto de datos.
- Problema común en el análisis de datos y en Machine Learning.



- La **calidad y la completitud** de los datos son **fundamentales** para obtener **resultados precisos y confiables**.
- Ya que muchos **algoritmos de aprendizaje automático no soportan datos con valores faltantes**, es importante abordar este problema.

Motivación: problema general

Los Missing Values (MV) pueden generarse por razones como:



- **Falta de recursos:** dinero, tiempo o máquinas (observatorios, etc.)
- **Fallas en capturar datos** por efectos ambientales o fallas de hardware (redes de sensores).
- Prohibición por **protocolos** (hospitales).
- **No contestar una encuesta de satisfacción.**

Datos Faltantes

- Un ejemplo i en un problema de clasificación esta representado por $x_{i:} = [x_{i1}, x_{i2}, \dots, x_{ij}, \dots x_{id} | c_k]$ donde d es el número total de variables (o características) y c_k es la k -ésima clase (o etiqueta) de c clases en total.
- Cuando se trata con datos faltantes, un set de datos D esta compuesto por:

$$D = \{X, C, M\} \quad (1)$$

donde $C = [c_1, c_2, \dots, c_c]$ es el conjunto de etiquetas y $m_{i:} = [m_{i1}, m_{i2}, \dots m_{id}]$ indica **cuales variables del vector i son desconocidas**.

- X , para el caso de datos faltantes, es dividida en $X = \{X_o, X_m\}$ donde X_o son los datos observables y X_m son los datos faltantes.

Mecanismos de datos Faltantes

El mecanismo de datos faltantes esta caracterizado por la **distribución condicional de M dada X** ,

$$p[M | X, \xi] = p[M | X_o, X_m, \xi] \quad (2)$$

donde ξ denota los **parámetros desconocidos que definen el mecanismo de datos faltantes**. Little and Rubin definen 3 tipos de mecanismos de datos faltantes:

Proceso de generación de datos faltantes (Taxonomía de Little y Rubin, 1987, 2019 [1])

- MAR (Missing At Random)
- MCAR (Missing Completely At Random)
- MNAR (Missing Not At Random)

Ubicacion de los datos faltantes

- Datos faltantes solo en los datos de entrenamiento.
- Datos faltantes solo en los datos de prueba.
- **Datos faltantes en los datos de entrenamiento y en los datos de prueba.**

El problema
oooooooo

Estado del arte
●○

Marco teórico
ooooo

Propuesta
oooooooooooo

Experimentos
oooooooooooooooooooo

Conclusiones y trabajo futuro
oooo

Tabla de contenidos

1 El problema

2 Estado del arte

3 Marco teórico

4 Propuesta

5 Experimentos

6 Conclusiones y trabajo futuro

Estado del arte

Los algoritmos de imputación los dividiremos en 3 categorías:

① Basados en análisis estadístico:

- Media, mediana, moda [1]
- Multiple imputation [2]
- Multivariate Imputation by Chained-Equations (MICE) [3]

② Basados en factorización de matrices:

- Singular Value Decomposition (SVD) [4]
- Singular Value Thresholding (SVT) [5]

③ Basados en Machine Learning:

- KNN [6]
- Random Forest (MissForest) [7]
- Técnicas de clustering [8]
- Deep Learning [9] [10]

El problema
oooooooo

Estado del arte
oo

Marco teórico
●oooo

Propuesta
oooooooooooo

Experimentos
oooooooooooooooooooo

Conclusiones y trabajo futuro
oooo

Tabla de contenidos

- 1 El problema
- 2 Estado del arte
- 3 Marco teórico
- 4 Propuesta
- 5 Experimentos
- 6 Conclusiones y trabajo futuro

Algoritmo EM (Expectation Maximization)

- Algoritmo iterativo para la estimación de parámetros mediante la **maximización de la verosimilitud**.
- Formalmente, los pasos se pueden expresar como:

$$\text{Paso } E : Q(\theta_t) = E[I(\theta|X, z)]$$

$$\text{Paso } M : \theta_{t+1} = \arg \max_{\theta} Q(\theta_t)$$

- donde θ_t es el **vector de parámetros** en la iteración t .
- X es la matriz de datos.
- $I(\cdot)$ la **función de log-verosimilitud**.
- z contiene la información de los MV.

Algoritmo EM regularizado (EMreg)

Un trabajo muy importante para esta tesis es el trabajo de **Tapio Schneider** [11].

- ① Schneider propone un algoritmo **EM regularizado**, basado en EM.
- ② Comienza **estimando la media y matriz de covarianza**.
- ③ Con esto se realiza **imputación de los MV** a través de los pasos E y M.
- ④ Vuelve al paso 2.

Algoritmo EM regularizado (EMreg)

La imputación viene dada por un modelo de regresión lineal:

$$x_m = \mu_m + (x_o - \mu_o)\beta + e \quad (3)$$

- $x_o \in \mathbb{R}^{1 \times p_o}$ vector de datos observados.
- $x_m \in \mathbb{R}^{1 \times p_m}$ vector a imputar.
- $\mu_o \in \mathbb{R}^{1 \times p_o}$ vector de medias de variables con datos observados.
- $\mu_m \in \mathbb{R}^{1 \times p_m}$ vector de medias de variables con MV.
- $\beta \in \mathbb{R}^{p_o \times p_m}$ matriz de coeficientes de regresión.

$e \in \mathbb{R}^{1 \times p_m}$ es un **vector aleatorio con media cero y matriz de covarianza** $C \in \mathbb{R}^{p_m \times p_m}$ desconocida.

Algoritmo EM regularizado (EMreg)

EMreg se basa en el algoritmo EM con la diferencia de que la matriz $\widehat{\Sigma}_{oo}^{-1}$ estimada para calcular β se reemplaza por:

$$\widehat{\Sigma}_{oo}^{-1} \leftarrow (\widehat{\Sigma}_{oo} + h^2 \widehat{D})^{-1} \quad (4)$$

- $\widehat{D} = \text{Diag}(\widehat{\Sigma}_{oo})$
- h parámetro de regularización.
- Busca solucionar el problema de cuando la **dimensión de los datos es mayor al número de datos**.
- Evita que la matriz $\widehat{\Sigma}$ sea singular.

Tabla de contenidos

1 El problema

2 Estado del arte

3 Marco teórico

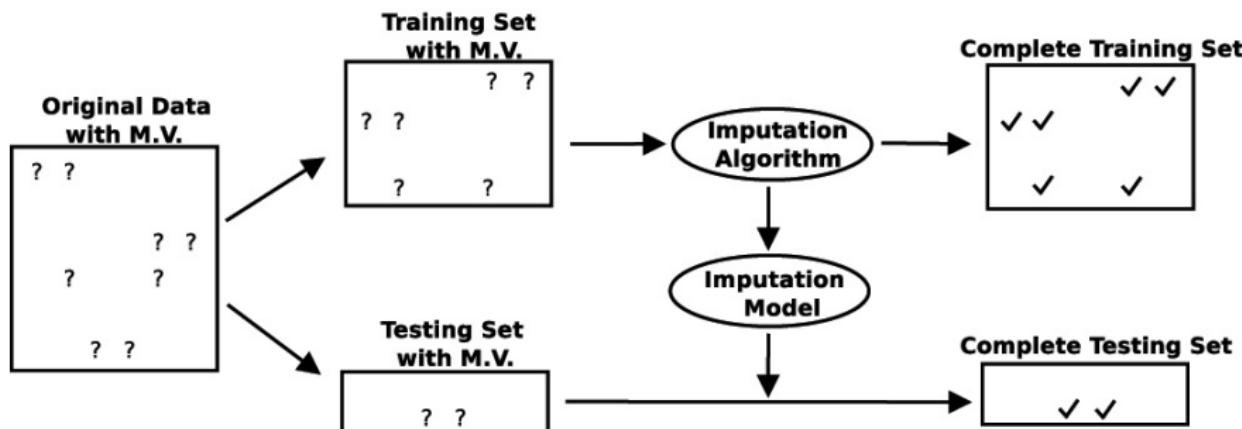
4 Propuesta

5 Experimentos

6 Conclusiones y trabajo futuro

Metodología en aprendizaje supervisado

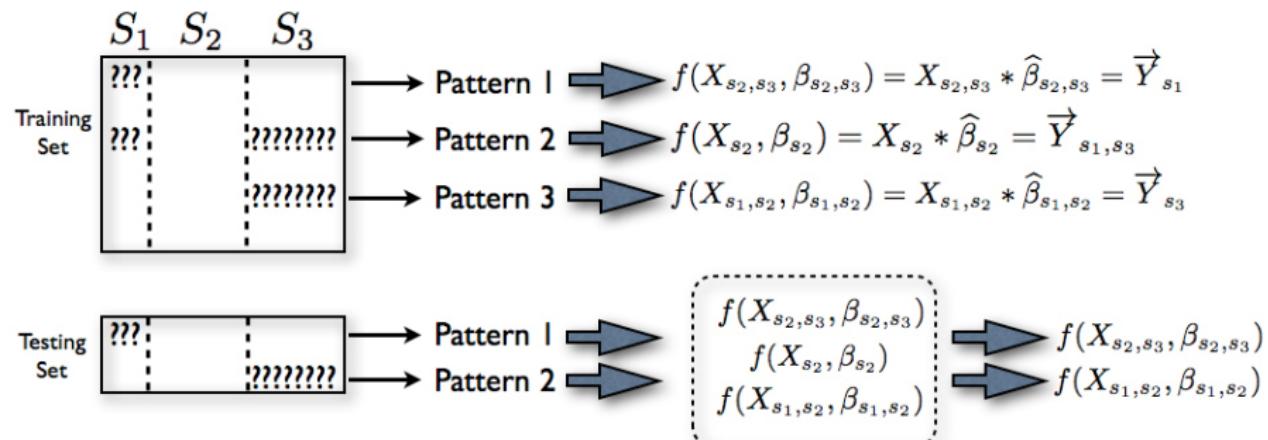
- El algoritmo EMreg trabaja separadamente sobre datos de entrenamiento (training set) y prueba (testing set).



El algoritmo EMreg no permite crear un modelo de imputación.

Propuesta 1: EMreg Out-of-Sample ($EMreg_{oos}$)

Se presenta una nueva versión del algoritmo EM, basado en la propuesta de Schneider.



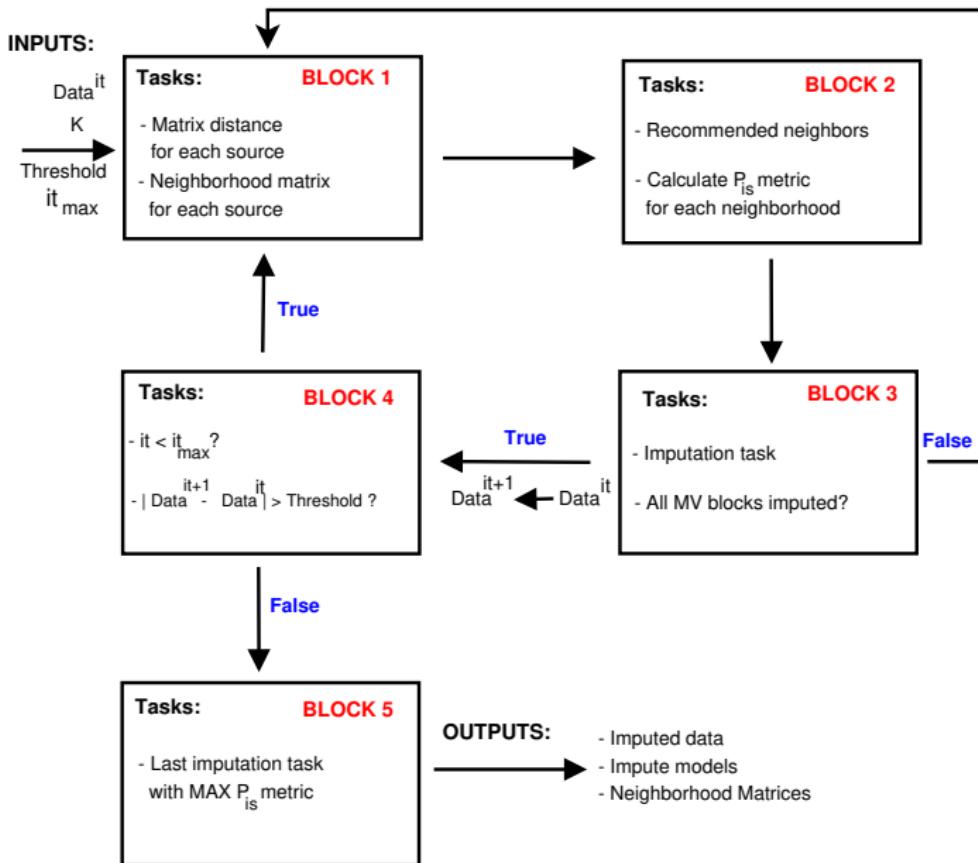
El algoritmo crea un modelo general que consta de tantos modelos de regresión como patrones de MV existen en el training set.

Propuesta 2: EMreg-knn (con vecinos recomendados)

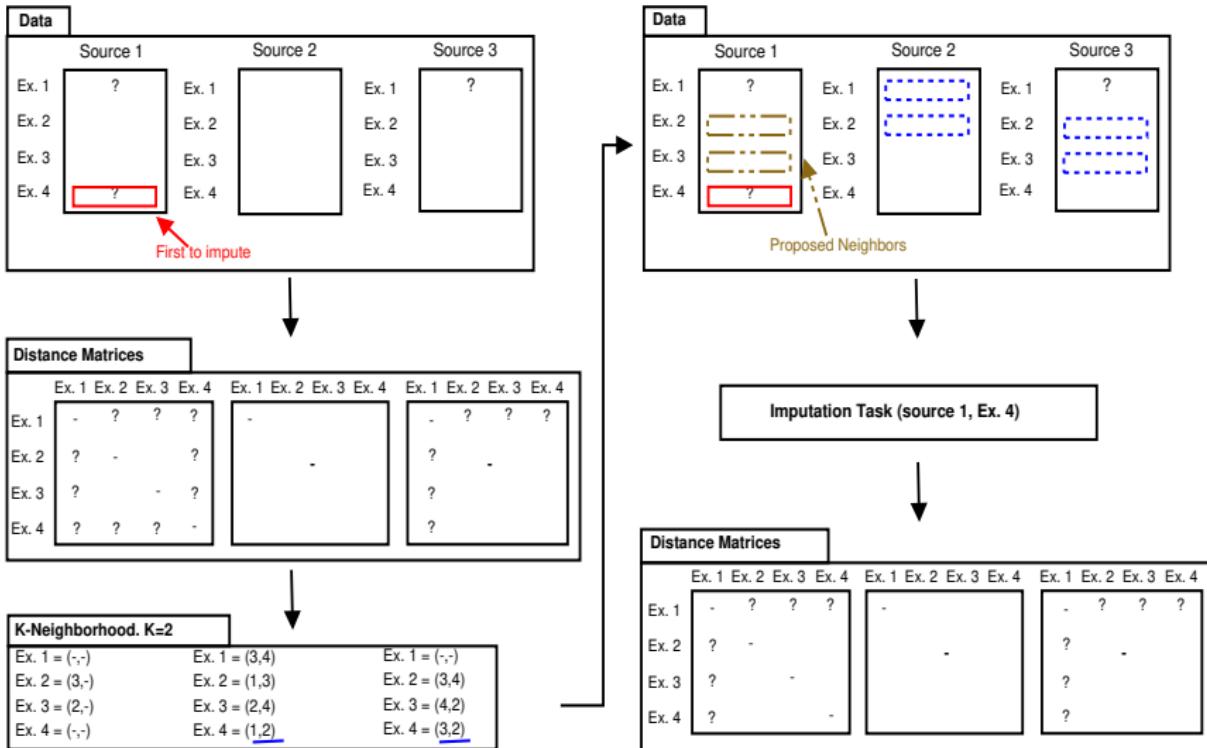
Versión actualizada del algoritmo EMreg-oos que incorpora varios aspectos adicionales que buscan **mejorar la clasificación posterior al proceso de imputación**.

- **Se crean vecindarios** por cada dato y por cada fuente.
- Las fuentes comunican información mediante **recomendación de vecinos**.
- Se **utilizan las etiquetas** de los datos en la etapa del entrenamiento.
- Los datos de test se imputan con un **ensamblado de regresiones**.

EMreg-knn: dividido en 5 bloques



EMreg-knn: Block 1-2, ejemplo de vecinos recomendados



EMreg-knn: Block 3, tarea de imputación

- Con los vecindarios creados, se realiza la imputación.
- Por cada bloque de MV, se construirá una **matriz de datos que incluya todas las fuentes** del dato a imputar y su respectivo vecindario.
- Esto busca construir un modelo de imputación con solo los datos del vecindario.
- Se **imputa siguiendo la ecuación 3** donde es necesario almacenar los parámetros del modelo μ y β (versión Out-of-Sample)
- Se tendrán **tantos modelos de imputación como bloques de MV existan.**

EMreg-knn: Block 5

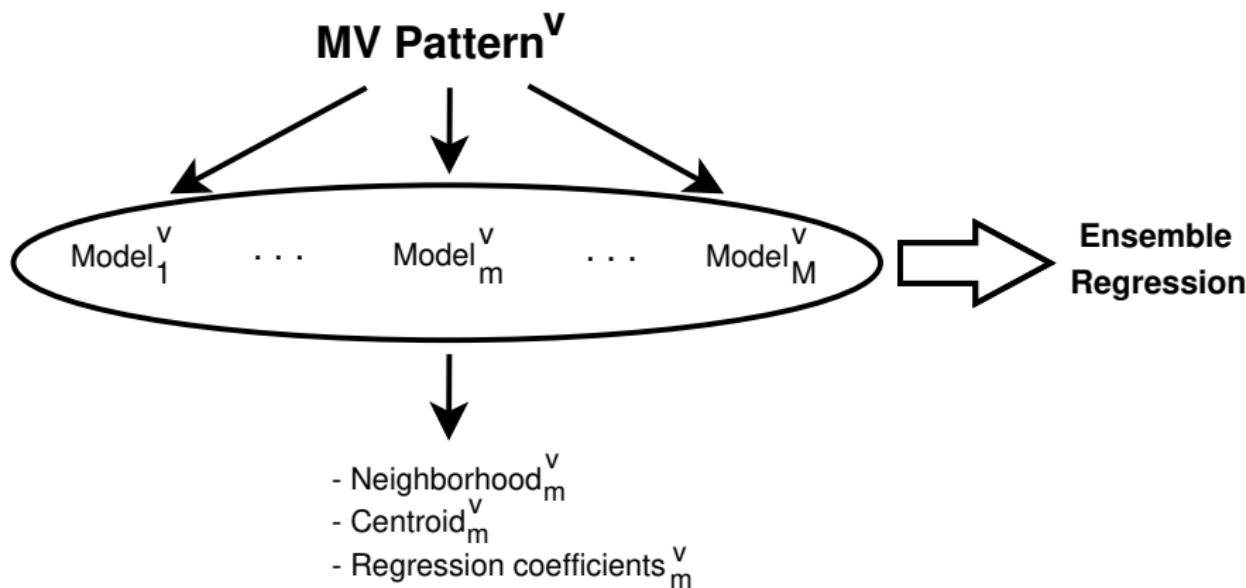
- Se realiza una **última imputación** de todos los bloques de MV.
- Se consideran los **vecindarios con el $\pi(i, s)$ mayor**.
- $0 \leq \pi(i, s) \leq 1$ que entrega la pureza del vecindario.

Finalmente, se debe **calcular el modelo de imputación** que permitirá realizar las imputaciones en el dataset de prueba, es decir, el modelo **Out-of-sample**.

EMreg-knn: imputación del testing set

- Se debe **identificar el tipo de patrón** de MV.
- Por cada patrón de MV v , **existen M modelos de imputación, asociados a cada uno de los ejemplos con ese patrón** de MV durante el entrenamiento.
- Esos modelos se usan para **crear un Ensamblado** que permite imputar el dato de test.
- Cada uno de los modelos depende de su vecindario, el cual tiene un **vector representante (centroide)**.

EMreg-knn: imputación del testing set



EMreg-knn: imputación del testing set

- El centroide es un vector representativo del vecindario (vector de medias).
- Este vector r_m^v es el vector representante del modelo m con el patrón de MV v y **se utilizará para calcular la distancia entre este vector y el vector de prueba x_i^{test}** que se debe imputar.

El cálculo de la distancia es realizada segun la ecuación 5:

$$d_{mi}^v = D_*(r_m^v, x_i^{test}) \quad (5)$$

donde $D_*(\cdot)$ es una función de distancia aplicada solo a las variables con datos observados.

EMreg-knn: imputación del testing set

La imputación final del dato de test considerando que el patrón de MV v tiene M modelos de imputación es dado por la ecuación 6:

$$x_i = \sum_{m=1}^M W_{mi}^v \cdot f_m^v(X, \beta) \quad (6)$$

$$W_{mi}^v = \frac{(1 - H_{mi}^v)}{M-1} \qquad \qquad H_{mi}^v = \frac{d_{mi}^v}{\sum_{m=1}^M d_{mi}^v}$$

Tabla de contenidos

1 El problema

2 Estado del arte

3 Marco teórico

4 Propuesta

5 Experimentos

6 Conclusiones y trabajo futuro

Estandarización de los datos

- Por cada set de datos, se realizó un proceso de estandarización a los datos de entrenamiento y prueba.
- Siguiendo la estrategia Out-of-Sample, los datos de prueba se estandarizaron segun la ecuación 7:

$$X_{std,j}^{test} = \frac{X_{raw,j}^{test} - \mu_j^{train}}{\sigma_j^{train}} \quad (7)$$

Algoritmos a competir

- ① MissForest [7]: utilizando una implementación en python (missingpy ²).
- ② MICE [3]: utilizando una implementación en python.
- ③ SVD [4]: utilizando una implementación en python del paquete *fancyimpute*. No crea modelo de imputación.
- ④ EMreg [11]: implementación original. No crea modelo de imputación.

²<https://github.com/epsilon-machine/missingpy>

Clasificadores usados

En cada experimento, se dividió el dataset en 75 % para entrenamiento y 25 % para prueba.

Los clasificadores y la sintonización de hiperparámetros fue:

- ① K-NN: número de vecinos y distancia.
- ② ν -SVM: ν en un rango $(0, 1]$ y *kernel*.
- ③ RF: el número de árboles y el número de variables.
- ④ ANN: dropout. 2 capas ocultas, cantidad de neuronas igual a $n_var \cdot 1.2$, función de activación *Relu*, algoritmo de optimización *Adam* y número de epochas 10.

ADNI

- La primera columna muestra el número de datos por cada clase.
- Las otras columnas muestra la cantidad de datos con MV por fuente de información.

	ADNI	CSF	MRI	PET
AD	185	85	0	114
HC	210	107	0	141
pMCI	164	80	0	102
sMCI	217	114	0	132
Total	776	386	0	489

- CSF contiene 3 variables.
- MRI y PET tienen 83 variables cada una.

ADNI

Se considerarán 4 problemas de clasificación, los cuales se dividen segun las clases involucradas.

Problema	#Instancias	#Instancias sin MV (# Instancias por clase)
(a) AD vs HC	395	72 (37 AD - 35 HC)
(b) MCI vs HC	591	110 (75 MCI - 35 HC)
(c) pMCI vs sMCI	381	75 (34 pMCI - 41 sMCI)
(d) AD vs HC vs MCI	776	147 (37 AD - 35 HC - 75 MCI)

Las metricas obtenidas (media y desviación estandar) son el resultado de 50 corridas.

ADNI: AD vs HC

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
K-NN	<i>none</i>	80.6(8.6)	92.1(6.8)	91.1(9.2)	72.2(13.9)	81.3(9.1)
	MICE	73.7(5.8)	83.5(5.7)	86.8(6.1)	58.4(10.9)	78.1(4.9)
	SVDimpute	84.1(3.6)	92.4(3.0)	95.0(2.6)	71.4(7.6)	86.6(3.2)
	MissForest	83.4(4.1)	91.9(3.1)	93.8(3.2)	72.2(8.0)	86.3(3.6)
	EMreg	83.3(4.3)	91.3(3.8)	94.5(3.5)	70.1(9.1)	86.0(3.7)
	EMreg-oos	84.4(3.8)	91.4(3.8)	94.3(3.1)	72.8(7.0)	86.7(3.3)
	EMreg-KNN	85.7(3.3)*	92.1(2.9)	92.8(3.3)	77.5(6.6)*	87.5(3.0)*
	<i>none</i>	85.4(9.2)	93.7(5.8)	84.1(14.3)	87.0(11.8)	83.8(11.7)
	MICE	78.7(3.4)	89.4(2.7)	78.6(8.9)	79.5(12.0)	79.9(3.3)
SVM	SVDimpute	88.1(3.4)	94.1(2.4)	90.8(4.2)	85.2(5.7)	89.2(3.2)
	MissForest	87.9(3.0)	93.9(2.4)	89.9(4.3)	85.8(6.2)	88.9(2.9)
	EMreg	88.2(3.3)	94.0(2.4)	91.5(4.4)	84.5(6.3)	89.3(3.2)
	EMreg-oos	88.6(3.0)	94.1(2.3)	91.9(3.1)	85.0(5.6)	89.7(2.9)
	EMreg-KNN	89.1(2.6)	94.5(2.2)	90.5(3.2)	87.7(5.6)*	90.0(2.4)

ADNI: AD vs HC

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
RF	<i>none</i>	82.7(8.9)	91.9(6.5)	84.6(13.4)	82.2(12.7)	81.8(10.0)
	MICE	86.2(3.3)	92.9(2.7)	87.3(4.3)	85.0(5.9)	87.3(3.1)
	SVDimpute	86.3(3.6)	93.3(2.5)	87.9(5.2)	84.7(5.6)	87.4(3.3)
	MissForest	83.3(4.5)	92.0(2.7)	81.5(7.9)	85.8(8.6)	84.0(4.4)
	EMreg	84.6(3.8)	92.9(2.6)	87.2(7.5)	81.5(10.2)	85.9(3.8)
	EMreg-oos	86.8(3.2)*	93.5(2.4)	88.0(4.5)	85.7(6.4)	87.8(2.9)*
ANN	EMreg-KNN	86.5(3.8)	93.2(2.4)	86.4(5.0)	86.9(5.5)*	87.4(3.5)
	<i>none</i>	84.9(8.6)	94.0(6.4)	84.9(11.7)	85.4(12.1)	83.7(9.5)
	MICE	79.1(3.8)	78.8(4.4)	81.6(5.0)	76.3(7.4)	80.8(3.9)
	SVDimpute	87.5(3.5)	92.7(2.7)	89.9(3.9)*	85.0(6.6)	88.6(3.4)
	MissForest	86.7(3.9)	92.8(2.8)	87.0(6.5)	86.8(6.3)	87.6(4.0)
	EMreg	85.7(4.1)	91.7(3.1)	87.1(7.9)	84.2(6.6)	86.7(4.3)
	EMreg-oos	87.6(3.3)	92.6(2.5)	89.5(4.7)	85.4(5.9)	88.6(3.3)
	EMreg-KNN	88.1(2.9)*	93.7(2.2)*	89.3(4.0)	86.9(5.3)*	89.1(2.7)*

ADNI: AD vs HC vs MCI

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	F (%)
K-NN	<i>none</i>	50.7(9.7)	53.2(6.5)	50.7(9.7)	42.5(11.0)
	MICE	53.2(3.4)	58.1(2.3)	53.2(3.4)	47.5(4.3)
	SVDimpute	54.0(3.2)	59.1(2.3)	54.0(3.2)	48.6(4.1)
	MissForest	53.3(3.8)	58.5(2.7)	53.3(3.8)	48.1(4.8)
	EMreg	53.0(3.4)	57.8(2.6)	53.0(3.4)	47.2(4.6)
	EMreg-oos	53.9(3.5)	59.2(2.5)	53.9(3.5)	48.8(4.4)
	EMreg-KNN	54.5(3.3)*	59.7(2.6)*	54.5(3.3)*	49.9(4.6)*
SVM	<i>none</i>	53.1(10.5)	59.0(8.4)	53.1(10.5)	50.7(11.8)
	MICE	54.0(3.7)	61.0(3.2)	54.0(3.7)	52.9(4.5)
	SVDimpute	54.7(3.8)	61.5(3.2)	54.7(3.8)	53.7(4.3)
	MissForest	54.5(4.0)	60.9(3.2)	54.5(4.0)	53.1(4.3)
	EMreg	54.8(4.0)	60.8(3.1)	54.8(4.0)	52.9(4.4)
	EMreg-oos	55.5(4.0)	62.3(3.1)	55.5(4.0)	54.6(4.2)
	EMreg-KNN	56.5(3.3)*	63.3(2.8)*	56.5(3.3)*	55.8(3.5)*

ADNI: AD vs HC vs MCI

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	F (%)
RF	<i>none</i>	55.5(8.5)	60.6(7.1)	55.5(8.5)	53.9(8.7)
	MICE	57.2(3.5)	63.1(2.9)	57.2(3.5)	56.0(3.6)
	SVDimpute	57.4(3.7)	62.9(3.1)	57.4(3.7)	56.0(3.9)
	MissForest	56.5(3.6)	61.7(2.9)	56.5(3.6)	54.6(4.2)
	EMreg	55.8(3.7)	60.2(3.1)	55.8(3.7)	52.5(4.6)
	EMreg-oos	57.3(3.5)	63.4(2.8)	57.3(3.5)	56.5(3.5)
	EMreg-KNN	58.0(3.8)*	64.4(3.1)*	58.0(3.8)*	57.5(3.9)*
ANN	<i>none</i>	52.5(9.1)	58.4(7.0)	52.5(9.1)	49.1(10.9)
	MICE	55.0(3.0)	62.2(2.8)	55.0(3.0)	54.2(3.5)
	SVDimpute	56.2(3.1)	63.0(3.3)	56.2(3.1)	55.4(3.8)
	MissForest	56.0(3.4)	62.7(3.6)	56.0(3.4)	55.0(4.4)
	EMreg	55.1(3.4)	61.4(3.2)	55.1(3.4)	53.7(4.7)
	EMreg-oos	55.5(2.9)	62.6(2.5)	55.5(2.9)	55.0(3.0)
	EMreg-KNN	56.3(3.1)	63.7(2.4)*	56.3(3.1)*	55.8(3.1)*

Handwritten digits

- Handwritten³ es un data set clásico de dígitos manuscritos.
- 2000 imágenes, 200 por cada dígito.
- Las variables se dividen en 6 fuentes, cada una con el siguiente número de variables :
 - 76 coeficientes de fourier.
 - 216 correlaciones de perfil (profile correlations).
 - 64 coeficiente de Karhunen-Love.
 - 240 promedio de pixeles de ventanas de 2×3 .
 - 47 momentos de Zernike.
 - 6 características morfológicas.

³<https://archive.ics.uci.edu/ml/datasets/Multiple+Features>

Handwritten digits

- Se tomaron tres fuentes: Coeficientes de Fourier, coeficientes de Karhunen-Love y momentos de Zernike, obteniendo un total de 187 variables.
- No existen datos faltantes, por lo tanto se crean patrones de MV en diferentes porcentajes: 20 %, 40 % y 60 %.
- Aleatoriamente, se elijen los ejemplos que serán afectados por MV. Una vez que es elegido un ejemplo, aleatoriamente se elige el patrón de MV que se le aplicará.

Las metricas obtenidas (media y desviación estandar) son el resultado de 30 corridas.

Handwritten digits

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
K-NN	<i>none</i>	0	97.2(0.7)	98.5(0.4)	97.2(0.7)	97.2(0.7)
	MICE	20	93.6(1.1)	96.4(0.6)	93.6(1.1)	93.5(1.0)
	SVDimpute	20	92.7(1.3)	95.9(0.7)	92.7(1.3)	92.7(1.2)
	MissForest	20	92.2(1.3)	95.7(0.7)	92.2(1.3)	92.2(1.2)
	EMreg	20	93.5(1.1)	96.4(0.6)	93.5(1.1)	93.5(1.1)
	EMreg-oos	20	94.0(1.0)	96.6(0.6)	94.0(1.0)	93.9(1.0)
	EMreg-KNN	20	94.0(1.0)	96.7(0.6)	94.0(1.0)	94.0(1.0)
	MICE	40	91.1(1.2)	95.0(0.7)	91.1(1.2)	91.0(1.2)
	SVDimpute	40	89.5(1.5)	94.2(0.9)	89.5(1.5)	89.4(1.5)
	MissForest	40	88.5(1.5)	93.6(0.8)	88.5(1.5)	88.6(1.5)
	EMreg	40	90.8(1.2)	94.9(0.7)	90.8(1.2)	90.8(1.2)
	EMreg-oos	40	91.4(1.3)	95.2(0.7)	91.4(1.3)	91.3(1.3)
	EMreg-KNN	40	91.6(1.2)*	95.3(0.7)*	91.6(1.2)	91.5(1.2)*
	MICE	60	89.2(1.7)	94.0(1.0)	89.2(1.7)	89.2(1.7)
	SVDimpute	60	87.3(1.7)	92.9(0.9)	87.3(1.7)	87.2(1.7)
	MissForest	60	86.2(1.8)	92.3(1.0)	86.2(1.8)	86.2(1.8)
	EMreg	60	88.3(1.3)	93.5(0.7)	88.3(1.3)	88.3(1.3)
	EMreg-oos	60	89.5(1.3)	94.2(0.7)	89.5(1.3)	89.4(1.4)
	EMreg-KNN	60	90.3(1.5)*	94.6(0.9)*	90.3(1.5)	90.2(1.6)*

Handwritten digits

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
	<i>none</i>	0	98.3(0.5)	99.1(0.3)	98.3(0.5)	98.3(0.5)
	MICE	20	96.5(0.8)	98.0(0.4)	96.5(0.8)	96.5(0.8)
	SVDimpute	20	96.2(0.8)	97.9(0.5)	96.2(0.8)	96.2(0.8)
	MissForest	20	95.6(1.0)	97.6(1.0)	95.6(1.0)	95.6(1.0)
	EMreg	20	96.4(0.9)	98.0(0.5)	96.4(0.9)	96.4(0.9)
	EMreg-oos	20	96.6(0.8)	98.1(0.4)	96.6(0.8)	96.6(0.8)
	EMreg-KNN	20	96.7(0.8)	98.2(0.4)	96.7(0.8)	96.7(0.8)
SVM	MICE	40	94.4(1.1)	96.9(0.6)	94.4(1.1)	94.3(1.1)
	SVDimpute	40	94.2(1.2)	96.8(0.7)	94.2(1.2)	94.2(1.2)
	MissForest	40	92.7(1.4)	96.0(0.8)	92.7(1.4)	92.8(1.4)
	EMreg	40	94.0(1.3)	96.7(0.7)	94.0(1.3)	94.0(1.3)
	EMreg-oos	40	94.5(1.2)	97.0(0.7)	94.5(1.2)	94.5(1.2)
	EMreg-KNN	40	94.8(1.0)*	97.1(0.6)*	94.8(1.0)	94.8(1.0)*
	MICE	60	92.6(1.1)	95.9(0.6)	92.6(1.1)	92.6(1.1)
	SVDimpute	60	92.3(1.2)	95.7(0.6)	92.3(1.2)	92.2(1.2)
	MissForest	60	90.6(1.5)	94.8(0.8)	90.6(1.5)	90.6(1.4)
	EMreg	60	92.3(1.0)	95.7(0.6)	92.3(1.0)	92.2(1.0)
	EMreg-oos	60	93.0(0.8)*	96.1(0.4)*	93.0(0.8)	93.0(0.8)*
	EMreg-KNN	60	92.9(0.8)	96.1(0.5)	92.9(0.9)	92.9(0.9)

Handwritten digits

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
RF	<i>none</i>	0	97.7(0.7)	98.7(0.4)	97.7(0.7)	97.7(0.7)
	MICE	20	95.5(1.1)	97.5(0.6)	95.5(1.1)	95.5(1.1)
	SVDimpute	20	95.4(1.1)	97.4(0.6)	95.4(1.1)	95.4(1.1)
	MissForest	20	94.4(1.0)	96.9(0.6)	94.4(1.0)	94.4(1.0)
	EMreg	20	95.4(0.9)	97.4(0.5)	95.4(0.9)	95.4(0.9)
	EMreg-oos	20	95.8(1.0)	97.7(0.6)	95.8(1.0)	95.8(1.0)
	EMreg-KNN	20	96.0(0.9)*	97.8(0.5)*	96.0(0.9)	96.0(0.9)*
	MICE	40	93.2(1.4)	96.3(0.8)	93.2(1.4)	93.2(1.4)
	SVDimpute	40	92.9(1.3)	96.1(0.7)	92.9(1.3)	92.9(1.3)
	MissForest	40	91.0(1.2)	95.0(0.7)	91.0(1.2)	91.0(1.2)
	EMreg	40	93.0(1.2)	96.1(0.7)	93.0(1.2)	92.9(1.2)
	EMreg-oos	40	93.5(1.2)	96.4(0.6)	93.5(1.2)	93.5(1.2)
	EMreg-KNN	40	93.7(1.0)*	96.5(0.6)*	93.7(1.0)	93.7(1.0)*
	MICE	60	91.2(1.2)	95.1(0.6)	91.2(1.2)	91.2(1.7)
	SVDimpute	60	91.4(1.3)	95.2(0.7)	91.4(1.3)	91.4(1.3)
	MissForest	60	88.2(1.6)	93.4(0.9)	88.2(1.6)	88.2(1.6)
	EMreg	60	90.9(1.4)	94.9(0.8)	90.9(1.4)	90.9(1.4)
	EMreg-oos	60	91.9(1.1)*	95.5(0.6)*	91.9(1.1)	91.9(1.1)*
	EMreg-KNN	60	91.9(1.0)*	95.5(0.6)*	91.9(1.0)	91.9(1.0)*

Handwritten digits

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
ANN	<i>none</i>	0	97.7(0.6)	98.7(0.3)	97.7(0.6)	97.7(0.6)
	MICE	20	95.6(0.9)	97.6(0.5)	95.6(0.9)	95.6(1.0)
	SVDimpute	20	95.6(1.0)	97.6(0.6)	95.6(1.0)	95.6(1.0)
	MissForest	20	94.8(1.0)	97.1(0.5)	94.8(1.0)	94.8(1.0)
	EMreg	20	95.6(0.9)	97.5(0.5)	95.6(0.9)	95.6(0.9)
	EMreg-oos	20	96.0(0.9)	97.8(0.5)	96.0(0.9)	96.0(0.9)*
	EMreg-KNN	20	96.0(0.8)*	97.8(0.5)*	96.0(0.8)	96.0(0.9)*
	MICE	40	94.7(0.8)	97.0(0.5)	94.7(0.8)	94.7(0.8)
	SVDimpute	40	93.4(1.4)	96.3(0.8)	93.4(1.4)	93.4(1.4)
	MissForest	40	91.8(1.3)	95.4(0.7)	91.8(1.3)	91.8(1.3)
	EMreg	40	93.5(1.3)	96.4(0.7)	93.5(1.3)	93.4(1.3)
	EMreg-oos	40	93.9(1.2)	96.6(0.7)	93.9(1.2)	93.4(1.2)
	EMreg-KNN	40	94.1(1.1)	96.7(0.6)	94.1(1.1)	94.1(1.1)*
	MICE	60	92.0(1.0)	95.6(0.6)	92.0(1.0)	92.0(1.0)
	SVDimpute	60	91.6(1.1)	95.3(0.6)	91.6(1.1)	91.6(1.1)
	MissForest	60	89.3(1.6)	94.1(0.9)	89.3(1.6)	89.3(1.6)
	EMreg	60	91.1(1.1)	95.1(0.6)	91.1(1.1)	91.0(1.2)
	EMreg-oos	60	92.6(1.0)*	95.9(0.6)*	92.6(1.0)	92.6(1.0)*
	EMreg-KNN	60	92.3(1.0)*	95.7(0.6)*	92.3(1.0)	92.2(1.0)*

Tabla de contenidos

- 1 El problema
- 2 Estado del arte
- 3 Marco teórico
- 4 Propuesta
- 5 Experimentos
- 6 Conclusiones y trabajo futuro

Conclusiones

- ① EMreg-KNN junto a EMreg-oos obtienen mejores resultados en casi todos los casos estudiados.
- ② El no imputar datos, casi en la totalidad de los casos entrega peores resultados (para ADNI).
- ③ Es costoso sintonizar el parámetro K de EMreg-KNN.
- ④ EMreg-KNN como resultado adicional, calcula vecindarios en un ambiente de MV, lo que podría ser utilizado por otros métodos.

Referencias I

- [1] Roderick J. A. Little y Donald B. Rubin. *Statistical Analysis with Missing Data, Third Edition.* 2.^a ed. Wiley-Interscience, 2019. DOI: [10.1002/9781119482260](https://doi.org/10.1002/9781119482260).
- [2] Donald B. Rubin. "Multiple imputations in sample surveys - A phenomenological bayesian approach to nonresponse". En: *Proceedings of the Survey Research Methods Section*. 1978, págs. 20-28.
- [3] Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk et al. "A multivariate technique for multiply imputing missing values using a sequence of regression models". En: *Survey methodology* 27.1 (2001), págs. 85-96.
- [4] O. Troyanskaya, M. Cantor, G. Sherlock et al. "Missing value estimation methods for DNA microarrays.". En: *Bioinformatics (Oxford, England)* 17.6 (jun. de 2001), págs. 520-525. URL: <http://dx.doi.org/10.1093/bioinformatics/17.6.520>.

Referencias II

- [5] Jian-Feng Cai, Emmanuel J. Candès y Zuowei Shen. "A Singular Value Thresholding Algorithm for Matrix Completion". En: *SIAM Journal on Optimization* 20.4 (2010), págs. 1956-1982.
- [6] Gustavo E. A. P. A. Batista y Maria Carolina Monard. "A Study of K-Nearest Neighbour as an Imputation Method". En: *HIS*. 2002, págs. 251-260.
- [7] Daniel J. Stekhoven y Peter Bühlmann.
"MissForest—non-parametric missing value imputation for mixed-type data". En: *Bioinformatics* 28.1 (2012), págs. 112-118.
- [8] Sanaz Nikfalazar, Chung-Hsing Yeh, Susan E. Bedingfield et al.
"Missing data imputation using decision trees and fuzzy clustering with iterative learning". En: *Knowledge and Information Systems* 62 (2019), págs. 2419-2437.

Referencias III

- [9] Chung-Yuan Cheng, Wan-Ling Tseng, Ching-Fen Chang et al. "A Deep Learning Approach for Missing Data Imputation of Rating Scales Assessing Attention-Deficit Hyperactivity Disorder". En: *Frontiers in Psychiatry* 11 (2020), pág. 673.
- [10] Najmeh Abiri, Björn Linse, Patrik Edén et al. "Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems". En: *Neurocomputing* 365 (2019), págs. 137-146.
- [11] Tapio Schneider. "Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values". En: *Journal of Climate* 14 (2001), págs. 853-871.

GR__AS P_R SU ATENCIÓN

Repository EMreg-KNN



Apendice

Algoritmo EMreg

Algoritmo EM regularizado (T. Schneider)

La matriz de coeficientes de regresión β puede ser estimada mediante:

$$\hat{\beta} = \hat{\Sigma}_{oo}^{-1} \hat{\Sigma}_{om} \quad (8)$$

$\hat{\Sigma}_{oo}$ es la matriz de covarianza estimada de variables con datos observables y $\hat{\Sigma}_{om} = \hat{\Sigma}_{mo}^T$ es la matriz de covarianza estimada de variables con datos observables y faltantes.

Algoritmo EM regularizado (T. Schneider)

Luego que los datos faltantes son imputados, se itera para actualizar μ y Σ mediante:

$$\hat{\mu}^{t+1} = \frac{1}{n} \sum_{i=1}^n X_i \quad (9)$$

donde el nuevo estimador de la media es la media muestral, y:

$$\hat{\Sigma}^{t+1} = \frac{1}{\tilde{n}} \sum_{i=1}^n \{\hat{S}_i^t - (\hat{\mu}^{t+1})^T \hat{\mu}^{t+1}\} \quad (10)$$

donde $\hat{S}^t \equiv E[X_i^T X_i | x_o; \hat{\mu}^t, \hat{\Sigma}^t]$ es la esperanza condicional y \tilde{n} es la constante de normalización.

Algoritmos de imputación

Algoritmos a competir

- MissForest [7]: utilizando una implementación en python (`missingpy`⁴). Se utilizan los parámetros por defecto con un número de árboles igual a 100 y cantidad máxima de iteraciones 10.
- MICE [3]: utilizando una implementación en python que permite crear modelos de imputación para aplicarlos al testing set. Se utilizan los parámetros por defecto.
- SVD [4]: utilizando una implementación en python del paquete `fancyimpute` que permite imputar matrices con datos faltantes. Este método no crea un modelo de imputación que permita imputar un testing set.

⁴<https://github.com/epsilon-machine/missingpy>

Clasificadores

Clasificadores usados

En cada experimento, se dividió el dataset en 75 % para entrenamiento y 25 % para prueba. Los clasificadores usados y sintonización de hiper-parámetros fue:

- K-NN: K en un rango entre 10 y $\lfloor num_subject / num_classes \rfloor$ lo cual busca relacionar la cantidad de datos con el número de clases. Para *weights* se utilizó *uniform* y *distance*.
- ν -SVM: ν en un rango $(0, 1]$, para *kernel* se utilizó *kernel polinomial* y *rbf* y *degree* = $[1, 2, 3, 4, 5]$.

Clasificadores usados

- RF: el número de árboles
 $n_estimators = [10, 50, 100, 150, 200]$ y el número de variables a considerar para realizar la mejor división
 $max_features = [\lfloor n_var/8 \rfloor, \lfloor n_var/4 \rfloor, \lfloor n_var/2 \rfloor, \lfloor n_var \cdot (3/4) \rfloor]$, donde n_var es el número de variables de los datos.
- ANN: con $dropout_rate = [0.0, 0.1, 0.3, 0.5, 0.7, 0.9]$. La arquitectura de la red, la cual se dejó fija después de probar varias configuraciones, está compuesta de 2 capas ocultas, ambas con cantidad de neuronas igual a $n_var \cdot 1.2$ con función de activación *Relu*. La red utiliza como función de pérdida *binary cross entropy*, algoritmo de optimización *Adam* y número de épocas 10.

Resultados ADNI

ADNI: MCI vs HC

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
K-NN	<i>none</i>	70.9(7.8)	72.5(10.2)	32.8(16.7)	89.4(8.4)*	39.8(16.7)
	MICE	67.7(3.8)	71.2(8.0)	41.3(17.9)	83.0(7.7)	45.3(16.0)
	SVDimpute	70.6(3.5)	76.5(3.3)	58.3(9.7)	78.0(6.0)	58.4(5.6)
	MissForest	70.2(3.1)	75.4(3.5)	55.4(9.5)	78.9(5.4)	56.9(5.7)
	EMreg	69.5(4.6)	75.8(4.0)	53.7(13.1)	79.0(7.2)	55.1(9.6)
	EMreg-oos	70.0(4.0)	75.8(3.8)	56.9(9.9)	77.9(6.0)	57.4(6.4)
SVM	EMreg-KNN	70.9(3.9)	75.9(3.4)	58.3(9.5)*	78.4(5.5)	58.8(8.1)*
	<i>none</i>	69.8(11.4)	68.8(18.4)	47.8(22.4)	81.0(15.6)	48.0(17.9)
	MICE	70.6(3.9)	72.7(11.8)	51.3(13.3)	81.6(5.9)	54.7(9.9)
	SVDimpute	70.8(9.3)	74.6(14.5)	56.9(10.6)	78.8(13.1)	58.5(8.8)
	MissForest	70.3(4.9)	74.5(13.9)	55.2(17.7)	79.3(12.0)	55.9(10.2)
	EMreg	70.5(3.8)	76.4(8.2)	53.3(13.8)	80.5(8.7)	55.8(7.6)
	EMreg-oos	71.6(4.1)	75.9(11.8)	57.3(13.4)	80.0(7.9)	58.6(7.7)
	EMreg-KNN	72.1(3.4)*	77.6(8.3)	59.2(13.7)*	79.5(6.9)	59.6(8.1)*

ADNI: MCI vs HC

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
RF	<i>none</i>	72.1(7.2)	72.4(10.8)	40.6(17.7)	87.3(6.9)	46.1(17.2)
	MICE	72.3(3.2)	77.3(3.1)	51.7(7.5)	84.1(3.8)	57.1(5.4)
	SVDimpute	72.9(3.1)	78.4(3.1)	51.0(7.6)	85.4(4.4)	57.3(5.6)
	MissForest	70.8(4.1)	76.2(3.7)	43.1(13.8)	86.7(5.6)	50.3(10.5)
	EMreg	71.2(4.3)	76.8(3.7)	44.1(15.5)	86.6(6.5)	50.7(13.2)
	EMreg-oos	72.1(3.4)	77.7(3.4)	52.8(7.0)	83.2(4.4)	57.5(5.4)
	EMreg-KNN	73.4(3.0)*	78.1(3.1)	54.7(6.3)*	84.2(3.9)	59.6(4.2)*
ANN	<i>none</i>	70.9(7.2)	74.0(10.2)	43.0(23.4)	84.8(8.0)*	44.7(20.5)
	MICE	70.2(4.2)	73.3(7.4)	54.1(16.8)	79.5(8.6)	54.7(15.1)
	SVDimpute	72.6(3.5)	78.5(3.6)	60.2(9.1)	79.8(4.4)	61.0(6.8)
	MissForest	71.8(3.5)	77.9(3.7)	55.1(11.2)	81.3(4.9)	57.9(7.3)
	EMreg	71.5(4.4)	77.7(3.7)	55.9(11.8)	80.5(5.7)	58.0(8.4)
	EMreg-oos	72.6(3.6)	78.8(3.6)	60.1(9.6)	80.0(4.9)	60.9(6.2)
	EMreg-KNN	73.3(3.3)*	79.4(3.2)*	61.8(9.5)*	80.1(4.5)	62.2(5.6)*

ADNI: pMCI vs sMCI

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
K-NN	<i>none</i>	54.6(8.8)	59.7(9.8)	53.2(18.2)*	60.9(19.5)	54.9(12.5)
	MICE	62.4(4.8)	66.1(5.8)	42.0(12.3)	79.0(9.2)	48.0(8.8)
	SVDimpute	63.9(4.9)	69.9(5.2)	45.9(12.0)	78.2(8.4)	51.4(8.1)
	MissForest	63.5(5.5)	69.3(5.0)	47.4(11.4)	76.1(8.2)	52.1(8.0)
	EMreg	63.9(4.5)	70.0(4.7)	45.5(11.3)	78.2(8.0)	51.2(8.0)
	EMreg-oos	63.5(4.6)	70.1(4.4)	46.5(11.4)	77.0(8.6)	51.5(7.5)
	EMreg-KNN	65.3(4.8)	71.1(4.9)	50.8(9.2)	76.8(7.8)	55.4(5.8)*
SVM	<i>none</i>	52.0(9.5)	51.4(12.0)	59.2(24.7)*	46.7(26.3)	55.1(14.8)
	MICE	62.3(4.5)	67.2(7.8)	50.4(12.9)	71.9(10.3)	52.7(7.5)
	SVDimpute	63.4(3.8)	69.5(4.4)	51.9(9.9)	72.4(7.3)	54.4(6.2)
	MissForest	62.3(5.7)	67.1(9.8)	54.4(14.3)	68.9(13.1)	54.4(8.1)
	EMreg	62.9(4.4)	65.2(10.9)	51.7(10.2)	71.7(8.1)	54.0(6.4)
	EMreg-oos	63.5(4.1)	67.7(7.6)	53.7(8.6)	71.2(6.6)	55.5(5.5)
	EMreg-KNN	64.1(5.0)	70.6(4.4)*	56.6(12.0)	70.3(10.8)	57.0(6.4)*

ADNI: pMCI vs sMCI

Classifier	Imputation	Acc. (%)	AUC (%)	Sens. (%)	Spec. (%)	F (%)
RF	<i>none</i>	57.1(8.8)	61.9(10.1)	64.3(15.4)*	51.3(16.4)	61.5(9.3)*
	MICE	62.4(4.8)	67.9(5.2)	49.6(8.2)	72.5(7.0)	52.8(5.8)
	SVDimpute	63.0(4.5)	69.4(4.5)	52.0(9.2)	72.0(7.0)	54.3(5.3)
	MissForest	62.1(4.8)	67.6(4.5)	52.9(13.5)	69.9(9.8)	53.6(8.0)
	EMreg	62.3(4.7)	68.3(4.8)	48.2(12.6)	73.2(9.6)	51.5(8.1)
	EMreg-oos	63.0(4.4)	70.1(4.7)	54.2(8.4)	70.1(5.8)	55.5(5.6)
	EMreg-KNN	64.9(4.6)*	71.2(4.6)*	54.9(7.9)	72.9(6.4)	57.1(5.3)
ANN	<i>none</i>	58.2(10.1)	66.5(10.6)	68.1(18.9)*	50.1(22.7)	63.2(10.3)*
	MICE	62.8(5.2)	68.1(4.7)	54.3(12.4)	69.9(9.0)	54.8(10.0)
	SVDimpute	64.6(3.9)	70.0(4.8)	56.1(9.5)	71.5(6.9)	57.2(6.0)
	MissForest	63.3(3.9)	68.4(4.3)	55.1(11.3)	69.9(8.5)	55.7(6.3)
	EMreg	64.2(4.2)	68.7(3.8)	52.9(11.5)	72.8(8.2)	55.2(8.1)
	EMreg-oos	64.1(4.3)	69.5(4.1)	54.6(11.1)	71.7(7.9)	55.8(9.3)
	EMreg-KNN	65.2(4.1)	70.7(3.8)*	57.3(8.5)	71.6(7.5)	58.3(5.2)

Resultados Caltech-101

Caltech-101

- 9146 imágenes de 101 objetos distintos (pianos, motos, caras, etc).
- Se redujo el dataset a solo 4 objetos: Caras (435), Leopardos (200), Motos (798) y Autos (123), obteniendo 1556 imágenes en total.
- Las variables estan organizadas por tipo de fuentes, las cuales son:
 - 46 Coeficientes de Gabor.
 - 40 Wavelets moments.
 - 254 CENSus TRansform hISTogram (CENTRIST features)

Se realizó el mismo procedimiento que con los datos Handwritten digits.

Caltech-101

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
K-NN	<i>none</i>	0	97.7(0.9)	98.5(0.6)	97.7(0.9)	97.7(0.9)
	MICE	20	96.1(1.4)*	97.2(1.0)	96.1(1.4)*	96.1(1.4)*
	SVDimpute	20	95.2(1.1)	96.7(0.8)	95.2(1.1)	95.2(1.1)
	MissForest	20	96.0(1.2)	97.2(0.9)	96.0(1.2)	96.0(1.2)
	EMreg	20	95.6(1.2)	97.1(0.8)	95.6(1.2)	95.6(1.2)
	EMreg-oos	20	95.7(1.2)	97.2(0.8)	95.7(1.2)	95.7(1.2)
	EMreg-KNN	20	95.6(1.3)	97.1(0.9)	95.6(1.3)	95.7(1.3)
	MICE	40	95.3(1.2)	96.5(1.0)	95.3(1.2)	95.3(1.2)
	SVDimpute	40	94.3(1.3)	95.9(1.1)	94.3(1.3)	94.3(1.4)
	MissForest	40	94.5(1.6)	96.0(1.1)	94.5(1.6)	94.5(1.6)
	EMreg	40	95.0(1.5)	96.5(1.2)	95.0(1.5)	95.0(1.5)
	EMreg-oos	40	95.2(1.4)	96.7(1.0)	95.2(1.4)	95.2(1.4)
	EMreg-KNN	40	95.4(1.3)	96.8(0.9)	95.4(1.3)	95.4(1.3)
	MICE	60	94.2(1.5)	95.5(1.2)	94.2(1.5)	94.2(1.5)
	SVDimpute	60	94.1(1.6)	95.8(1.2)	94.1(1.6)	94.1(1.6)
	MissForest	60	93.9(1.5)	95.5(1.1)	93.9(1.5)	93.9(1.5)
	EMreg	60	94.6(1.5)	96.3(1.1)	94.6(1.5)	94.6(1.5)
	EMreg-oos	60	95.2(1.4)	96.7(1.0)	95.2(1.4)	95.2(1.4)
	EMreg-KNN	60	95.2(1.2)*	96.7(0.8)	95.2(1.2)*	95.2(1.2)*

Caltech-101

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
SVM	<i>none</i>	0	99.4(0.3)	99.6(0.2)	99.4(0.3)	99.4(0.3)
	MICE	20	98.5(0.7)	98.9(0.6)	98.5(0.7)	98.5(0.7)
	SVDimpute	20	98.4(0.9)	98.8(0.6)	98.4(0.9)	98.4(0.9)
	MissForest	20	97.9(0.9)	98.3(0.7)	97.9(0.9)	97.9(0.9)
	EMreg	20	98.8(0.7)	99.1(0.5)	98.8(0.7)	98.8(0.7)
	EMreg-oos	20	98.9(0.6)	99.3(0.4)	98.9(0.6)	98.9(0.6)
	EMreg-KNN	20	98.9(0.7)	99.2(0.5)	98.9(0.7)	98.9(0.7)
	MICE	40	98.2(0.8)	98.6(0.7)	98.2(0.8)	98.2(0.8)
	SVDimpute	40	98.0(0.8)	98.5(0.6)	98.0(0.8)	98.0(0.8)
	MissForest	40	97.1(1.1)	97.8(0.8)	97.1(1.1)	97.1(1.1)
	EMreg	40	98.2(0.8)	98.7(0.6)	98.2(0.8)	98.2(0.8)
	EMreg-oos	40	98.6(0.7)	99.0(0.5)	98.6(0.7)	98.6(0.7)
	EMreg-KNN	40	98.6(0.6)	99.0(0.5)	98.6(0.6)	98.6(0.6)
	MICE	60	97.6(0.9)	98.2(0.7)	97.6(0.9)	97.6(0.9)
	SVDimpute	60	97.3(0.9)	98.0(0.7)	97.3(0.9)	97.3(0.9)
	MissForest	60	96.3(2.0)	97.2(1.4)	96.3(2.0)	96.3(1.9)
	EMreg	60	97.7(0.8)	98.4(0.6)	97.7(0.8)	97.7(0.8)
	EMreg-oos	60	98.1(0.7)	98.6(0.6)	98.1(0.7)	98.1(0.7)
	EMreg-KNN	60	98.0(0.8)	98.5(0.6)	98.0(0.8)	98.0(0.8)

Caltech-101

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
RF	<i>none</i>	0	99.1(0.4)	99.4(0.3)	99.1(0.4)	99.1(0.4)
	MICE	20	97.8(0.8)	98.3(0.6)	97.8(0.8)	97.8(0.8)
	SVDimpute	20	98.0(0.9)	98.5(0.7)	98.0(0.9)	98.0(0.9)
	MissForest	20	97.0(1.0)	97.5(0.9)	97.0(1.0)	97.0(1.0)
	EMreg	20	98.2(0.7)	98.7(0.6)	98.2(0.7)	98.2(0.7)
	EMreg-oos	20	98.4(0.7)	98.8(0.5)	98.4(0.7)	98.4(0.7)
	EMreg-KNN	20	98.3(0.8)	98.8(0.6)	98.3(0.8)	98.3(0.8)
	MICE	40	96.9(1.1)	97.6(0.9)	96.9(1.1)	96.9(1.1)
	SVDimpute	40	96.9(0.9)	97.7(0.8)	96.9(0.9)	96.9(0.9)
	MissForest	40	95.2(1.7)	96.0(1.4)	95.2(1.7)	95.2(1.7)
	EMreg	40	97.3(0.9)	98.0(0.7)	97.3(0.9)	97.3(0.9)
	EMreg-oos	40	97.9(0.8)	98.5(0.6)	97.9(0.8)	97.9(0.8)
	EMreg-KNN	40	97.7(0.7)	98.3(0.6)	97.7(0.7)	97.7(0.7)
	MICE	60	96.5(1.0)	97.2(0.8)	96.5(1.0)	96.4(1.0)
	SVDimpute	60	96.3(1.1)	97.1(0.9)	96.3(1.1)	96.3(1.1)

Caltech-101

Classifier	Imputation	MV rate (%)	Acc. (%)	AUC (%)	Sens. (%)	F (%)
ANN	<i>none</i>	0	99.7(0.3)	99.8(0.2)	99.7(0.3)	99.7(0.3)
	MICE	20	98.5(0.7)	98.9(0.6)	98.5(0.7)	98.5(0.7)
	SVDimpute	20	98.1(1.2)	98.6(1.0)	98.1(1.2)	98.1(1.3)
	MissForest	20	97.8(0.9)	98.4(0.6)	97.8(0.9)	97.8(0.8)
	EMreg	20	98.5(0.8)	98.9(0.6)	98.5(0.8)	98.5(0.8)
	EMreg-oos	20	98.7(0.7)	99.0(0.6)	98.7(0.7)	98.7(0.7)
	EMreg-KNN	20	98.4(0.7)	98.8(0.6)	98.4(0.7)	98.4(0.7)
	MICE	40	98.1(0.9)	98.6(0.7)	98.1(0.9)	98.1(0.9)
	SVDimpute	40	97.6(0.9)	98.2(0.7)	97.6(0.9)	97.6(0.9)
	MissForest	40	96.0(1.7)	97.1(1.1)	96.0(1.7)	96.0(1.7)
	EMreg	40	97.8(1.2)	98.3(1.1)	97.8(1.2)	97.8(1.3)
	EMreg-oos	40	98.2(1.0)	98.7(0.7)	98.2(1.0)	98.2(1.0)
	EMreg-KNN	40	98.2(0.8)	98.6(0.6)	98.2(0.8)	98.2(0.8)
	MICE	60	97.5(1.0)	98.1(0.8)	97.5(1.0)	97.5(0.9)
	SVDimpute	60	97.1(1.2)	97.9(0.9)	97.1(1.2)	97.1(1.2)
	MissForest	60	95.2(2.1)	96.6(1.4)	95.2(2.1)	95.3(2.0)
	EMreg	60	97.1(0.9)	97.8(0.7)	97.1(0.9)	97.1(0.9)
	EMreg-oos	60	97.8(0.9)*	98.3(0.6)	97.8(0.9)*	97.8(0.9)*
	EMreg-KNN	60	97.5(0.9)	98.1(0.7)	97.5(0.9)	97.5(0.9)

Mecanismo de MV

Missing completely at random (MCAR): el mecanismo MCAR ocurre cuando la probabilidad de que una variable tenga valor faltante es independiente de la misma variable y de cualquier otra influencia externa. MCAR puede ser expresada por:

$$p[M | X_o, X_m, \xi] = p[M | \xi] \quad (11)$$

lo cual significa que la falta de valor en las variables no depende de los valores de entrada.

Missing at Random (MAR): el mecanismo MAR ocurre cuando la probabilidad de que una variable tenga valor faltante es independiente de las variables con valores faltantes pero dependiente de las otras variables con valores observables. MAR puede ser expresada por:

$$p[M | X_o, X_m, \xi] = p[M | X_o, \xi] \quad (12)$$

Not Missing at Random (NMAR): el mecanismo NMAR ocurre cuando la probabilidad de que una variable tenga valor faltante no es al azar, por lo tanto depende de las variables faltantes.

En contraste al mecanismo MAR, los valores faltantes no pueden ser estimados solo con las variables con datos observables. En este caso, no hay un método general para manejar los datos faltantes apropiadamente.

Deep Learning no siempre gana.



ELSEVIER

Expert Systems with Applications

Volume 227, 1 October 2023, 120201



Deep learning versus conventional methods for missing data imputation: A review and comparative study

Yige Sun ^a✉, Jing Li ^a✉, Yifan Xu ^b✉, Tingting Zhang ^c✉, Xiaofeng Wang ^d✉