



INTELIGENCIA DE NEGOCIO

2020 - 2021

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- Tema 7. Modelos Avanzados de Minería de Datos.
- Tema 8. Big Data

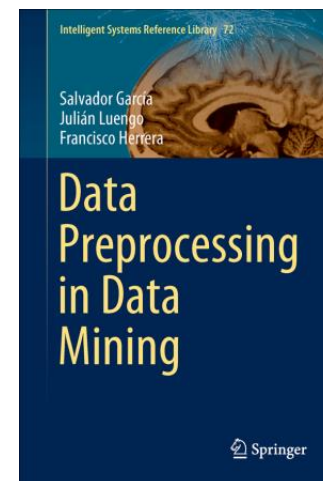
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

Bibliografía:

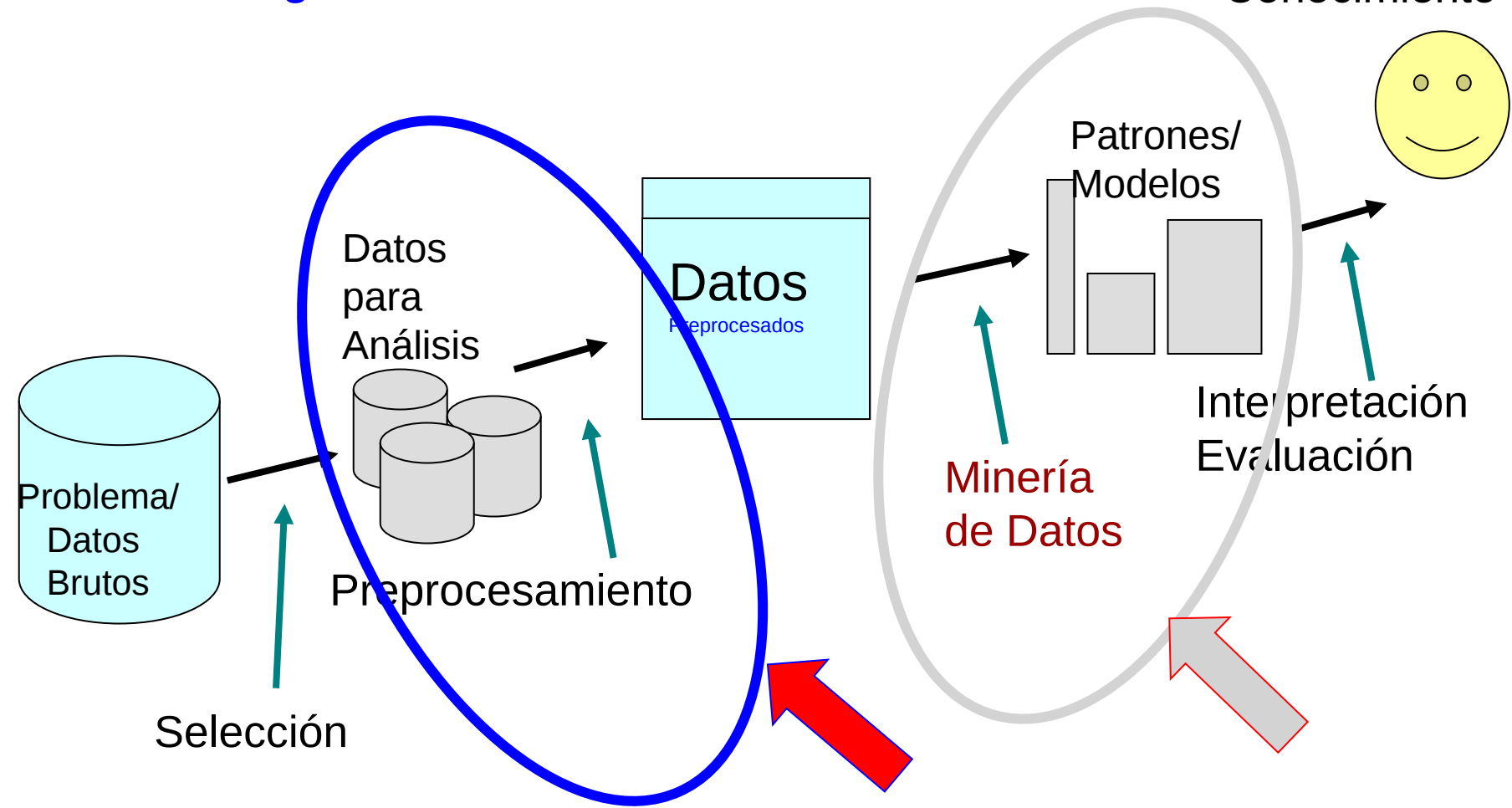
S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



Motivación

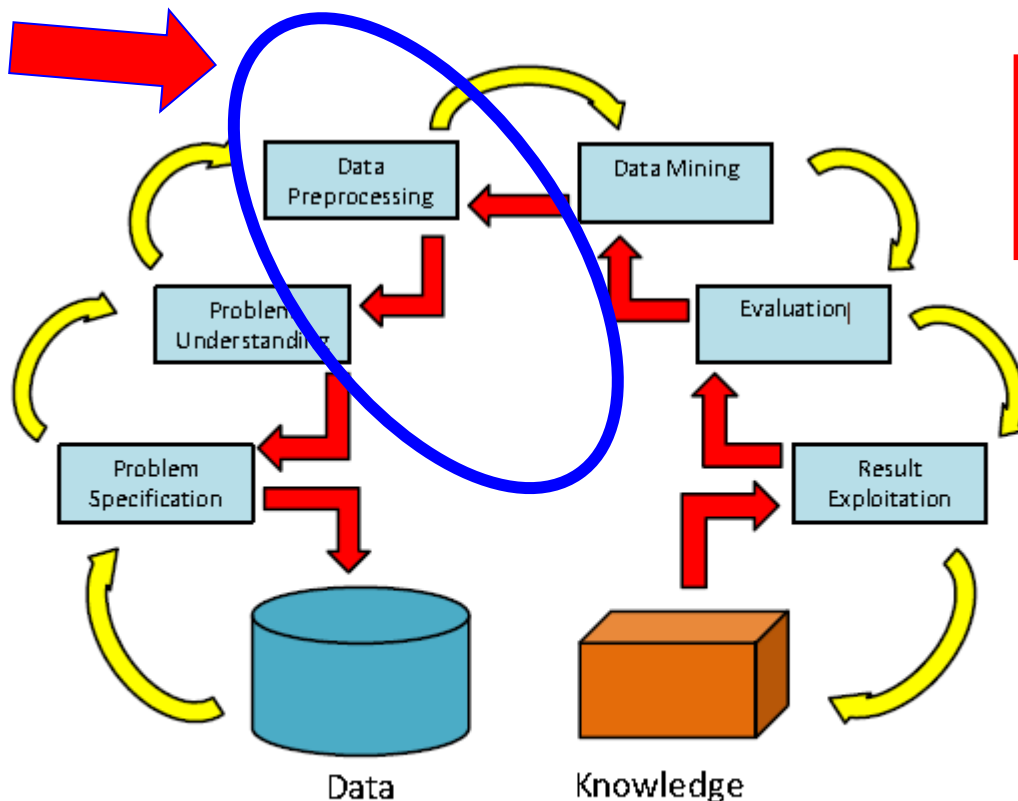
Preprocesamiento: Tareas para disponer de datos de calidad previos al uso de algoritmos de extracción de conocimiento.

Conocimiento



Motivación

Preprocesamiento: Tareas para disponer de datos de calidad previos al uso de algoritmos de extracción de conocimiento.



**Quality Data for
Quality Decisions**

Fig. 1.1: KDD process.

Objetivos



- Entender los distintos problemas a resolver en procesos de recopilación y preparación de datos.
- Conocer problemas presentes en la integración de datos de distintas fuentes y técnicas para resolverlos.
- Conocer problemas a resolver para limpiar los datos y procesar datos imperfectos y algunas técnicas que los resuelven.
- Entender la necesidad, en ocasiones, de aplicar técnicas de transformación de datos.
- Conocer las técnicas de reducción de datos y la necesidad de aplicación.

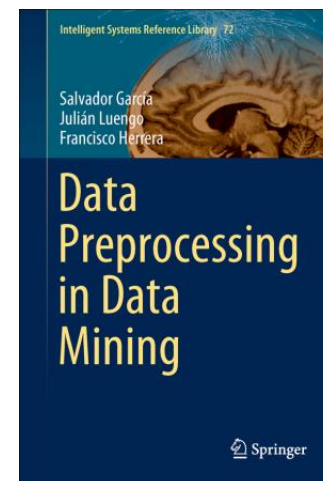
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

Bibliografía:

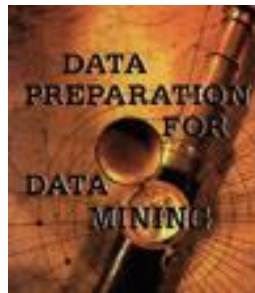
S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



INTRODUCCIÓN

D. Pyle, 1999, pp. 90:

“The fundamental purpose of data preparation is to manipulate and transform raw data so that the information content enfolded in the data set can be exposed, or made more easily accesible.”



Dorian Pyle
Data Preparation for Data Mining Morgan Kaufmann Publishers, 1999

INTRODUCCIÓN

D. Pyle, 1999, pp. 90:

“El propósito fundamental de la preparación de los datos es la manipulación y transformación de los datos sin refinar para que la información contenida en el conjunto de datos pueda ser descubierta o estar accesible de forma más fácil.”



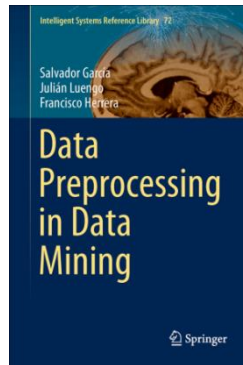
Dorian Pyle
Data Preparation for Data Mining Morgan Kaufmann Publishers, 1999

Data Preprocessing

S. García, J. Luengo, F. Herrera, 2015, Preface vii:

“Data preprocessing is an often neglected but major step in the data mining process.”

“El preprocesamiento de datos es un paso a menudo descuidado pero muy importante en el proceso de minería de datos”

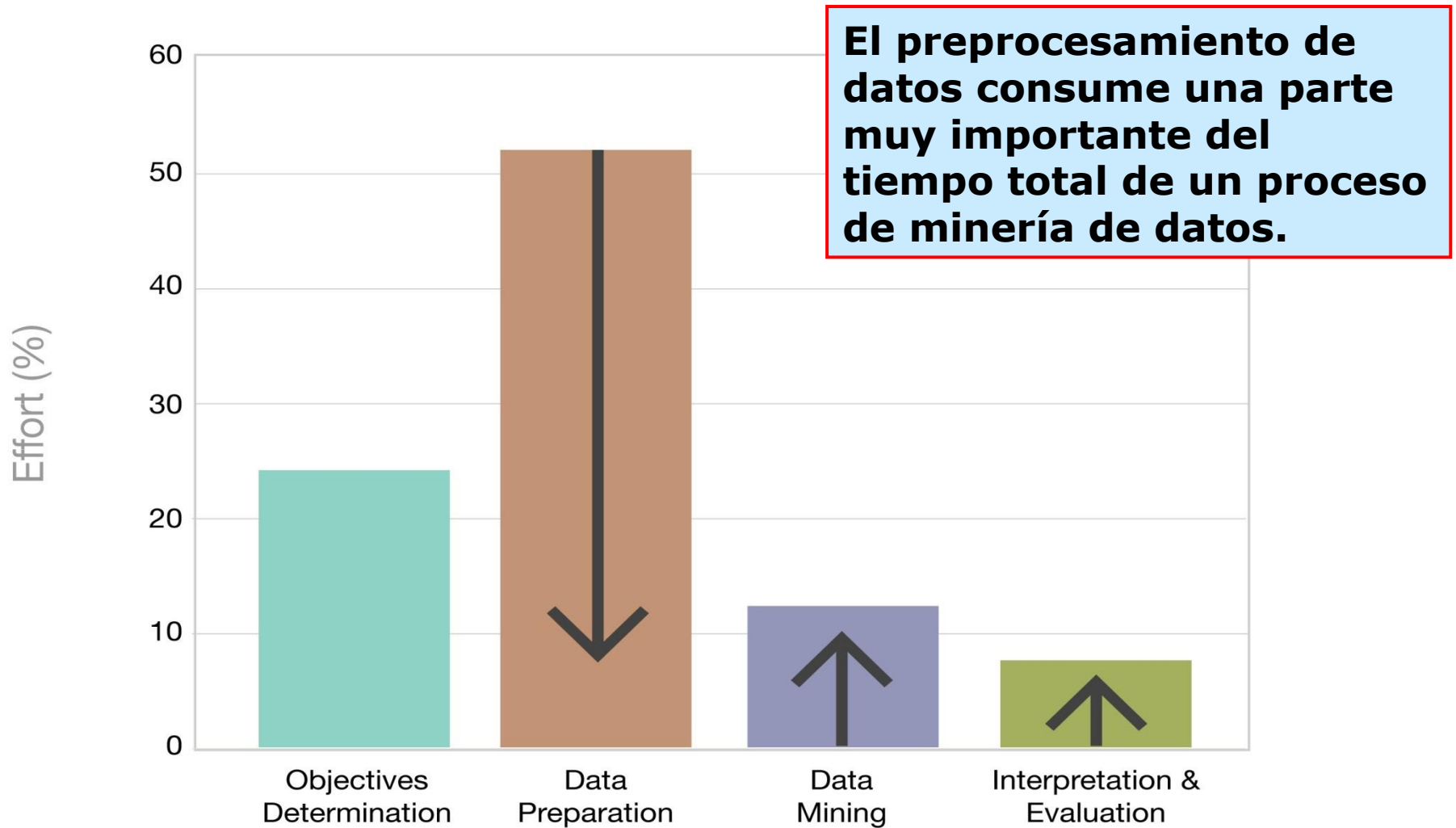


S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

Website:

<http://sci2s.ugr.es/books/data-preprocessing>

INTRODUCCIÓN



Preprocesamiento de Datos

- ❑ **Importancia del Preprocesamiento de Datos**
- ❑ **¿Qué incluye el Preprocesamiento de Datos?**

Preprocesamiento de Datos

Importancia del Preprocesamiento de Datos

1. Los datos reales pueden ser impuros, pueden conducir a la extracción de patrones/reglas poco útiles.

Esto se puede deber a:

Datos Incompletos: falta de valores de atributos, ...

Datos con Ruido

Datos inconsistentes (incluyendo discrepancias)

Preprocesamiento de Datos

Importancia del Preprocesamiento de Datos

2. El preprocesamiento de datos puede generar un conjunto de datos más pequeño que el original, lo cual puede mejorar la eficiencia del proceso de Minería de Datos.

Esta actuación incluye:

Selección relevante de datos: eliminando registros duplicados, eliminando anomalías, ...

Reduccion de Datos: Selección de características, muestreo o selección de instancias, discretización.

Preprocesamiento de Datos

Importancia del Preprocesamiento de Datos

3. El preprocesamiento de datos genera “datos de calidad”, los cuales pueden conducir a “patrones/reglas de calidad”.

Por ejemplo, se puede:

Recuperar información incompleta.

Eliminar outliers

Resolver conflictos

Seleccionar variables relevantes, ...

Preprocesamiento de Datos

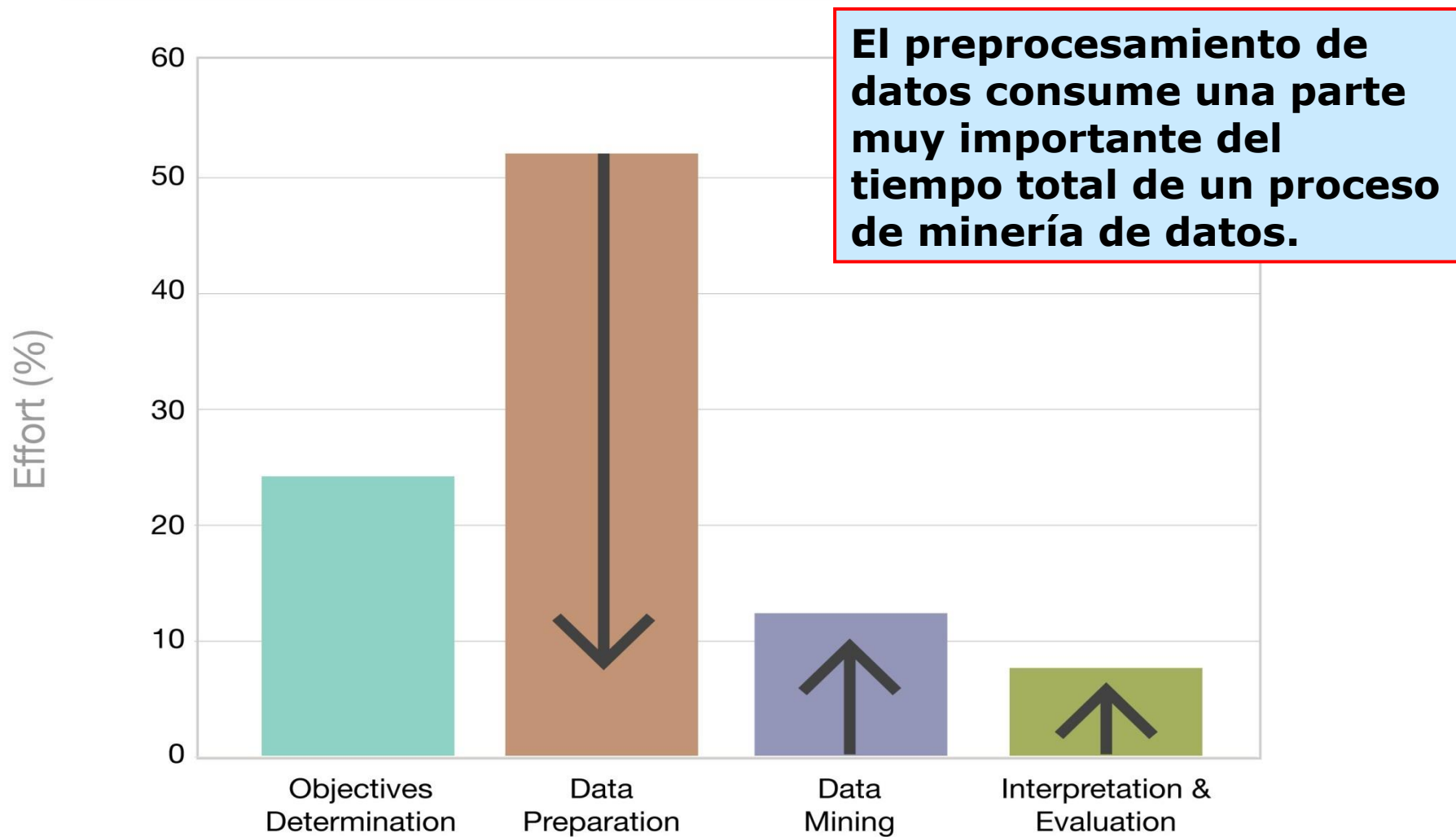
Importancia del Preprocesamiento de Datos

- Datos de baja calidad puede llevar a modelos de minería de datos de baja calidad.

Decisiones de calidad deben ser basadas en datos de calidad.

- El preprocesamiento de datos (limpieza, transformación, reducción....) puede llevar la mayor parte del tiempo de trabajo en una aplicación de minería de datos (80%).

Preprocesamiento de Datos



Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

“El Preprocesamiento de Datos”

engloba a todas aquellas técnicas de análisis de datos que permite mejorar la calidad de un conjunto de datos de modo que las técnicas de extracción de conocimiento/minería de datos puedan obtener mayor y mejor información (mejor porcentaje de clasificación, reglas con más completitud, etc.)

Preprocesamiento de Datos

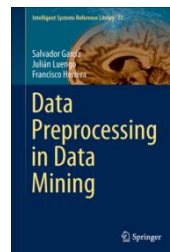
¿Qué incluye el Preprocesamiento de Datos?

S. García, J. Luengo, F. Herrera, 2015, Preface vii:

“Data preprocessing includes data preparation, compounded by integration, cleaning, normalization and transformation of data; and data reduction tasks; such as feature selection, instance selection, discretization, etc.

...

The result expected after a reliable chaining of data preprocessing tasks is a final dataset, which can be considered correct and useful for further data mining algorithms.”



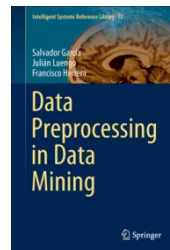
S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

We refer to **data preparation** as the set of techniques that initialize the data properly to serve as input for a certain DM algorithm.

Data reduction comprises the set of techniques that, in one way or another, obtain a reduced representation of the original data.



S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

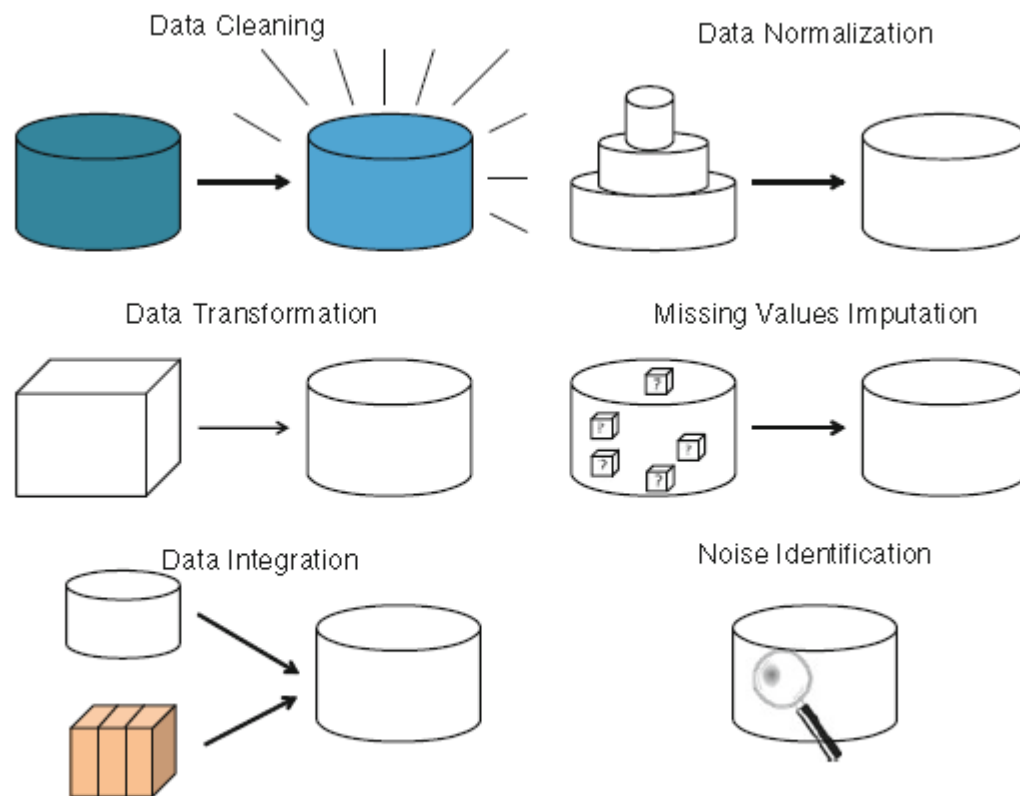


Fig. 1.3 Forms of data preparation

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

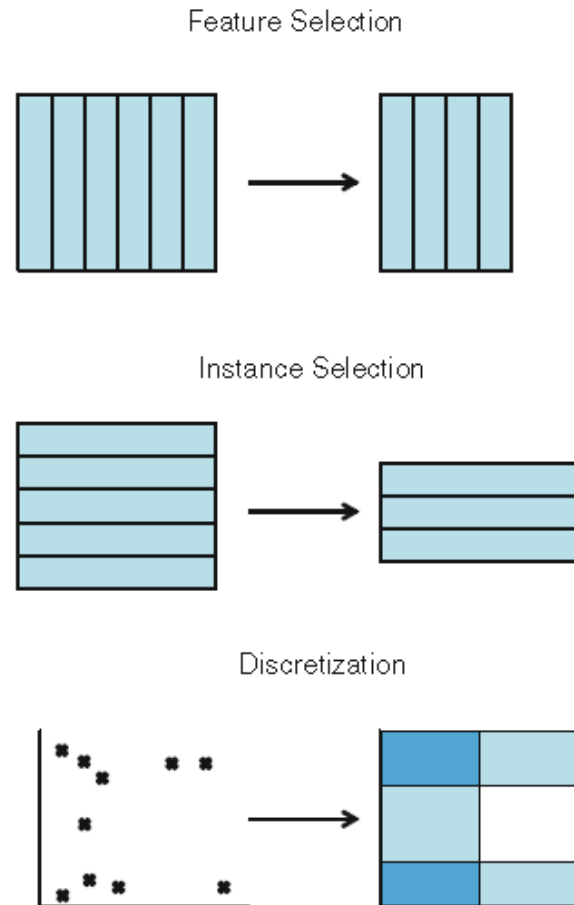


Fig. 1.4 Forms of data reduction

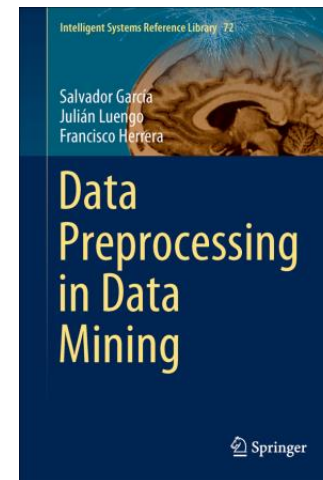
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

Bibliografía:

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



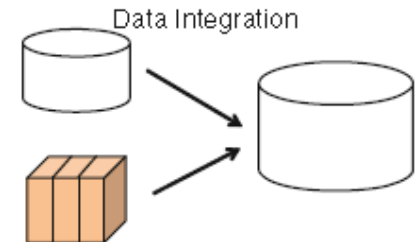
Snapshot on Data Preparation

- How do I clean up the data?—**Data Cleaning**



- How do I incorporate and adjust data?

—**Data Integration**

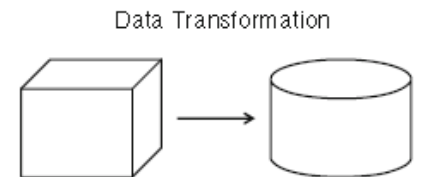


How do I provide accurate data?

—**Data Transformation**

- How do I unify and scale data?

—**Data Normalization**



Integración, Limpieza y Transformación

Integración, Limpieza y Transformación

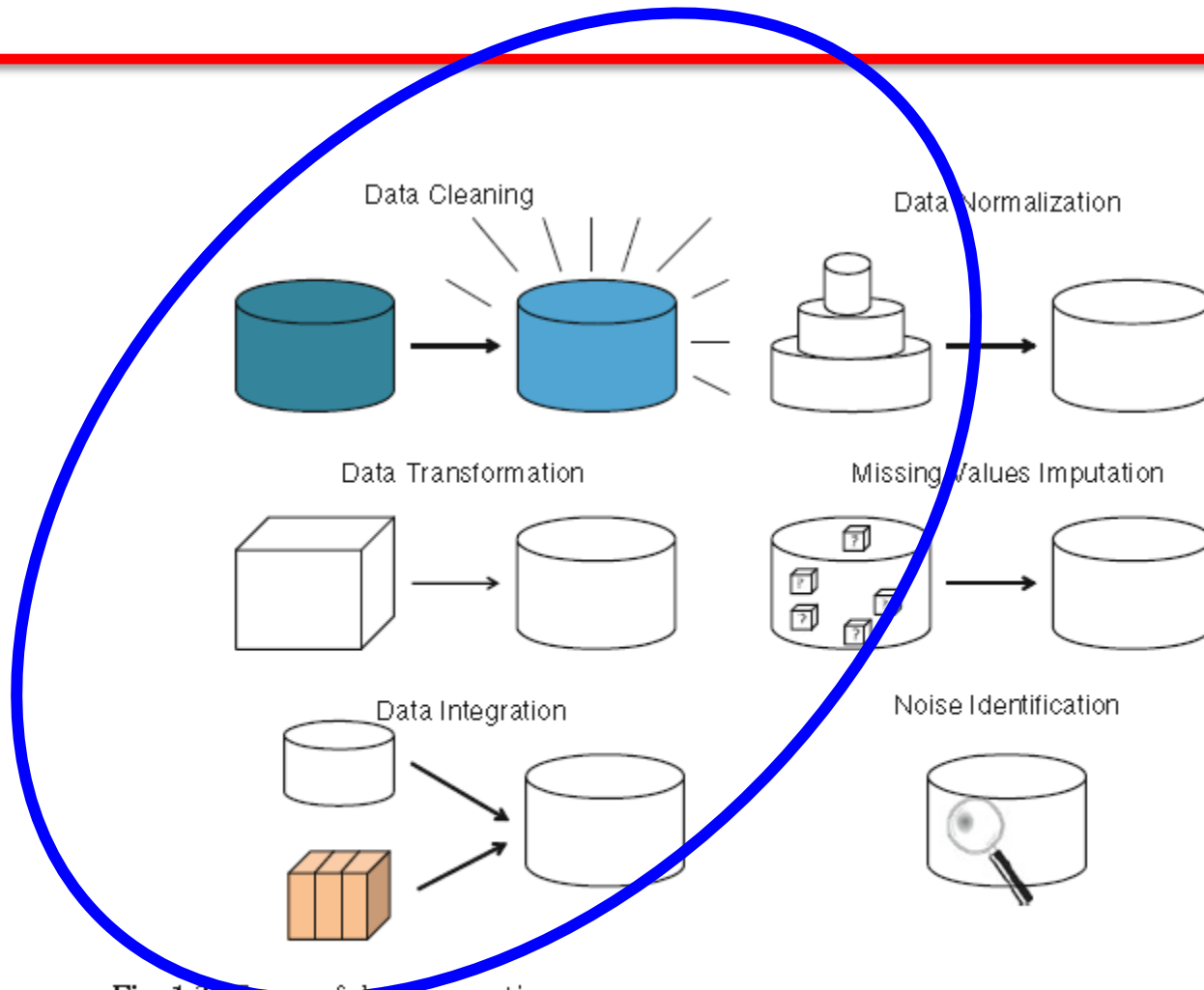


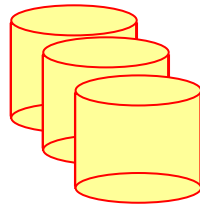
Fig. 1.3 Forms of data preparation

Integración de datos

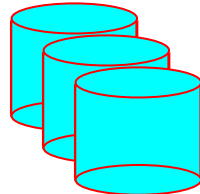
- ✿ Obtiene los datos de diferentes fuentes de información
- ✿ Resuelve problemas de representación y codificación
- ✿ Integra los datos desde diferentes tablas para crear información homogénea, ...

Integración de datos

Base de
Datos 1



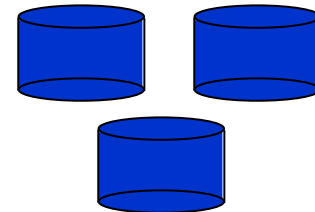
Base de
Datos 2



Extracción,
agregación ..



Servidor
Data Warehouse



Integración de datos

Ejemplos

- Diferentes escalas: Salario en dólares versus peniques



- Atributos derivados: Salario mensual versus salario anual

item	Salario/mes
1	5000
2	2400
3	3000

item	Salario
6	50,000
7	100,000
8	40,000

Integración de datos

Cuestiones a considerar al realizar la integración de datos desde distintas fuentes:

- **Integración del esquema.** ¿Cómo asegurar que entidades equivalentes se emparejan correctamente cuando se produce la fusión desde distintas fuentes?.

Ejemplo: *id-cliente* y *num-cliente*.

Solución: Utilizar los metadatos que normalmente se almacenan en las BBDD y los DW.

- **Detección de datos duplicados e inconsistencias.**
- **Redundancia.** Un atributo es redundante si puede obtenerse a partir de otros.
Una forma de detectar redundancia es mediante análisis de correlaciones.

Integración de datos

Análisis de correlaciones

Objetivo: medir la fuerza con la que un atributo implica a otro, en función de los datos disponibles.

La correlación entre dos atributos A y B puede medirse como.

$$r_{A,B} = \frac{\sum (A - \bar{A})(B - \bar{B})}{(n-1)\sigma_A\sigma_B}$$

n : número de datos

$\bar{A}$: media

σ_A : desviación estándar

- $r_{A,B} > 0 \rightarrow A$ y B están correlacionados positivamente (ambas tienen comportamiento similar)
- $r_{A,B} = 0 \rightarrow A$ y B son independientes
- $r_{A,B} < 0 \rightarrow A$ y B están correlacionados negativamente (si un atributo crece, el otro decrece)

Integración de datos

Análisis de correlaciones

Ejemplo1: Con las variables x , x^2 y $1/x$ ($x=1,...5$)

	x	x^2	$1/x$
x	1		
x^2	0.98	1	
$1/x$	-0.9	-0.81	1

Ejemplo2:

	Edad	Tensión	Obesidad	Colesterol	Tabaquismo	Alcoholismo	Pulsaciones	Hierro
Edad	1							
Tensión	0.63	1						
Obesidad	0.34	0.22	1					
Colesterol	0.42	0.56	0.67	1				
Tabaquismo	-0.02	0.72	0.72	0.52	1			
Alcoholismo	0.15	0.43	0.32	0.27	0.58	1		
Pulsaciones	0.12	0.27	0.32	0.40	0.39	0.23	1	
Hierro	-0.33	-0.08	0.21	0.45	-0.12	-0.22	-0.15	1

Integración de datos

- **Detección y resolución de conflictos en los valores de los datos:** Un atributo puede diferir según la fuente de procedencia.
 - Puede deberse a diferencias en la representación, escala, o forma de codificar.
 - Ejemplos:
 - peso en kg. o en libras.
 - precio en función de la moneda o de si los impuestos están o no incluidos, etc.

Cuidar el proceso de integración a partir de múltiples fuentes reducirá y evitará redundancias e inconsistencias en los datos resultantes, mejorando la exactitud y velocidad del proceso de DM.

Limpieza de datos

- **Objetivos:**
 - resolver inconsistencias
 - Rellenar/imputar valores perdidos,
 - suavizar el ruido de los datos,
 - identificar o eliminar *outliers* ...

- Algunos algoritmos de DM tienen métodos propios para tratar con datos incompletos o ruidosos. Pero en general estos métodos no son muy robustos, lo normal es realizar previamente la limpieza de los datos.

Bibliografía:

W. Kim, B. Choi, E.-D. Hong, S.-K. Kim

A taxonomy of dirty data.

Data Mining and Knowledge Discovery 7, 81-99, 2003.

```
000000000130.06.19971979-10-3080145722 #000310 111000301.01.000100000000004
0000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.0000
00000000000. 000000000000000.000000000000000.0000000.....
000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.00
0000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.0000
000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000
0000000000.000000000000000.000000000000000.000000000000000.00 0000000000300.00 0000000000300.00
```

[illegible]

Limpieza de datos

Limpieza de Datos: Ejemplo de Inconsistencia

Presencia de discrepancias en datos

Edad="42"

Fecha de Nacimiento="03/07/1997"

Normalización

- **Objetivo:** pasar los valores de un atributo a un rango mejor.
- Útil para algunas técnicas como AANN o métodos basados en distancias (vecinos más próximos,...).
- Algunas técnicas de normalización:
 - **Normalización min-max:** Realiza una transformación lineal de los datos originales.

$$[\min_A, \max_A] \rightarrow [\text{nuevo}_{\min_A}, \text{nuevo}_{\max_A}]$$
$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{nuevo}_{\max_A} - \text{nuevo}_{\min_A}) + \text{nuevo}_{\min_A}$$

Las relaciones entre los datos originales se conservan.

Normalización

- **Normalización *zero-mean***. Se normaliza en función de la media y la desviación estándar.

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

Útil cuando se desconocen los límites o cuando los datos anómalos pueden dominar la normalización min-max.

- **Normalización por escala decimal**. Normaliza moviendo el punto decimal de los valores del atributo. El número de puntos decimales movidos depende del valor absoluto máximo de A.

$$v' = \frac{v}{10^j}$$

con j igual al menor entero tal que $\max(|v'|) < 1$.

P.e.: si los datos están en $[-986,917]$, entonces $j=3$.

Transformación de los datos

- **Objetivo:** Transformar los datos de la mejor forma posible para la aplicación de los algoritmos de DM.
- Algunas operaciones típicas:
 - Agregación. P.e. totalización de ventas mensuales en un único atributo que sea ventas anuales,...
 - Generalización de los datos. Se trata de obtener datos de más alto nivel a partir de los actuales, utilizando jerarquías de conceptos.
 - Calles → ciudades
 - Edad numérica → {joven, adulto, mediana-edad, anciano}
 - Normalización: Cambiar el rango $[-1,1]$ o $[0,1]$.
 - Transformaciones lineales, cuadráticas, polinomiales, ...

Bibliografía:

T. Y. Lin. Attribute Transformation for Data Mining I: Theoretical Explorations. International Journal of Intelligent Systems 17, 213-222, 2002.

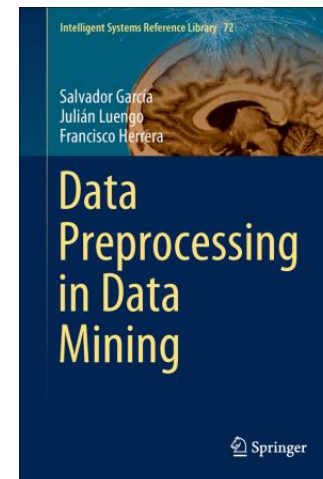
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

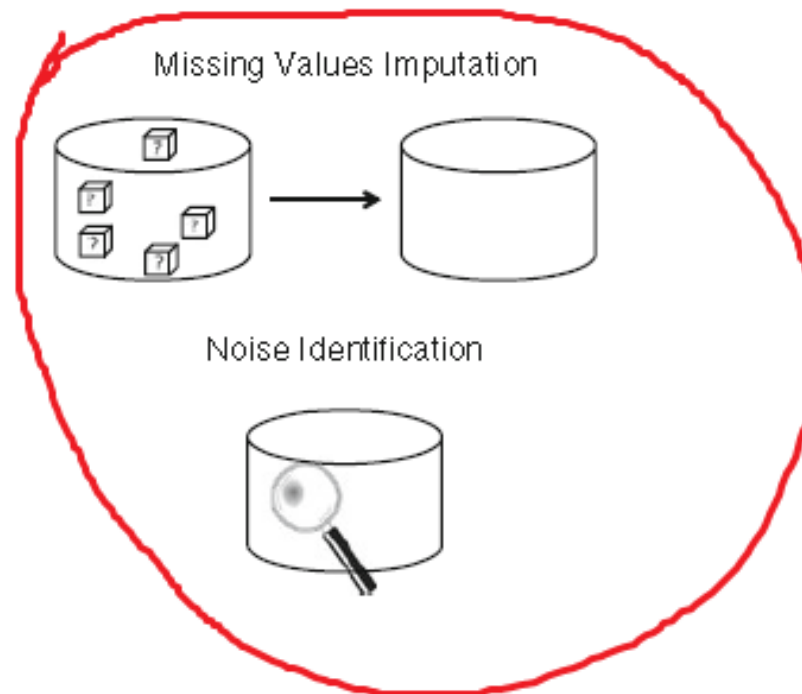
Bibliografía:

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



Snapshot on Data Preparation

- How do I handle imperfect data?— **Imputation, filtering, relabeling, ...**



Datos Imperfectos

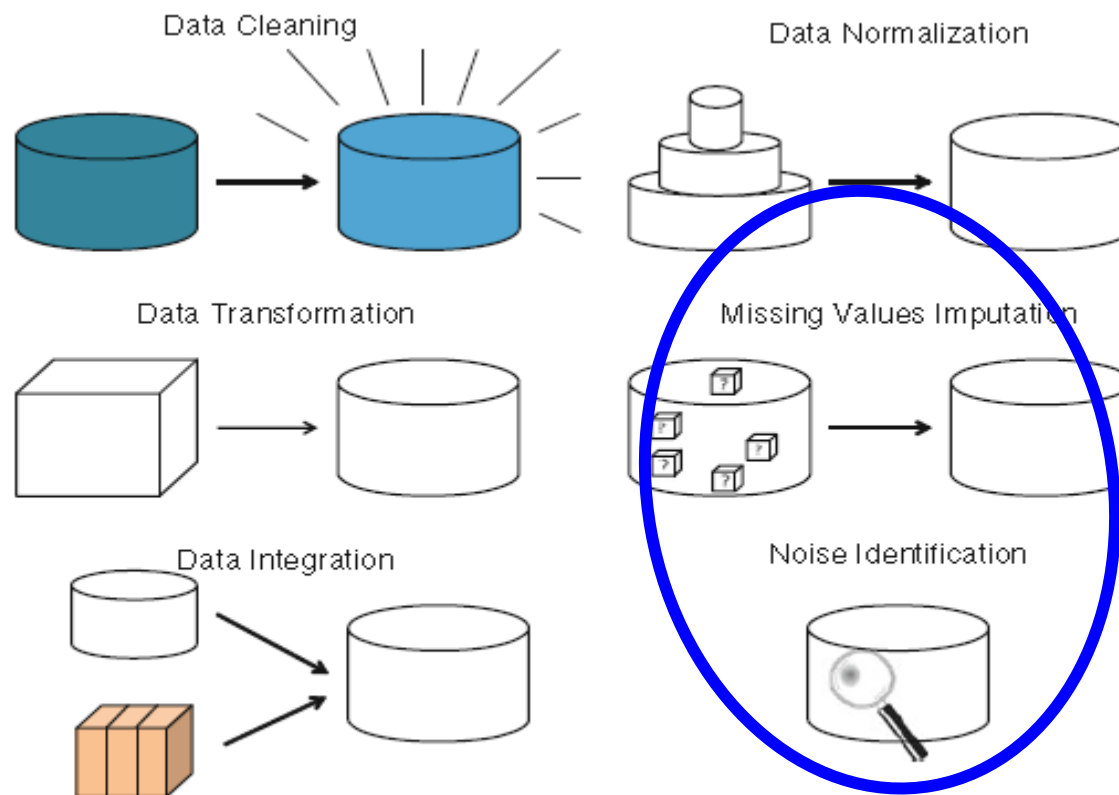


Fig. 1.3 Forms of data preparation

Datos Imperfectos

Information Sources

Attributes

Class

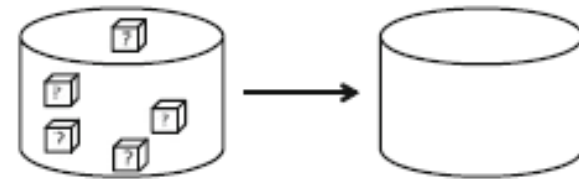
Att 1	Att 2	Class
0.25	red	positive
0.25	red	negative
0.99	green	negative
1.02	green	positive
2.05	?	negative
=	green	positive

Att. Noise

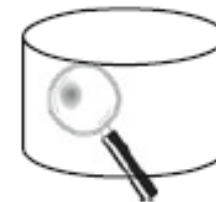
Class Noise

Kinds of Noise

Missing Values Imputation



Noise Identification



Valores perdidos

		Attributes					
		1	2	3	4	5	... m
Instances	1					?	
	2			?			
	3		?		?		
	4						
	5						
	6						?
	7			?		?	
	8						
	9						
	10			?			?
	11		?				
	:				?		
	n						?

Valores perdidos

Se pueden utilizar las siguientes opciones, aunque algunas de ellas sesgan los datos:

- **Ignorar la tupla.** Suele usarse cuando la variable a clasificar no tiene valor.
- **Rellenar manualmente los datos.** En general es impracticable.
- **Utilizar una constante global para la sustitución.** P.e. "desconocido", "?", ...
- **Rellenar utilizando la media/desviación del resto de las tuplas.**
- **Rellenar utilizando la media/desviación del resto de las tuplas pertenecientes a la misma clase.**
- **Rellenar con el valor más probable.** Para ello utilizar alguna técnica de inferencia, p.e. bayesiana o un árbol de decisión.

Valores perdidos (ejemplo preservando media y desviación)

Posición	Valor original	Pos. 11 perdida	Preservar la media	Preservar la desviación
1	0.0886	0.0886	0.0886	0.0886
2	0.0684	0.0684	0.0684	0.0684
3	0.3515	0.3515	0.3515	0.3515
4	0.9875	0.9875	0.9875	0.9875
5	0.4713	0.4713	0.4713	0.4713
6	0.6115	0.6115	0.6115	0.6115
7	0.2573	0.2573	0.2573	0.2573
8	0.2914	0.2914	0.2914	0.2914
9	0.1662	0.1662	0.1662	0.1662
10	0.4400	0.4400	0.4400	0.4400
11	0.6939	????	0.3731	0.6622
Media	0.4023	0.3731	0.3731	0.3994
SD	0.2785	0.2753	0.2612	0.2753
Error en la estimación			0.3208	0.0317

Valores perdidos (ejemplo)

- Rellenar buscando relaciones entre variables.

P.e.: de los datos de las columnas X e Y, se podría estimar $Y = 1.06 \cdot X$ y utilizarlo como estimador para valores perdidos de Y.

X (orig.)	Y (orig.)	Y estimado	error
0.55	0.53	0.51	0.02
0.75	0.37	0.31	0.06
0.32	0.83	0.74	0.09
0.21	0.86	0.85	0.01
0.43	0.54	0.63	0.09

Valores perdidos (ejemplo)

X	Y	Clase
a	a	+
a ?	n	+
n	a	-
n	n	-
n	a	+

- Estimar por el valor más probable (la moda)

$X = n \rightarrow$ error

- Estimar por el valor más probable (la moda) dentro de la clase (+)

$X = a$ (prob. 0.5) ó $X = n$ (prob. 0.5)

$\rightarrow$ No resuelve nada

Valores perdidos

Estimación mediante técnicas de aprendizaje automático:

- Imputación con k-NN (KNNI)
- Imputación con k-NN y pesos (WKNNI)
- Imputación basada en clustering (KMI)
- Imputación basada en algoritmos de SVM (SVMI)
- Otros: event covering (EC), singular value decomposition (SVDI), local least squares imputation (LLSI)

Valores perdidos

Algorithm 3 *k*NNI algorithm.

```
function kNNI(T - dataset with MVs, k - number of neighbors per instance to be chosen,  
D(x, y) - a distance or dissimilarity function of x and y, S - the imputed version of T)  
  initialize: S = {}  
  for each instance yi in T do  
     $\hat{y}_i \leftarrow y_i$   
    if yi contains any missing value then  
      Find set IKi with the k nearest instances to yi from T using D  
      for each missing value in attribute h of yi do  
        if h is numerical then  
           $\hat{y}_{ih} = \left( \sum_{j \in I_{K_ih}} y_{jh} \right) / (|I_{K_ih}|)$   
        else  
           $\hat{y}_{ih} = \text{mode}(I_{K_ih})$   
        end if  
      end for  
    end if  
     $S \leftarrow \hat{y}_{ih}$   
  end for  
  return S  
end function
```

Valores perdidos en clasificación. Ejemplo

Table 4.4 Average ranks for the Rule Induction Learning methods

	C45	Ripper	PART	Slipper	AQ	CN2	SRI	Ritio	Rules-6	Avg.	Ranks
IM	5	8.5	1	4	6.5	10	6.5	6	5	5.83	4
EC	2.5	8.5	6.5	1	6.5	5.5	6.5	6	1	4.89	3
KNNI	9	2.5	6.5	11	11	5.5	11.5	11	11	8.78	11
WKNNI	11	2.5	6.5	7	6.5	1	11.5	6	11	7.00	8
KMI	5	2.5	6.5	3	6.5	5.5	9.5	12	7.5	6.44	6
FKMI	7.5	2.5	6.5	10	2	5.5	1	2	3	4.44	1
SVMI	1	5.5	6.5	7	1	5.5	6.5	6	2	4.56	2
EM	13	12	6.5	7	12	13	3	6	4	8.50	10
SVDI	11	11	6.5	12	10	12	9.5	10	11	10.33	12
BPCA	14	13	13	7	13	14	13	13	13	12.56	14
LLSI	11	5.5	6.5	7	6.5	11	3	6	7.5	7.11	9
MC	7.5	8.5	6.5	2	6.5	5.5	3	6	7.5	5.89	5
CMC	5	8.5	12	13	3	5.5	6.5	1	7.5	6.89	7
DNI	2.5	14	14	14	14	5.5	14	14	14	11.78	13

Valores perdidos en clasificación. Ejemplo

Table 4.5 Average ranks for the Black Box methods

	RBFN	RBFD	RBFI	LOG	LVQ	MLP	NB	ν -SVM	C-SVM	SMO	Avg.	Ranks
IM	9	6.5	4.5	6	3.5	13	12	10	5.5	5.5	7.55	10
EC	1	1	1	3	7	8.5	10	13	1	2	4.75	1
KNNI	5	6.5	10.5	9	7	11	6.5	8	5.5	5.5	7.45	9
WKNNI	13	6.5	4.5	10	10	4.5	6.5	4.5	5.5	5.5	7.05	6
KMI	3.5	2	7	3	11	3	4.5	8	5.5	9	5.65	2
FKMI	12	6.5	10.5	3	1.5	4.5	11	4.5	5.5	3	6.20	3
SVMI	2	11.5	2.5	7.5	3.5	1.5	13	8	11	9	6.95	5
EM	3.5	6.5	13	12	12.5	10	4.5	4.5	10	11.5	8.80	11
SVDI	9	6.5	7	11	12.5	8.5	3	11.5	12	11.5	9.25	12
BPCA	14	14	14	13	7	14	2	2	13	13	10.60	14
LLSI	6	6.5	10.5	7.5	7	6.5	9	4.5	5.5	9	7.20	7
MC	9	6.5	10.5	3	7	6.5	8	11.5	5.5	5.5	7.30	8
CMC	9	13	2.5	3	1.5	1.5	14	14	5.5	1	6.50	4
DNI	9	11.5	7	14	14	12	1	1	14	14	9.75	13

Limpieza de datos con ruido

MISSING VALUES			
Full Name	Short Name	Reference	
Delete Instances with Missing Values	Ignore-MV	P.A. Gourraud, E. Ginin, A. Cambon-Thomsen. Handling Missing Values In Population Data: Consequences For Maximum Likelihood Estimation Of Haplotype Frequencies. European Journal of Human Genetics 12:10 (2004) 805-812.	
Event Covering Synthesizing	EventCovering-MV	D.K.Y. Chiu, A.K.C. Wong. Synthesizing Knowledge: A Cluster Analysis Approach Using Event-Covering. IEEE Transactions on Systems, Man and Cybernetics, Part B 16:2 (1986) 251-259.	
K-Nearest Neighbor Imputation	KNN-MV	G.E.A.P.A. Batista, M.C. Monard. An Analysis Of Four Missing Data Treatment Methods For Supervised learning. Applied Artificial Intelligence 17:5 (2003) 519-533.	
Most Common Attribute Value	MostCommon-MV	J.W. Grzymala-Busse, L.K. Goodwin, W.J. Grzymala-Busse, X. Zheng. Handling Missing Attribute Values in Preterm Birth Data Sets. 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'05). LNCS 3642, Springer 2005, Regina (Canada, 2005) 342-351.	
Assign All Possible Values of the Attribute	AllPossible-MV	J.W. Grzymala-Busse. On the Unknown Attribute Values In Learning From Examples. 6th International Symposium on Methodologies For Intelligent Systems (ISMIS91). Charlotte (USA, 1991) 368-377.	
K-means Imputation	KMeans-MV	J. Deogun, W. Spaulding, B. Shuart, D. Li. Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. 4th International Conference of Rough Sets and Current Trends in Computing (RSCTC'04). LNCS 3066, Springer 2004, Uppsala (Sweden, 2004) 573-579.	
Concept Most Common Attribute Value	ConceptMostCommon-MV	J.W. Grzymala-Busse, L.K. Goodwin, W.J. Grzymala-Busse, X. Zheng. Handling Missing Attribute Values in Preterm Birth Data Sets. 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'05). LNCS 3642, Springer 2005, Regina (Canada, 2005) 342-351.	



15 methods
<http://www.keel.es/>

Valores perdidos

Bibliografía:

WEBSITE: <http://sci2s.ugr.es/MVDM/>



J. Luengo, S. García, F. Herrera, **A Study on the Use of Imputation Methods for Experimentation with Radial Basis Function Network Classifiers Handling Missing Attribute Values: The good synergy between RBFs and EventCovering method.** *Neural Networks*, [doi:10.1016/j.neunet.2009.11.014](https://doi.org/10.1016/j.neunet.2009.11.014), 23(3) (2010) 406-418.

S. García, F. Herrera, **On the choice of the best imputation methods for missing values considering three groups of classification methods.** *Knowledge and Information Systems* 32:1 (2012) 77-108, [doi:10.1007/s10115-011-0424-2](https://doi.org/10.1007/s10115-011-0424-2)

Limpieza de datos con ruido

Tipos de ejemplos

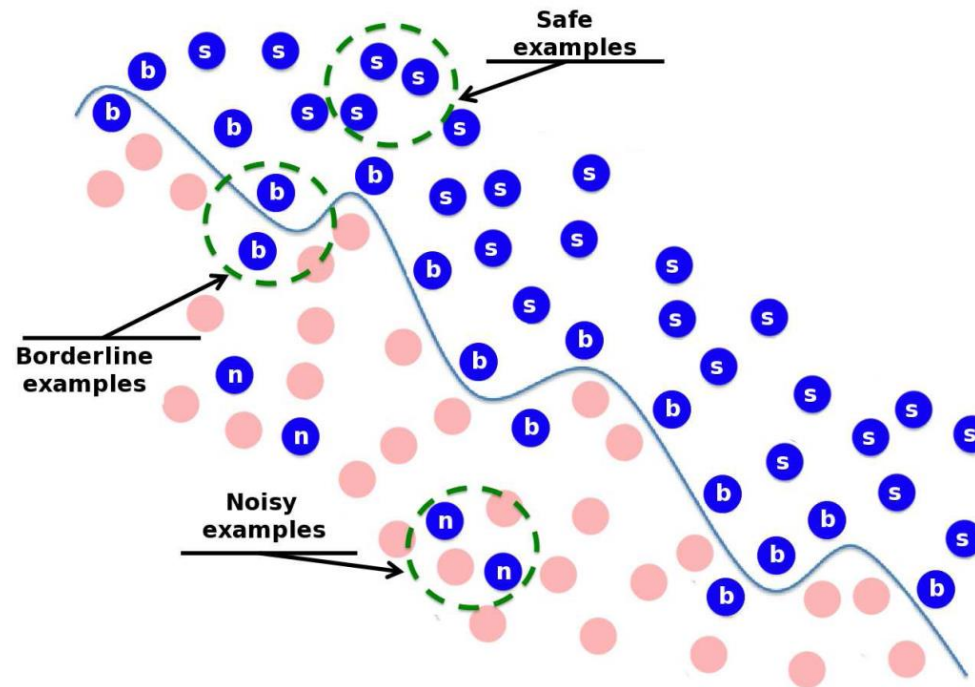


Fig. 5.2 The three types of examples considered in this book: safe examples (labeled as *s*), *borderline* examples (labeled as *b*) and *noisy* examples (labeled as *n*). The continuous line shows the decision boundary between the two classes

Limpieza de datos con ruido

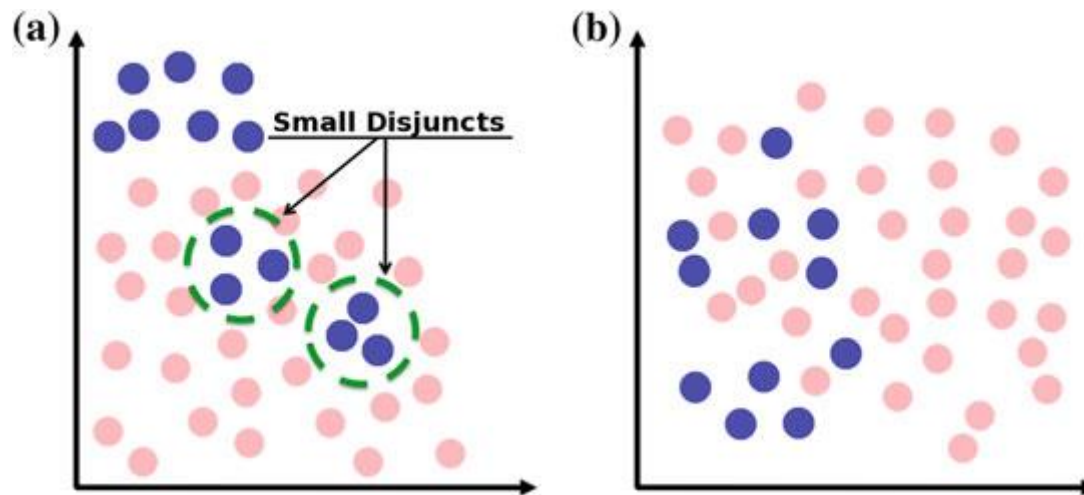


Fig. 5.1 Examples of the interaction between classes: a) small disjuncts and b) overlapping between classes

Limpieza de datos con ruido

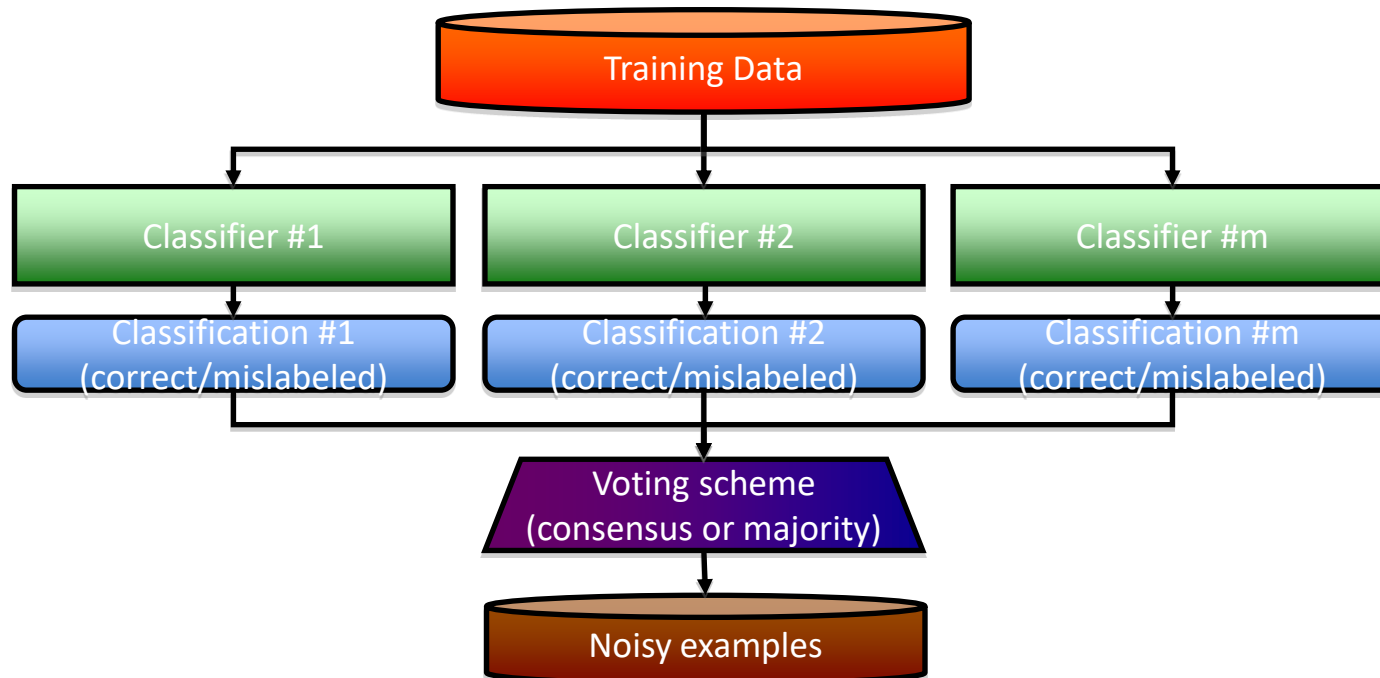
Uso de técnicas de filtrado de ruido en clasificación

Los tres filtros de ruido se mencionan a continuación , los más conocidos, utilizan un esquema de votación para determinar qué casos para eliminar del conjunto de entrenamiento.:

- ***Ensemble Filter (EF)***
- ***Cross-Validated Committees Filter***
- ***Iterative-Partitioning Filter***

Ensemble Filter (EF)

- C.E. Brodley, M.A. Friedl. **Identifying Mislabeled Training Data**. *Journal of Artificial Intelligence Research* 11 (1999) 131-167.
- **Different learning algorithm** (C4.5, 1-NN and LDA) are used to create classifiers in several subsets of the training data that serve as noise filters for the training sets.
- Two main steps:
 1. For each learning algorithm, a **k-fold cross-validation** is used to tag each training example as **correct** (prediction = training data label) or **mislabeled** (prediction $\neq$ training data label).
 2. A **voting scheme** is used to identify the final set of noisy examples.
 - **Consensus voting**: it removes an example if it is misclassified by all the classifiers.
 - **Majority voting**: it removes an instance if it is misclassified by more than half of the classifiers.



Ensemble Filter (EF)

Algorithm 4 EF algorithm.

```
function EF( $T$  - dataset with MVs,  $\Gamma$  - number of subsets,  $\mu$  - number of filters to be
used,  $F$  - set of classifiers)
    Split the training data set  $T$  into  $T_i, i = 1 \dots \Gamma$  equal sized subsets
    for each filter  $F_x, x = 1$  to  $\mu$  do
        for each subset  $T_i$  do
            Use  $\{T_j, j \neq i\}$  to train  $F_x$  resulting in  $F_x^i$ 
            for each instance  $t$  in  $T_i$  do
                Classify  $t$  with every  $F_x^i$ 
            end for
        end for
    end for
    for each instance  $t$  in  $T$  do
        Use a voting scheme to include  $t$  in  $T_N$  according to the classifications made by
        each filter  $F_x$ 
    end for
    return  $T - T_N$ 
end function
```

Cross-Validated Committees Filter (CVCF)

- S. Verbaeten, A.V. Assche. **Ensemble methods for noise elimination in classification problems**. *4th International Workshop on Multiple Classifier Systems (MCS 2003)*. LNCS 2709, Springer 2003, Guilford (UK, 2003) 317-325.
- CVCF is similar to EF → two main differences:

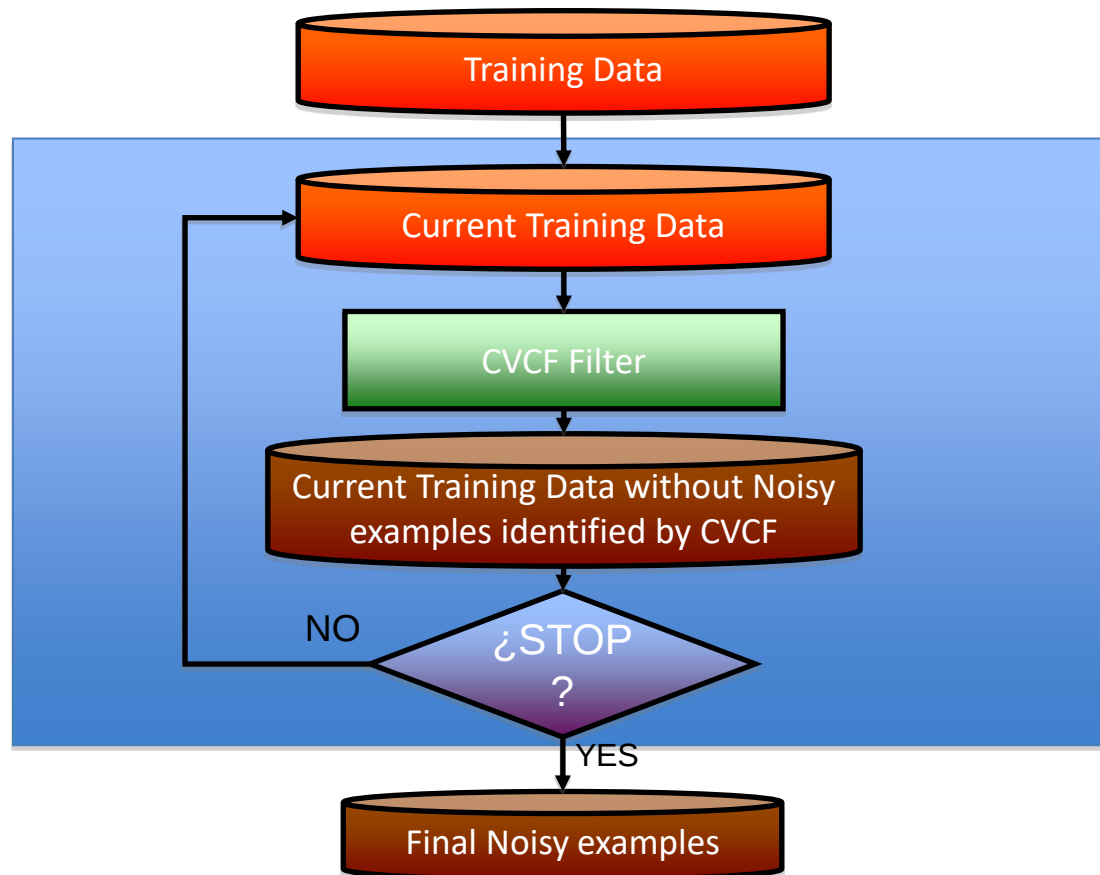
1. **The same learning algorithm (C4.5)** is used to create classifiers in several subsets of the training data.

The authors of CVCF place special emphasis on using **ensembles of decision trees** such as C4.5 because they work well as a filter for noisy data.

2. Each classifier built with the **k-fold cross-validation** is used to tag **ALL the training examples** (not only the test set) as **correct** (prediction = training data label) or **mislabeled** (prediction $\neq$ training data label).

Iterative Partitioning Filter (IPF)

- T.M. Khoshgoftaar, P. Reboours. [Improving software quality prediction by noise filtering techniques](#). *Journal of Computer Science and Technology* 22 (2007) 387-396.
- IPF removes noisy data in **multiple iterations** using **CVCF** until a stopping criterion is reached.
- The iterative process stops if, for a number of consecutive iterations, the number of noisy examples in each iteration is less than a percentage of the size of the training dataset.



Limpieza de datos con ruido

🏠 NOISY DATA FILTERING			
Full Name	Short Name	Reference	
Saturation Filter	SaturationFilter-F	D. Gamberger, N. Lavrac, S. Dzroski. Noise detection and elimination in data preprocessing: Experiments in medical domains. Applied Artificial Intelligence 14:2 (2000) 205-223.	📄
Pairwise Attribute Noise Detection Algorithm Filter	PANDA-F	J.D. Hulse, T.M. Khoshgoftaar, H. Huang. The pairwise attribute noise detection algorithm. Knowledge and Information Systems 11:2 (2007) 171-190.	📄
Classification Filter	ClassificationFilter-F	D. Gamberger, N. Lavrac, C. Groselj. Experiments with noise filtering in a medical domain. 16th International Conference on Machine Learning (ICML99). San Francisco (USA, 1999) 143-151.	📄
Automatic Noise Remover	ANR-F	X. Zeng, T. Martinez. A Noise Filtering Method Using Neural Networks. IEEE International Workshop on Soft Computing Techniques in Instrumentation, Measurement and Related Applications (SCIMA2003). Utah (USA, 2003) 26-31.	📄
Ensemble Filter	EnsembleFilter-F	C.E. Brodley, M.A. Friedl. Identifying Misabeled Training Data. Journal of Artificial Intelligence Research 11 (1999) 131-167.	📄
Cross-Validated Committees Filter	CVCcommitteesFilter-F	S. Verbaeten, A.V. Assche. Ensemble methods for noise elimination in classification problems. 4th International Workshop on Multiple Classifier Systems (MCS 2003). LNCS 2709, Springer 2003, Guilford (UK, 2003) 317-325.	📄
Iterative-Partitioning Filter	IterativePartitioningFilter-F	T.M. Khoshgoftaar, P. Rebour. Improving software quality prediction by noise filtering techniques. Journal of Computer Science and Technology 22 (2007) 387-396.	📄



<http://www.keel.es/>

Limpieza de datos con ruido

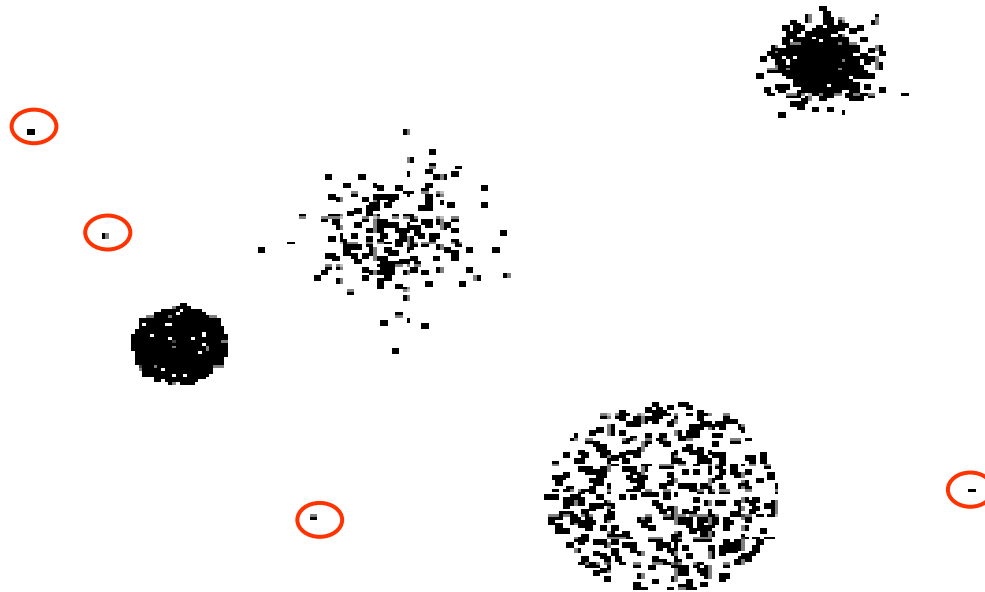
Table 5.2 Filtering of class noise over three classic classifiers

		Pairwise class noise					Uniform random class noise				
		0%	5%	10%	15%	20%	0%	5%	10%	15%	20%
SVM	None	90.02	88.51	86.97	86.14	84.86	90.02	87.82	86.43	85.18	83.20
	EF	90.49	89.96	89.07	88.33	87.40	90.49	89.66	88.78	87.78	86.77
	CVCF	90.56	89.86	88.94	88.28	87.76	90.48	89.56	88.72	87.92	86.54
	IPF	90.70	90.13	89.37	88.85	88.27	90.58	89.79	88.97	88.48	87.37
Ripper	None	82.46	81.15	80.35	79.39	78.49	82.46	79.81	78.55	76.98	75.68
	EF	83.36	82.87	82.72	82.43	81.53	83.46	83.03	82.87	82.30	81.66
	CVCF	83.17	82.93	82.64	82.03	81.68	83.17	82.59	82.19	81.69	80.45
	IPF	83.74	83.59	83.33	82.72	82.44	83.74	83.61	82.94	82.94	82.48
C4.5	None	83.93	83.66	82.81	82.25	81.41	83.93	82.97	82.38	81.69	80.28
	EF	84.18	84.07	83.70	83.20	82.36	84.16	83.96	83.53	83.38	82.66
	CVCF	84.15	83.92	83.24	82.54	82.13	84.15	83.61	83.00	82.84	81.61
	IPF	84.44	84.33	83.92	83.38	82.53	84.44	83.89	83.84	83.50	82.72

Detección de datos anómalos

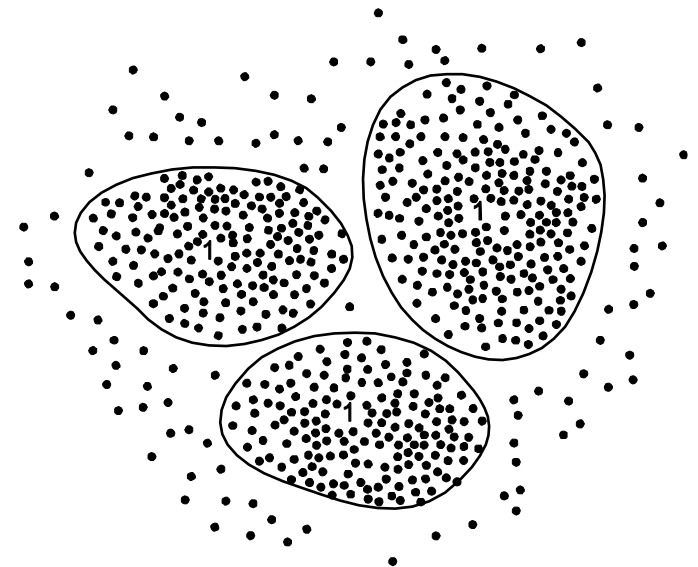
Outliers

Son objetos/datos con características que son considerablemente diferentes de la mayoría de los otros datos/objetos del conjunto.



Detección de datos anómalos

- Valores anómalos, atípicos o extremos (*outliers*): son correctos aunque sean anómalos estadísticamente.
- Pueden ser un inconveniente para métodos basados en ajuste de pesos (p.e. AANN).
- Técnicas de **detección**:
 - Definir una distancia y ver los individuos con mayor distancia media al resto de individuos.
 - Clustering parcial: los datos se agrupan en clusters y los datos que queden fuera pueden considerarse outliers.



Detección de datos anómalos

- Tratamiento de valores anómalos:
 - Ignorar. Algunos algoritmos son robustos a datos anómalos.
 - Filtrar (eliminar o reemplazar) la columna. Solución extrema, conveniente si existe una columna (atributo) dependiente con datos de mayor calidad.
 - Filtrar (eliminar o reemplazar) la fila. A veces puede sesgar los datos porque las causas de un dato erróneo están relacionadas con casos o tipos especiales.
 - Reemplazar el valor. Por valor nulo si el algoritmo de DM trabaja bien con datos nulos, con el máximo o mínimo o la media.
 - Discretizar: Si transformamos un valor continuo en discreto (muy alto,..., muy bajo), los datos anómalos caen en la categoría muy alto o muy bajo y se tratan sin problemas.

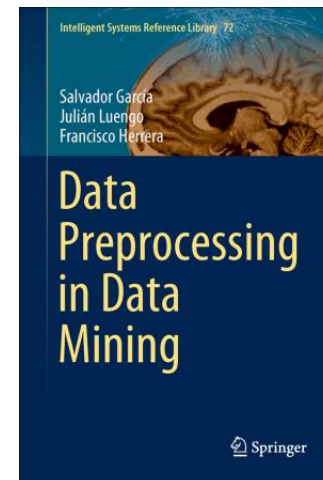
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

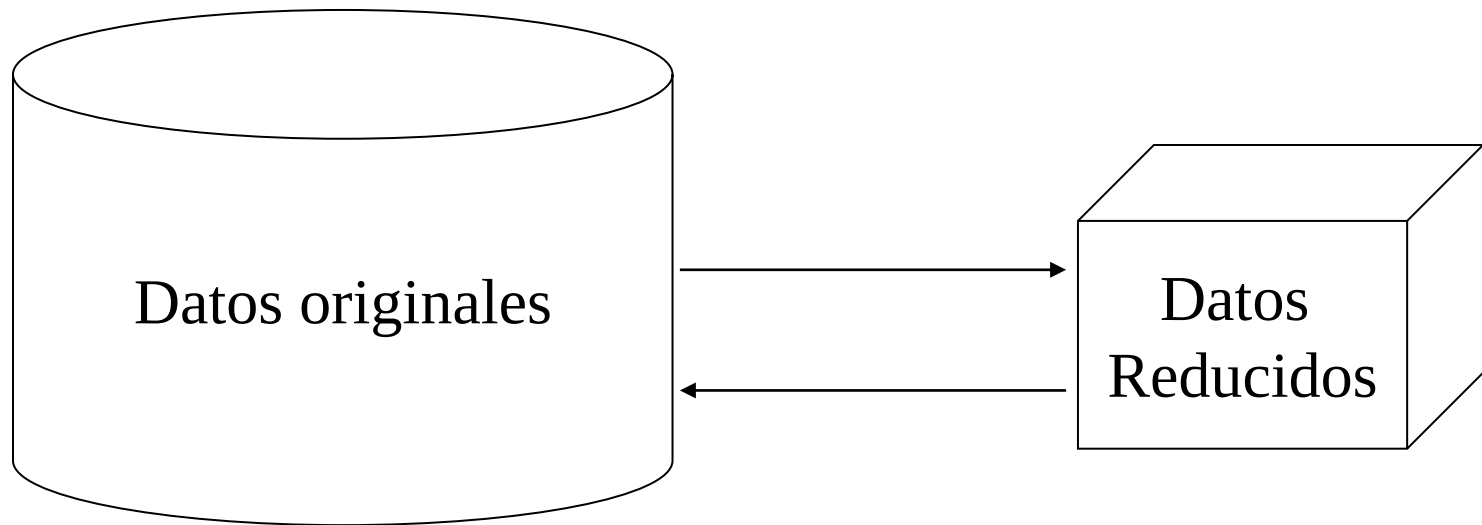
Bibliografía:

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



Reducción de Datos

- ✿ Selecciona/extrae datos relevantes para la tarea de la minería de datos/extracción de información.



Snapshot on Data Reduction

What are the basic issues that must be resolved in data reduction?

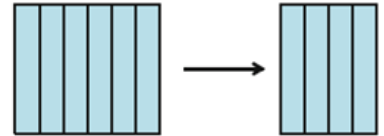
We provide a series of questions accompanied with the correct answers involving each type of process that belongs to the data reduction technique.

- How do I reduce the dimensionality of data?
- How do I remove redundant and/or conflictive examples?
- How do I simplify the domain of an attribute?
- How do I fill in gaps in data?

Snapshot on Data Reduction

- How do I reduce the dimensionality of data?

—**Feature Selection**



- How do I remove redundant and/or conflictive examples?

—**Instance Selection**

(prototype Selection vs training set selection)

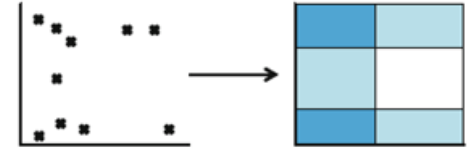


Snapshot on Data Reduction

- How do I simplify the domain of an attribute?

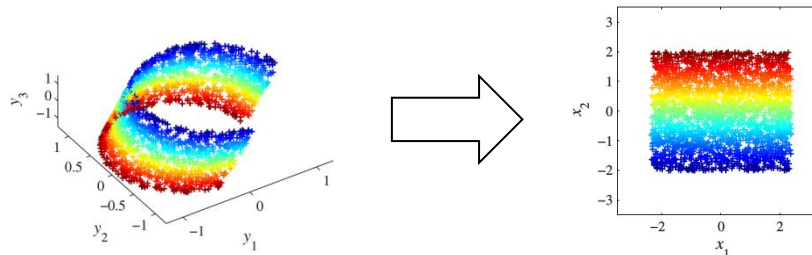
—Discretization

- Divide the range of numerical (continuous or not) attributes into intervals.
- Store the labels of the intervals.
- It is crucial for association rules and some classification algorithms, which only accept discrete data.



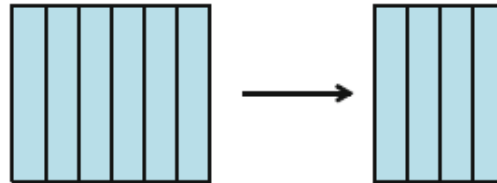
- How do I fill in gaps in data?

—Feature Extraction and/or Instance Generation



Reducción de Datos

Feature Selection



Instance Selection



Discretization

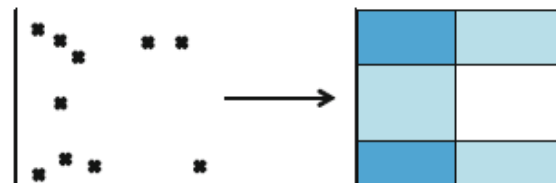
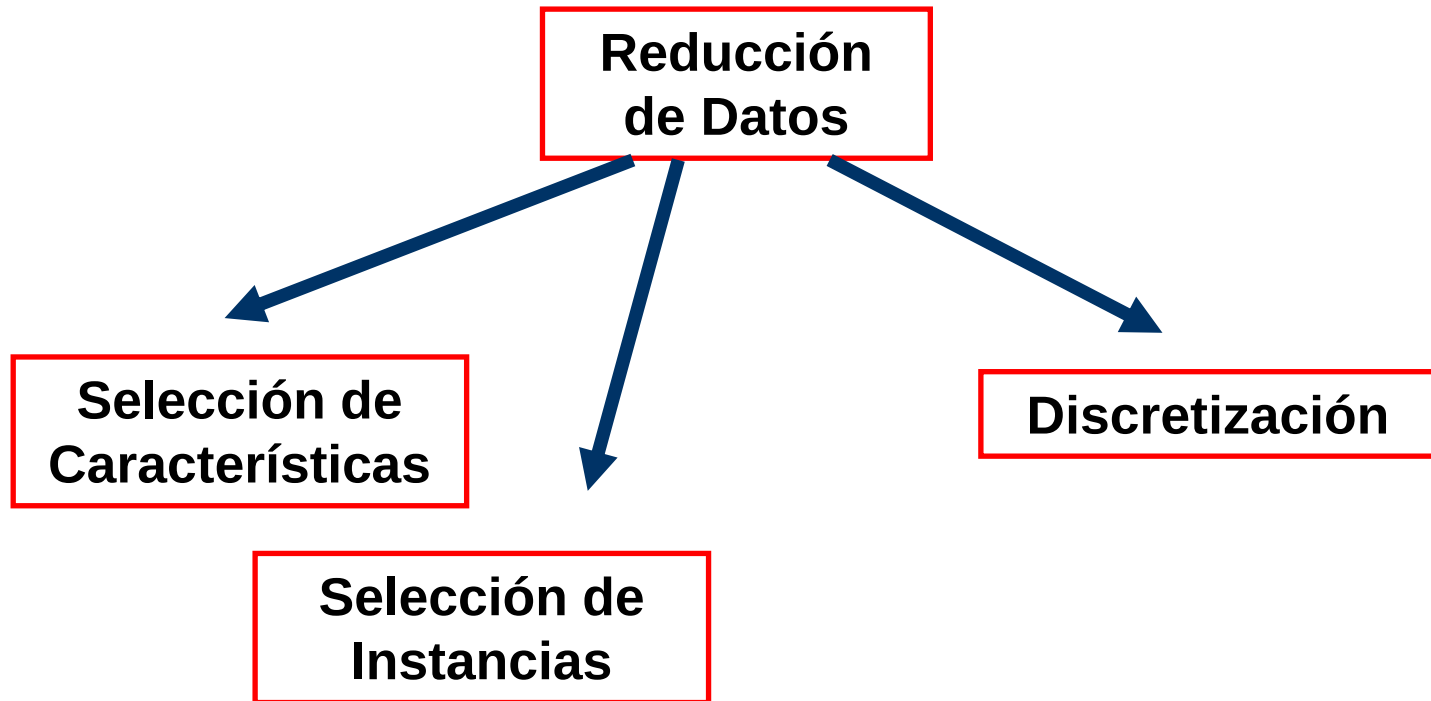


Fig. 1.4 Forms of data reduction

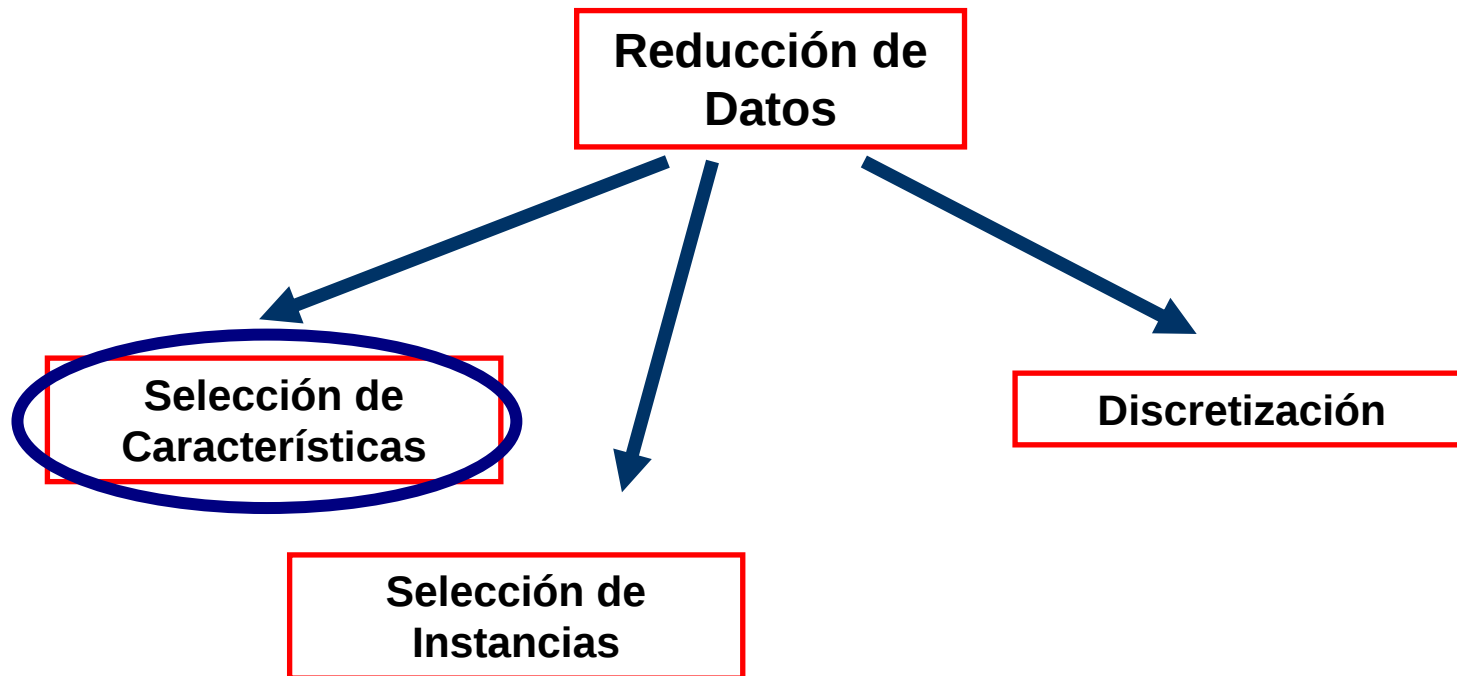
Reducción de Datos

- ✱ Diferentes vías para la Reducción de Datos:
 - ❖ Selección de Características
 - ❖ Selección de Instancias
 - ❖ Discretización

Reducción de Datos



Reducción de Datos



Bibliografía:

H. Liu, H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic, 1998.

H. Liu, H. Motoda (Eds.) Feature Extraction, Construction, and Selection: A Data Mining Perspective, Kluwer Ac., 1998.

Selección de Características

El problema de la selección de características (SC) o variables (*Feature Subset Selection, FSS*) consiste en encontrar un subconjunto de las variables del problema que optimice la probabilidad de clasificar correctamente

¿Por qué es necesaria la selección de variables?

- Más atributos no significa más éxito en la clasificación
- Trabajar con menos variables reduce la complejidad del problema y disminuye el tiempo de ejecución
- Con menos variables la capacidad de generalización aumenta
- Los valores para ciertos atributos pueden ser costosos de obtener

Selección de Características

- ✿ El resultado de la SC sería:
 - ❖ Menos datos → los algoritmos pueden aprender más rápidamente
 - ❖ Mayor exactitud → el clasificador generaliza mejor
 - ❖ Resultados más simples → más fácil de entender
- ✿ SC tiene su extensión en la Transformación (extracción y construcción de atributos)

Selección de Características

Var. 1.

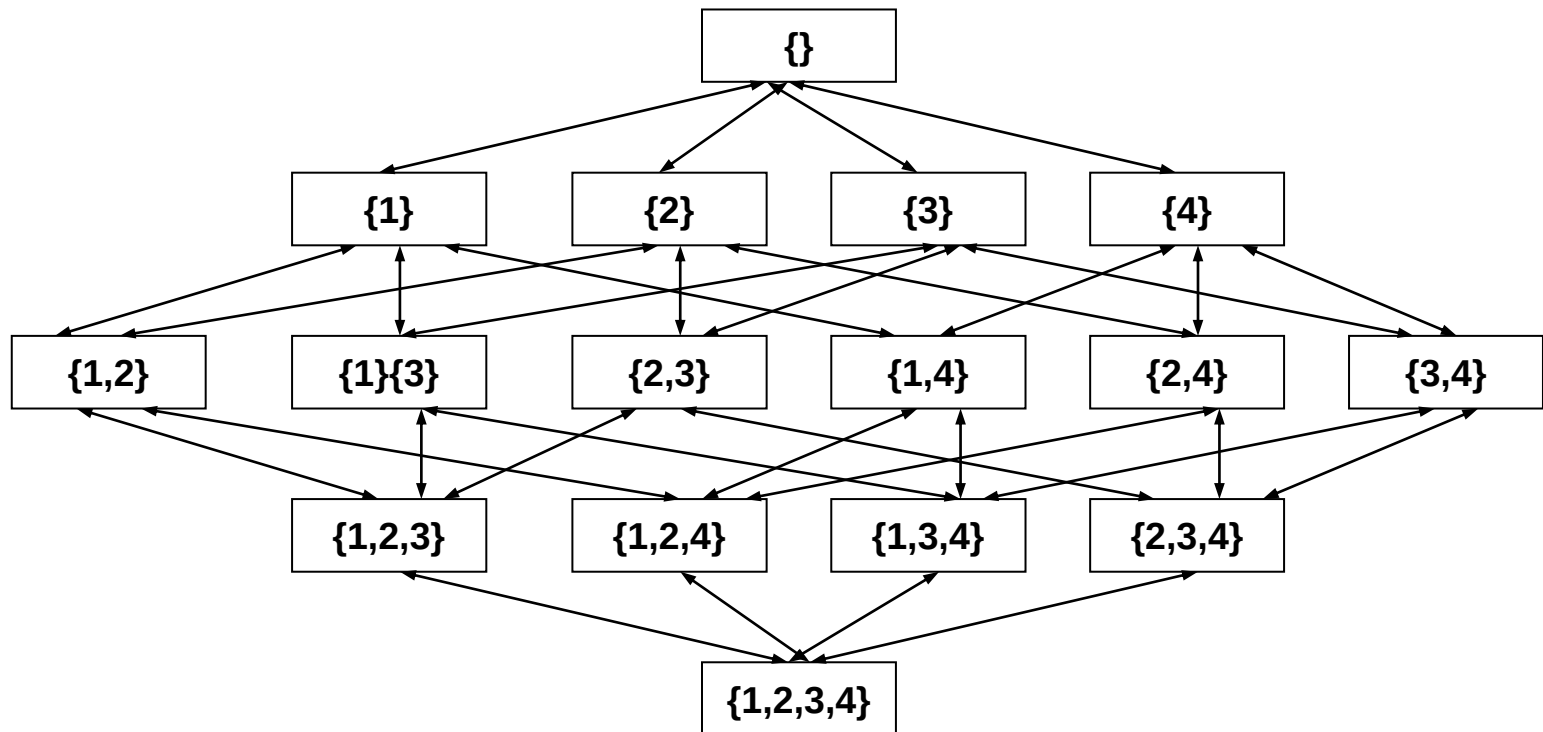
Var. 5

Var. 13

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
E	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1	0
F	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0	0

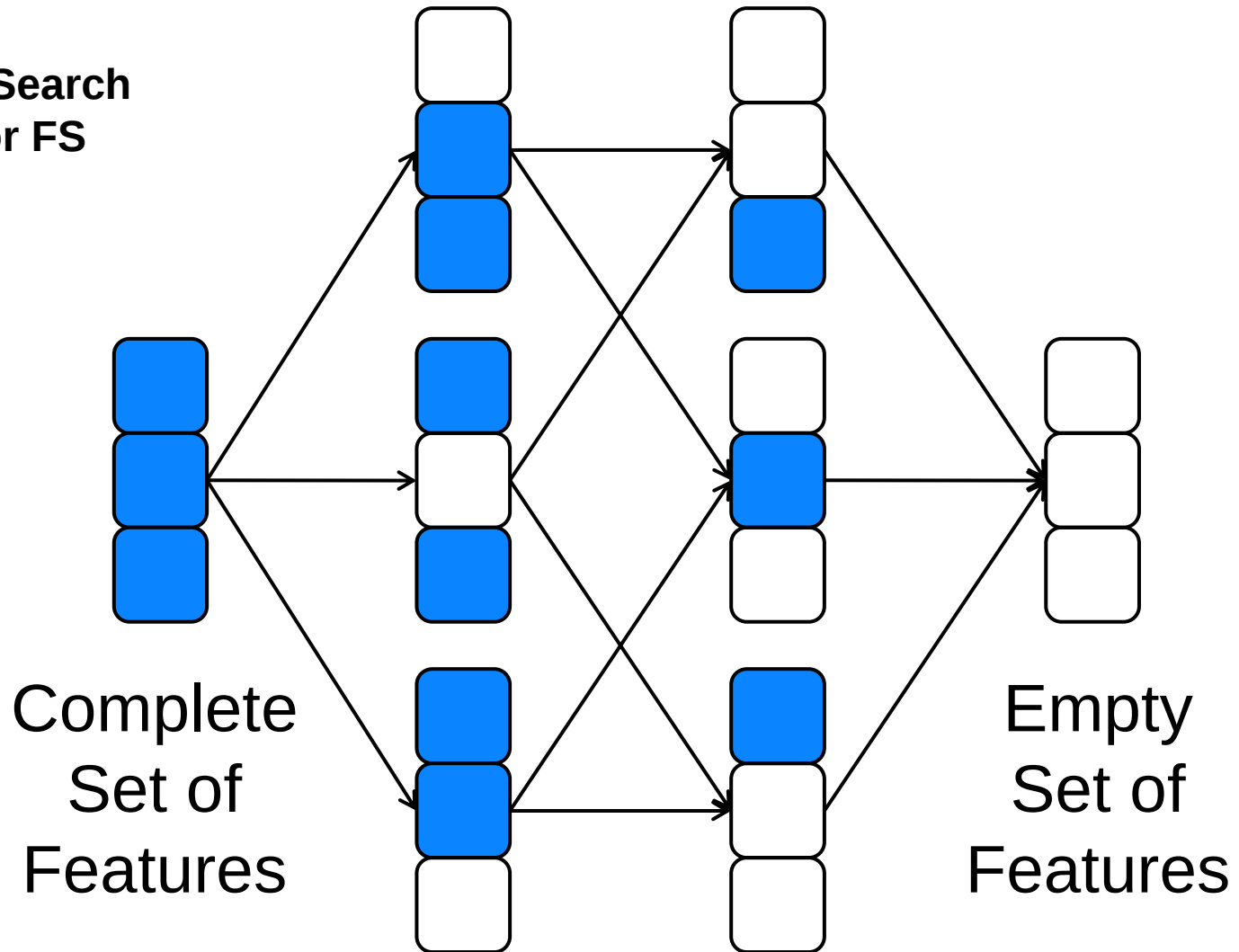
Selección de Características

La SC se puede considerar como un problema de búsqueda



Selección de Características

Fig. 7.1 Search space for FS



Selección de Características

- En un algoritmo de selección de características se distinguen dos componentes principales
 - Una estrategia de búsqueda para seleccionar subconjuntos candidatos
 - Una función objetivo que evalúe esos subconjuntos
- Estrategia de búsqueda
 - Dadas N variables, explorar todos los subconjuntos posibles supone 2^N (p.e. $2^{20}=1048576$)
 - Si queremos exactamente subconjuntos de M variables ($M \leq N$) entonces supone $\binom{N}{M}$. P.e. explorar subconjuntos de 10 variables de 20 posibles, daría 184756
 - Una búsqueda exhaustiva no es aceptable
- Función objetivo: evaluar la bondad del subconjunto seleccionado

Selección de Características

Ejemplo: Espacio de atributos para el problema *weather*

Algorithm 1 Sequential forward feature set generation - SFG.

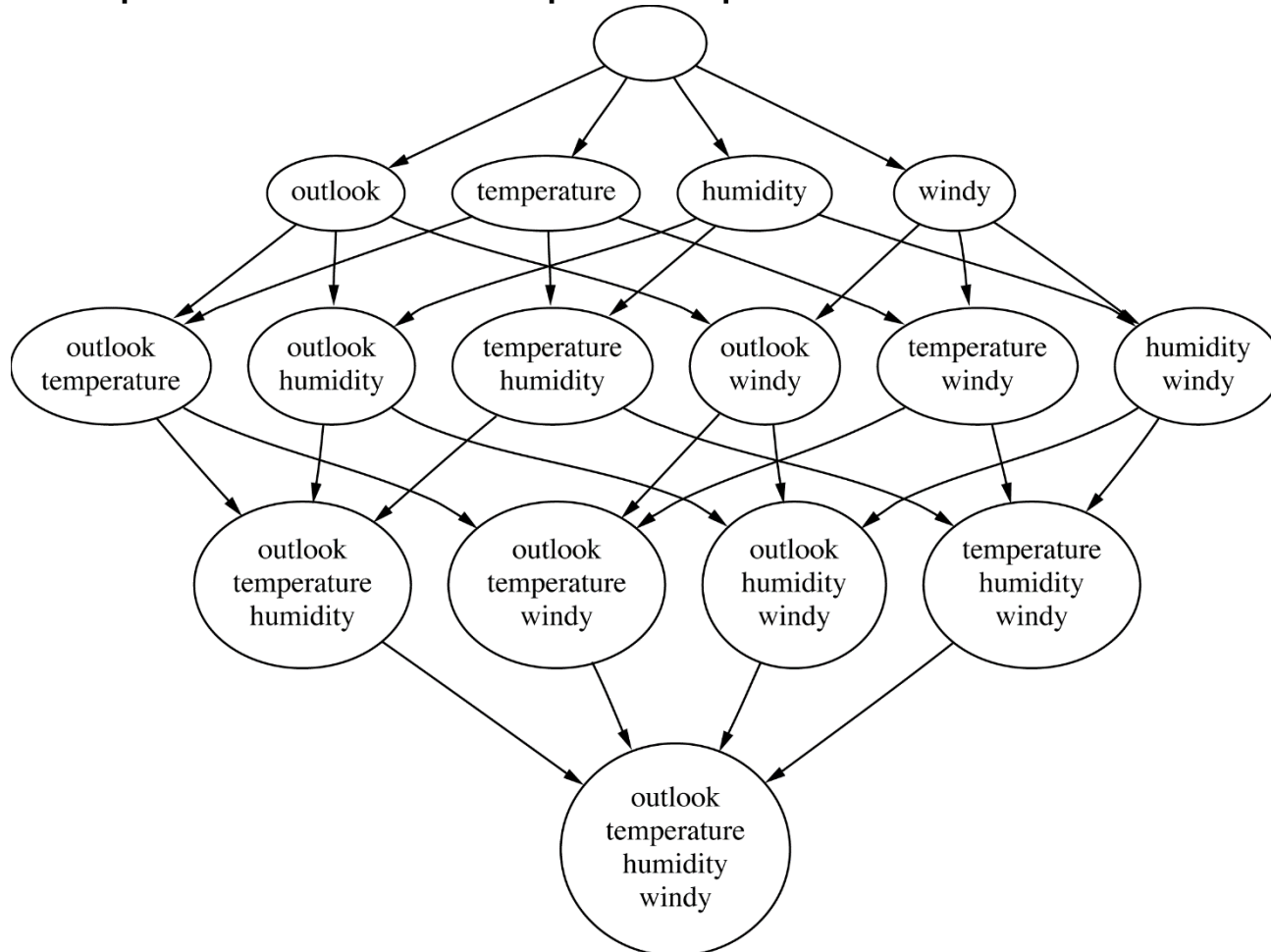
```
function SFG( $F$  - full set,  $U$  - measure)
  initialize:  $S = \{\}$  ▷  $S$  stores the selected features
  repeat
     $f = \text{FINDNEXT}(F)$ 
     $S = S \cup \{f\}$ 
     $F = F - \{f\}$ 
  until  $S$  satisfies  $U$  or  $F = \{\}$ 
  return  $S$ 
end function
```

Algorithm 2 Sequential backward feature set generation - SBG.

```
function SBG( $F$  - full set,  $U$  - measure)
  initialize:  $S = \{\}$  ▷  $S$  holds the removed features
  repeat
     $f = \text{GETNEXT}(F)$ 
     $F = F - \{f\}$ 
     $S = S \cup \{f\}$ 
  until  $S$  does not satisfy  $U$  or  $F = \{\}$ 
  return  $F \cup \{f\}$ 
end function
```

Selección de Características

Ejemplo: Espacio de atributos para el problema *weather*



Selección de Características

Funciones objetivo: Se distinguen dos enfoques distintos

- **Filtro (*filter*)**. La función objetivo evalúa los subconjuntos basándose en la información que contienen. Se utiliza como función objetivo medidas de separabilidad de clases, de dependencias estadísticas, basadas en teoría de la información,...)
- **Envolvente (*wrapper*)**. La función objetivo consiste en aplicar la técnica de aprendizaje que se utilizará finalmente sobre la proyección de los datos al conjunto de variables candidato. El valor devuelto suele ser el porcentaje de acierto del clasificador construido

Selección de Características

Medidas filtro

- **Medidas de separabilidad.** Miden la separabilidad entre clases: euclídeas, Mahalanobis,...
 - P.e. Para un problema con 2 clases, un proceso de SC basado en medidas de este tipo determina que X es mejor que Y si X induce una diferencia mayor que Y entre las dos probabilidades condicionales de las clases
- **Correlaciones.** Serán buenos subconjuntos los que estén muy correlacionados con la clase

$$f(X_1, \dots, X_M) = \frac{\sum_{i=1}^M \rho_{ic}}{\sum_{i=1}^M \sum_{j=i+1}^M \rho_{ij}}$$

donde ρ_{ic} es el coeficiente de correlación entre la variable X_i y la etiqueta c de la clase (C) y ρ_{ij} es el coeficiente de correlación entre X_i y X_j

Selección de Características

- Medidas basadas en teoría de información
 - La correlación sólo puede medir dependencias lineales. Un método bastante más potente es la información mutua $I(X_{1,...,M}; C)$

$$f(X_{1,...,M}) = I(X_{1,...,M}; C) = H(C) - H(C|X_{1,...,M}) =$$
$$\sum_{c=1}^{|C|} \int_{X_{1,...,M}} P(X_{1,...,M}, \omega_c) \log \frac{P(X_{1,...,M}, \omega_c)}{P(X_{1,...,M})P(\omega_c)} dX$$

donde H representa la entropía y ω_c la c-ésima etiqueta de la clase C

- La información mutua mide la cantidad de incertidumbre que disminuye en la clase C cuando se conocen los valores del vector $X_{1,...,M}$
- Por la complejidad de cálculo de I se suelen utilizar reglas heurísticas

$$f(X_{1,...,M}) = \sum_{i=1}^M I(X_i; C) - \beta \sum_{i=1}^M \sum_{j=i+1}^M I(X_i; X_j)$$

por ejemplo con $\beta=0.5$

Selección de Características

- Medidas de consistencia
 - Los tres grupos de medidas anteriores intentan encontrar las características que puedan, de forma maximal, predecir una clase frente al resto
 - Este enfoque no puede distinguir entre dos variables igualmente adecuadas, no detecta variables redundantes
 - Las medidas de consistencia tratan de encontrar el mínimo número de características que puedan separar las clases de la misma forma que lo hace el conjunto completo de variables

Selección de Características

Outlook	Temp.	Hum.	Windy	Class
0	1	1	0	0
0	1	1	1	0
1	1	1	0	1
2	0	1	0	1
2	0	0	0	1
2	0	0	1	0
1	0	0	1	1
0	1	1	0	0
0	0	0	0	1
2	1	0	0	1
0	1	0	1	1
1	1	1	1	1
1	1	0	0	1
2	0	1	1	0

Outlook: sunny(0),overcast(1),windy(2)
Temperature: <72(0),>=72(1)
Humidity: <85(0),>=85(1)
Windy: false(0), true(1)
Class: no(0), yes(1)

	Inf. Mutua	Dist. euclídea	Correlación
Outlook	0.074	0.357	0.176
Temp.	4.03E-4	0.143	-0.043
Hum.	0.045	0.463	-0.447
Windy	0.014	0.450	-0.258

Selección de Características

Ventajas

■ Envoltentes:

- Exactitud: generalmente son más exactos que los filtro, debido a la interacción entre el clasificador y el conjunto de datos de entrenamiento
- Capacidad para generalizar: poseen capacidad para evitar el sobreajuste debido a las técnicas de validación utilizadas

■ Filtro:

- Rápidos. Suelen limitarse a cálculos de frecuencias, mucho más rápido que entrenar un clasificador
- Generalidad. Al evaluar propiedades intrínsecas de los datos y no su interacción con un clasificador, sus resultados pueden ser utilizados por cualquier clasificador

Selección de Características

Inconvenientes

■ Envoltentes:

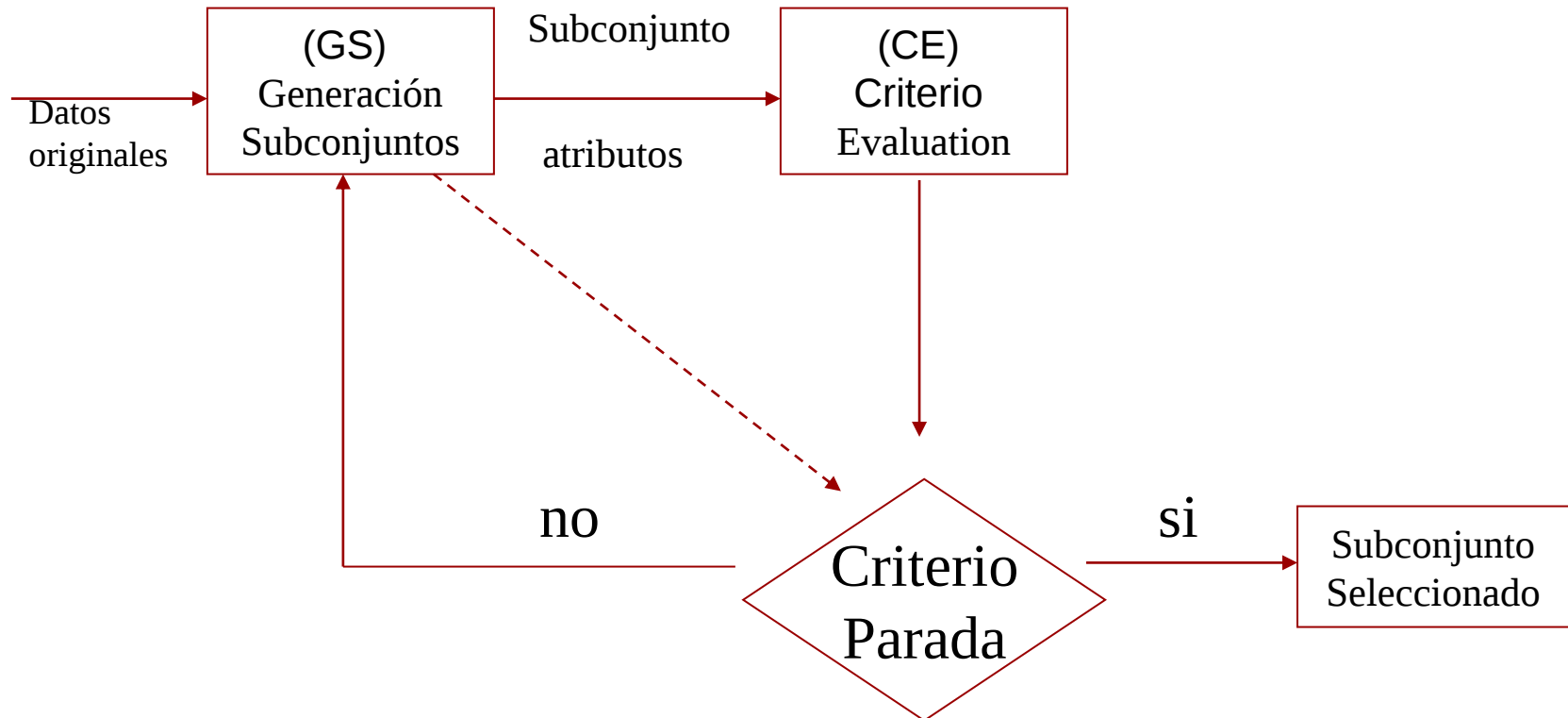
- Muy costosos: para cada evaluación hay que aprender un modelo y validarlo. No es factible para clasificadores costosos
- Pérdida de generalidad: La solución está sesgada hacia el clasificador utilizado

■ Filtros:

- Tendencia a incluir muchas variables. Normalmente se debe a las características monótonas de la función objetivo utilizada
 - El usuario deberá seleccionar un umbral

Selección de Características

Proceso



Selección de Características

Proceso

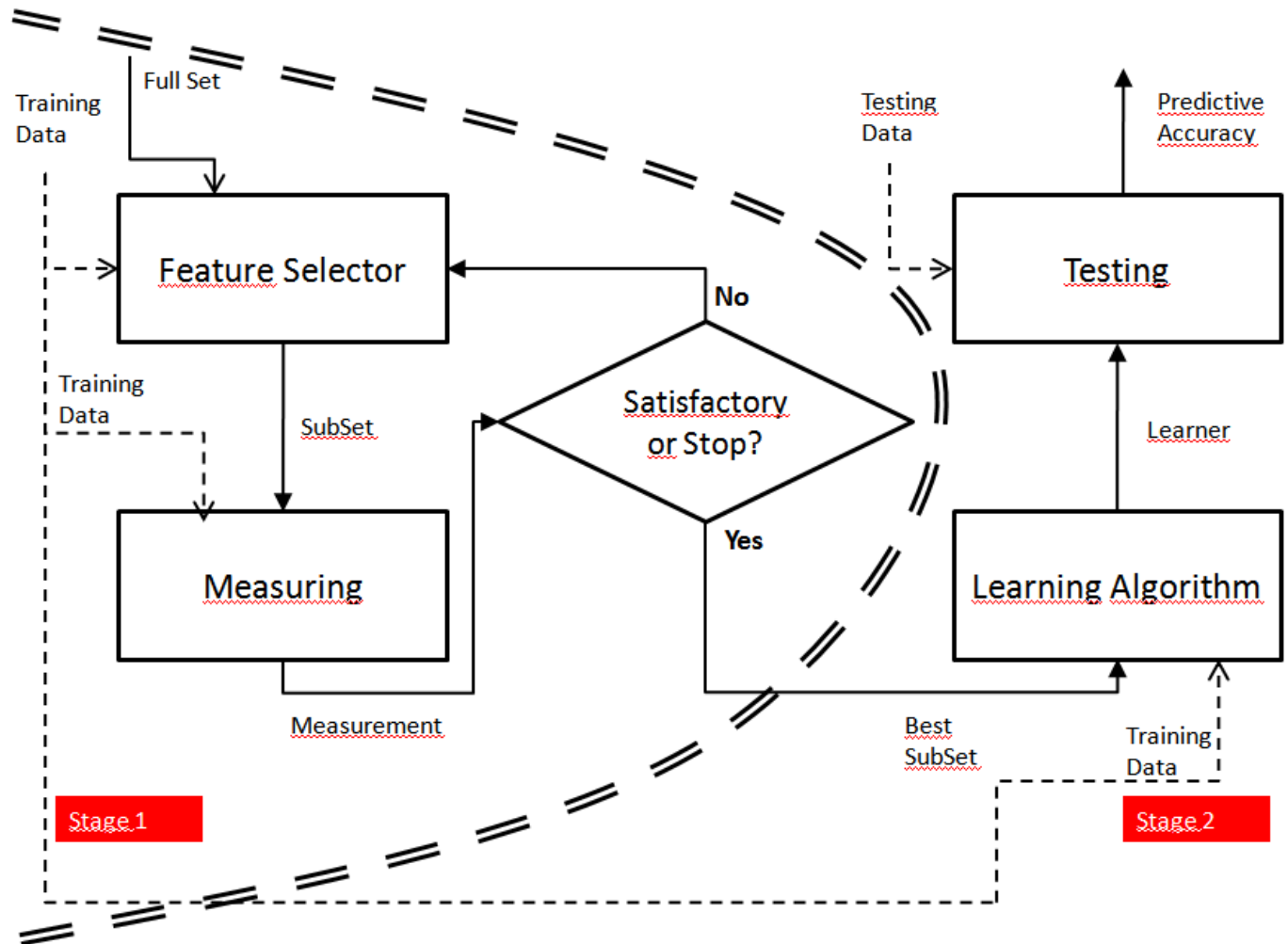


Fig. 7.2 A filter model for FS

Selección de Características

Proceso

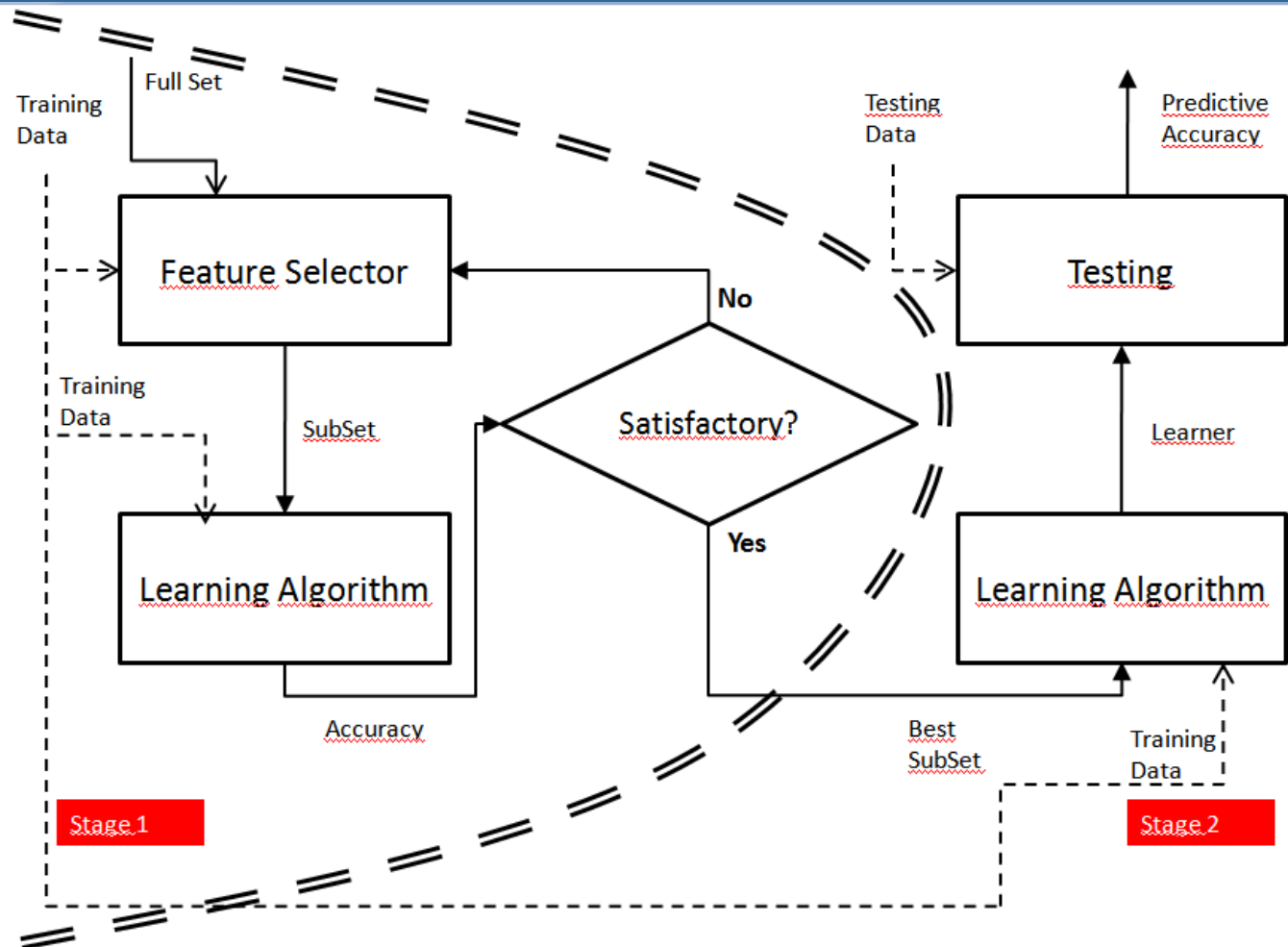


Fig. 7.2 A wrapper model for FS

Selección de Características

Distintas Clasificaciones

1. Según la evaluación:

filter

wrapper

2. Disponibilidad de la clase:

Supervisados

No supervisado

3. Según la búsqueda:

Completa $O(2^N)$

Heurística $O(N^2)$

Aleatoria ¿?

4. Según la salida del algoritmo:

Ranking

Subconjunto de atributos

Selección de Características

Algoritmos Subconjunto de Atributos

Devuelven un subconjunto de atributos optimizado según algún criterio de evaluación.

Entrada: x atributos - U criterio evaluación

Subconjunto = $\{\}$

Repetir

$S_k = \text{generarSubconjunto}(x)$

si existeMejora(S, S_k, U)

Subconjunto = S_k

Hasta CriterioParada()

Salida: Lista, atts más relevantes al principio

Selección de Características

Algoritmos de Ranking

Devuelven una lista de atributos ordenados según algún criterio de evaluación.

Entrada: x atributos - U criterio evaluación

Lista = {}

Para cada Atributo x_i , $i \in \{1, \dots, N\}$

$v_i = \text{calcular}(x_i, U)$

situar x_i dentro de Lista conforme v_i

Salida: Lista, atts más relevantes al principio

Selección de Características

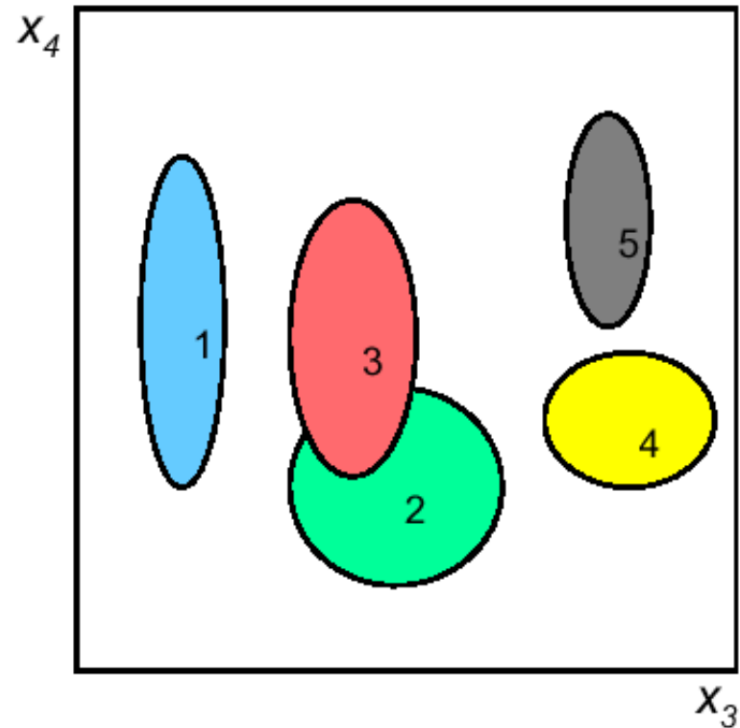
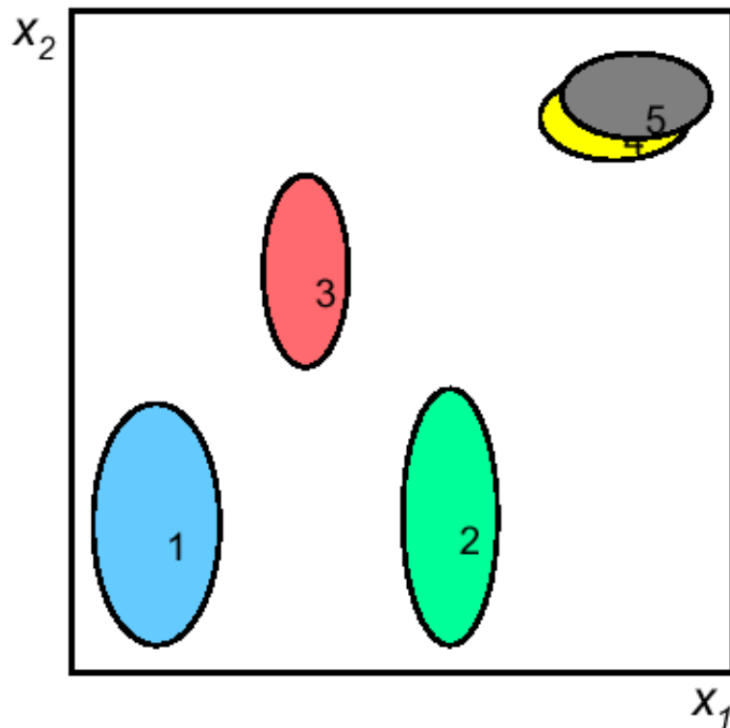
Validación algoritmos ranking

Atributos	A1	A2	A3	A4	A5	A6	A7	A8	A9
Ranking	A5	A7	A4	A3	A1	A8	A6	A2	A9
	A5	A7	A4	A3	A1	A8	(6 atributos)		

Selección de Características

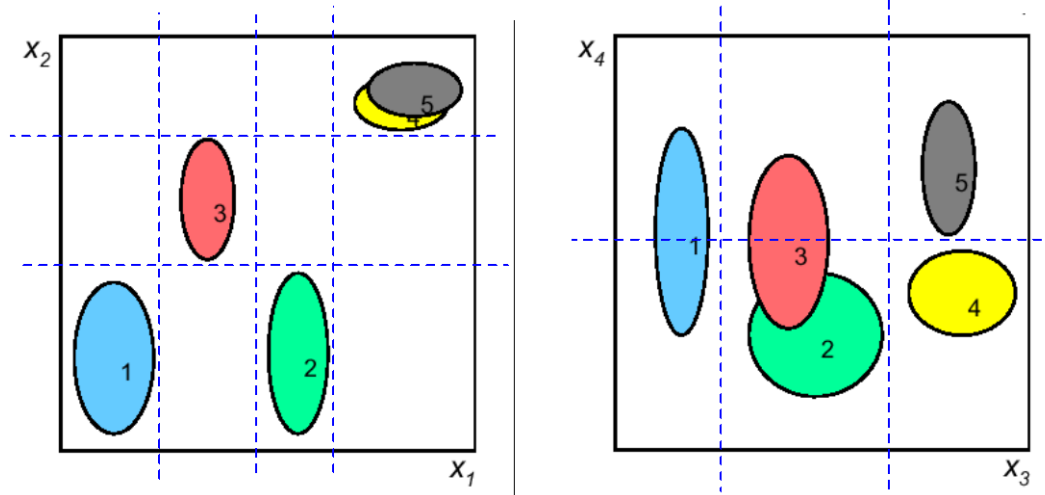
Algunos algoritmos.

¿Qué subconjunto de 2 variables seleccionamos?



Selección de Características

Algunos algoritmos.



X1: [1,2,3,{4,5}]

X2: [{1,2},3,{4,5}]

X3: [1,{2,3},{4,5}]

X4: [{1,2,3},4,5]

Es razonable que la función objetivo ofrezca un resultado como

$$f(X1) > f(X2) \approx f(X3) > f(X4)$$

→ Elegir {X1,X2} o {X1,X3}

Parece razonable que se elija {X1,X4}

Selección de Características

Algunos algoritmos.

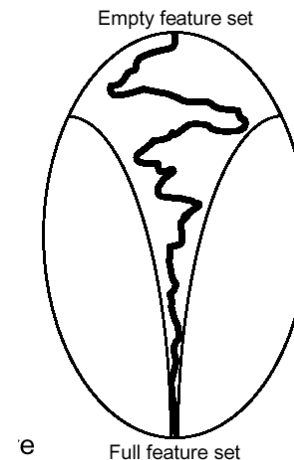
Selección hacia delante

La selección *forward* comienza con el conjunto vacío y de forma secuencial añade al subconjunto actual S el atributo X_i que maximiza $f(S, X_i)$

1. Comenzar con $S = \emptyset$
2. Seleccionar la variable

$$X^+ = \arg \max_{X \in U - S} f(S \cup X)$$

3. $S = S \cup \{X^+\}$
4. Ir al paso 2



Selección de Características

Algunos algoritmos.

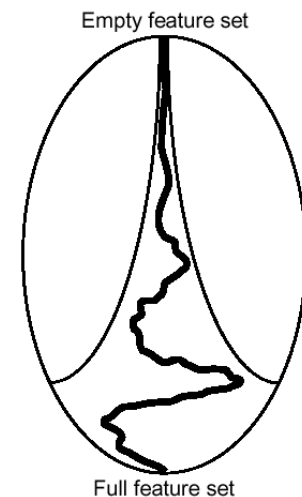
Selección hacia atrás

- La selección *backward* comienza con el conjunto completo U y de forma secuencial elimina del subconjunto actual S el atributo X que decrementa menos $f(S-X)$

1. Comenzar con $S=U$
2. Seleccionar la variable X^-

$$X^- = \arg \max_{X \in S} f(S - X)$$

3. $S = S - \{X^-\}$
4. Ir al paso 2



Selección de Características

Algunos algoritmos.

- Selección hacia delante:
 - Funciona mejor cuando el subconjunto óptimo tiene pocas variables
 - Es incapaz de eliminar variables
- Selección hacia atrás:
 - Funciona mejor cuando el subconjunto óptimo tiene muchas variables
 - El principal inconveniente es el de reevaluar la utilidad de algunos atributos previamente descartados
- Especialmente con el enfoque envolvente, ¿cuál sería computacionalmente más eficiente?

Selección de Características

Algunos algoritmos.

Selección l-más r-menos

- Es una generalización de forward y backward

1. Si $l > r$ entonces $S = \emptyset$
si no, $S = U$ e ir al paso 3

2. Repetir l veces

$$X^+ = \arg \max_{X \in U - S} f(S \cup X)$$

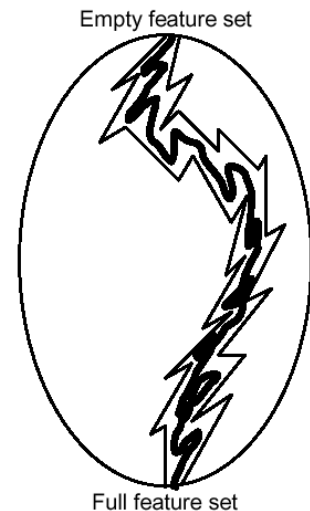
$$S = S \cup \{X^+\}$$

3. Repetir r veces

$$X^- = \arg \max_{X \in S} f(S - X)$$

$$S = S - \{X^-\}$$

4. Ir al paso 2



Selección de Características

Algunos algoritmos.

Selección bidireccional

- Es una implementación paralela de forward y backward
- Hay que asegurar que los atributos eliminados por backward no son introducidos por forward (y viceversa)

1. Comenzar *forward* con $S_F = \emptyset$
2. Comenzar *backward* con $S_B = U$
3. Seleccionar

$$X^+ = \arg \max_{X \in S_B - S_F} f(S_F \cup X)$$

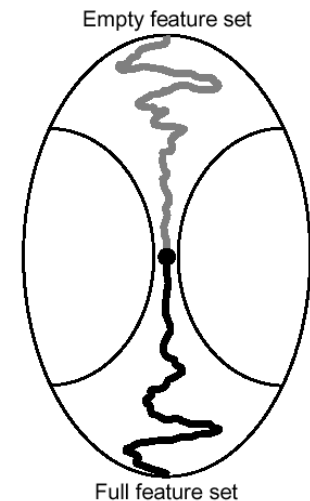
$$S_F = S_F \cup \{X^+\}$$

4. Seleccionar

$$X^- = \arg \max_{X \in S_B - S_F} f(S_B - X)$$

$$S_B = S_B - \{X^-\}$$

5. Ir al paso 3



Selección de Características

Algunos algoritmos. Selección flotante

- Extensión de l -más r -menos para evitar fijar el l y r a priori
- Hay dos métodos: uno comienza por el conjunto vacío y otro por el total

1. Comenzar con $S = \emptyset$

2. Seleccionar

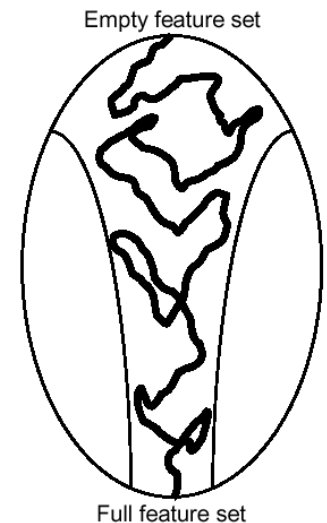
$$X^+ = \arg \max_{X \in U - S} f(S \cup X)$$

$$S = S \cup \{X^+\}$$

3. Seleccionar

$$X^- = \arg \max_{X \in S} f(S - X)$$

4. Si $f(S - X^-) > f(S)$
entonces $S = S - \{X^-\}$ e ir al paso 3
si no ir al paso 2

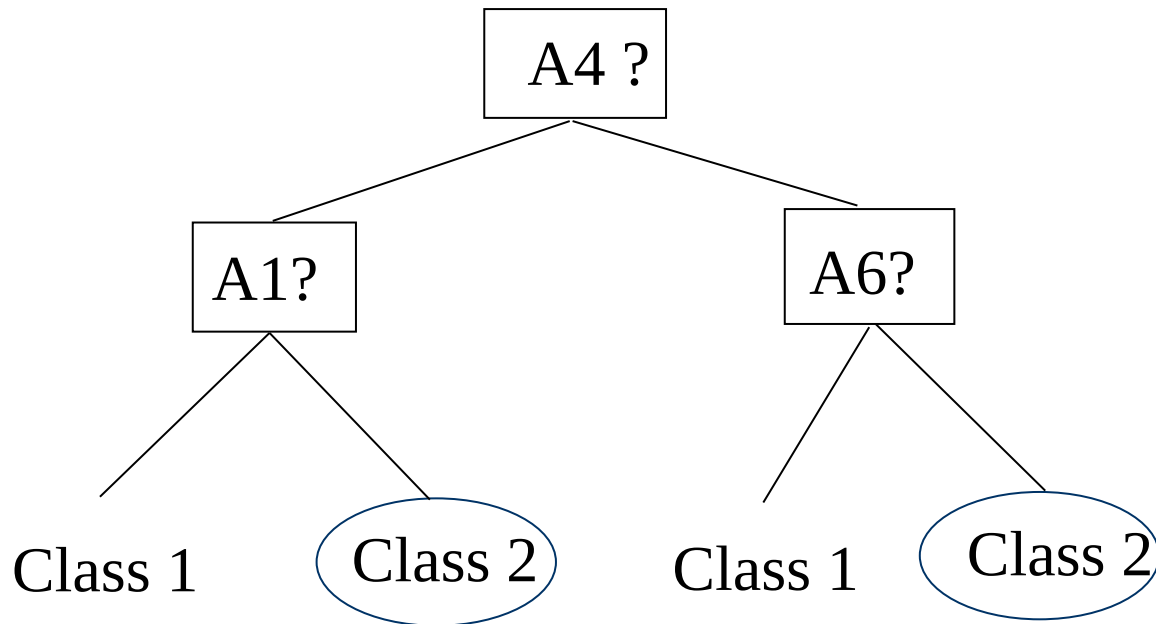


Selección de Características

Algunos algoritmos.

Selección de características con árboles de decisión

Conjunto inicial de atributos: {A1, A2, A3, A4, A5, A6}



Características seleccionadas: {A1,A4,A6}

Selección de Características

Algunos algoritmos relevantes:

- **Algoritmos secuenciales.** Añaden o eliminan variables al subconjunto candidato de forma secuencial. Suelen quedarse en óptimos locales
 - Selección hacia delante, selección hacia atrás, selección más-menos-r, búsqueda bidireccional, selección secuencial flotante
- **Algoritmos exponenciales.** El número de subconjuntos evaluados aumenta exponencialmente con la dimensionalidad del espacio de búsqueda
 - Branch and bound, beam search
- **Algoritmos estocásticos.** Utilizan aleatoriedad para escapar de óptimos locales
 - Ascensión de colinas con reinicios, enfriamiento estocástico, algoritmos genéticos, enfriamiento simulado

Selección de Características

Algunos algoritmos relevantes:

Table 7.3 All possible combinations for FS algorithms

Search direction	Evaluation measure	Search strategy		
		Exhaustive	Heuristic	Nondeterministic
Forward	Probability	C1	C7	–
	Consistency	C2	C8	–
	Accuracy	C3	C9	–
Backward	Probability	C4	C10	–
	Consistency	C5	C11	–
	Accuracy	C6	C12	–
Random	Probability	–	C13	C16
	Consistency	–	C14	C17
	Accuracy	–	C15	C18

Selección de Características

Algunos algoritmos relevantes:

- **Focus algorithm.** Consistency measure for forward search,
- **Mutual Information based Features Selection (MIFS).**
- **Las Vegas Filter (LVF)**
- **Las Vegas Wrapper (LVW)**
- **Relief Algorithm**
- **mRMR: Minimum Redundancy Maximum Relevance**

Selección de Características

Algorithm 6 LVF algorithm.

```
function LVF( $D$  - a data set with  $M$  features,  $U$  - the inconsistency rate,  $maxTries$  -  
stopping criterion,  $\gamma$  - an allowed inconsistency rate)  
  initialize: list  $L = \{\}$  ▷  $L$  stores equally good sets  
   $C_{best} = M$   
  for  $maxTries$  iterations do  
     $S = \text{RANDOMSET}(\text{seed})$   
     $C = \#(S)$  ▷  $\#$  - the cardinality of  $S$   
    if  $C < C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then  
       $S_{best} = S$   
       $C_{best} = C$   
       $L = \{S\}$  ▷  $L$  is reinitialized  
    else if  $C = C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then  
       $L = \text{APPEND}(S, L)$   
    end if  
  end for  
  return  $L$  ▷ all equivalently good subsets found by LVF  
end function
```

Selección de Características

Algorithm 7 LVW algorithm.

```
function LVW( $D$  - a data set with  $M$  features,  $LA$  - a learning algorithm,  $maxTries$  -  
stopping criterion,  $F$  - a full set of features)  
  initialize: list  $L = \{\}$  ▷  $L$  stores sets with equal accuracy  
   $A_{best} = \text{ESTIMATE}(D, F, LA)$   
  for  $maxTries$  iterations do  
     $S = \text{RANDOMSET}(\text{seed})$   
     $A = \text{ESTIMATE}(D, S, LA)$  ▷  $\#$  - the cardinality of  $S$   
    if  $A > A_{best}$  then  
       $S_{best} = S$   
       $A_{best} = A$   
       $L = \{S\}$  ▷  $L$  is reinitialized  
    else if  $A = A_{best}$  then  
       $L = \text{APPEND}(S, L)$   
    end if  
  end for  
  return  $L$  ▷ all equivalently good subsets found by LVW  
end function
```

Selección de Características

Algorithm 8 MIFS algorithm.

```
function MIFS( $F$  - all features in data,  $S$  - set of selected features,  $k$  - desired size of  $S$ ,  
 $\beta$  - regulator parameter)  
  initialize:  $S = \{\}$   
  for each feature  $f_i$  in  $F$  do  
    Compute  $I(C, f_i)$   
  end for  
  Find  $f_{max}$  that maximizes  $I(C, f)$   
   $F = F - \{f_{max}\}$   
   $S = S \cup f_{max}$   
  repeat  
    for all couples of features ( $f_i \in F, s_j \in S$ ) do  
      Compute  $I(f_i, s_j)$   
    end for  
    Find  $f_{max}$  that maximizes  $I(C, f) - \beta \sum_{s \in S} I(f_i, s_j)$   
     $F = F - \{f_{max}\}$   
     $S = S \cup f_{max}$   
  until  $|S| = k$   
  return  $S$   
end function
```

Selección de Características

Extracción de características

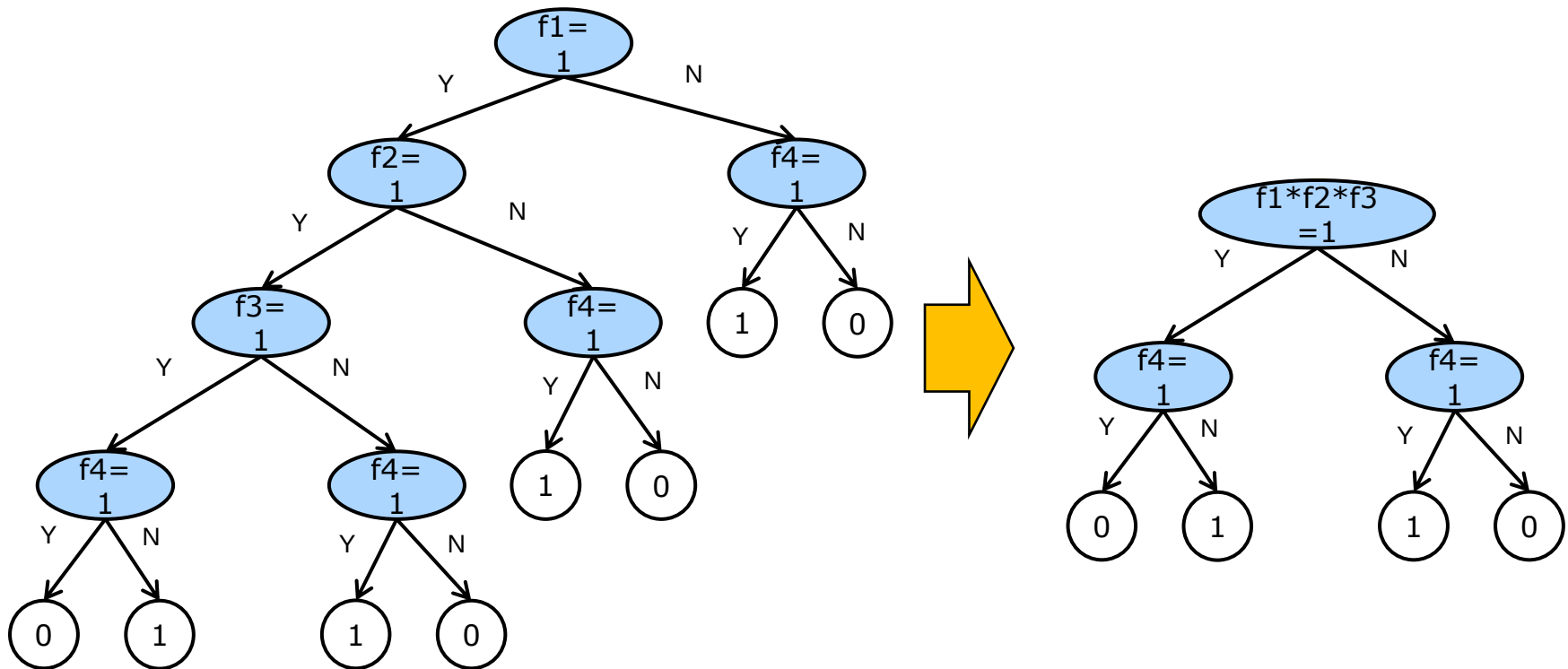
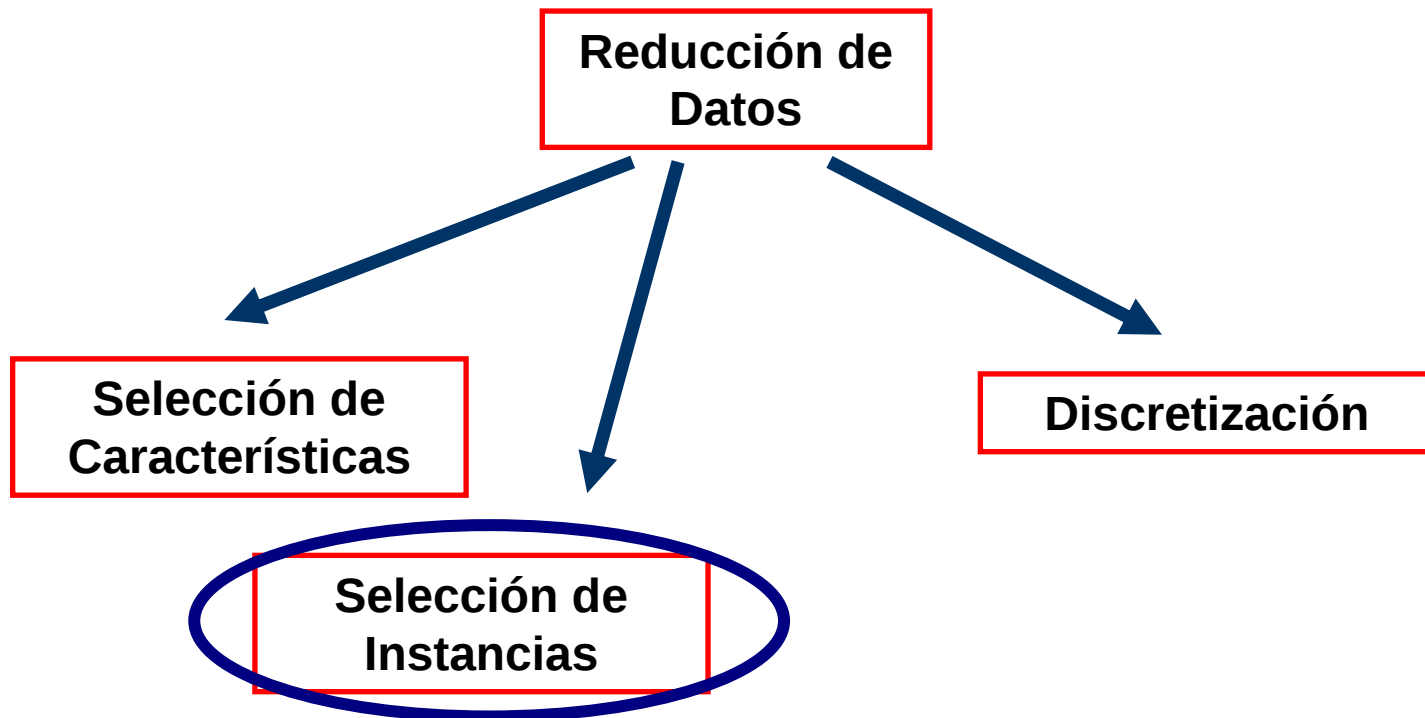


Fig. 7.4 The effect of using the product of features in decision tree modeling

Reducción de Datos



Bibliografía:

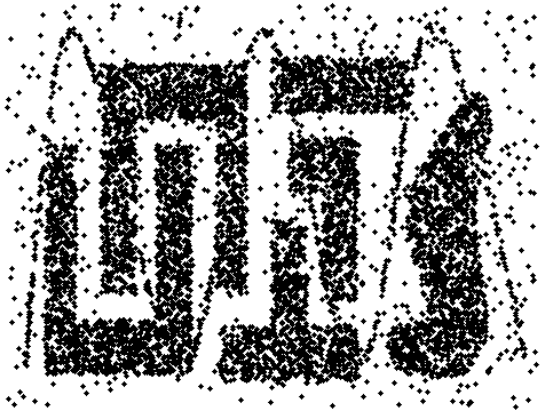
T. Reinartz. A Unifying View on Instance Selection.
Data Mining and Knowledge Discovery 6, 191-210, 2002.

Selección de Instancias

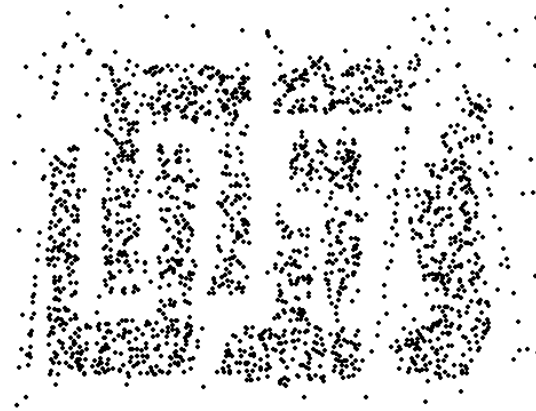
- ✿ La SI pretende elegir los ejemplos que sean relevantes para una aplicación y lograr el máximo rendimiento. El resultado de la SC sería:
 - ❖ Menos datos → los algoritmos pueden aprender más rápidamente
 - ❖ Mayor exactitud → el clasificador generaliza mejor
 - ❖ Resultados más simples → más fácil de entender
- ✿ SI y Transformación (compactación/agrupamiento)

Selección de Instancias

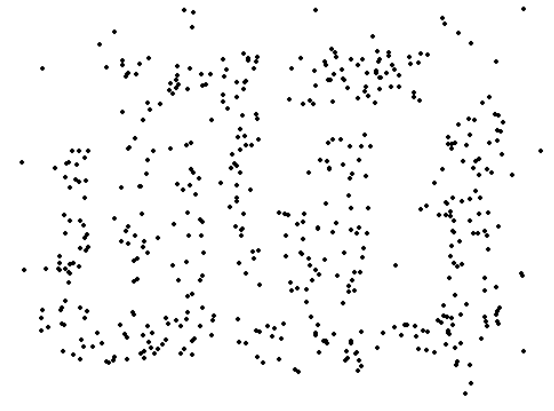
Ejemplos de diferentes tamaños



8000 puntos

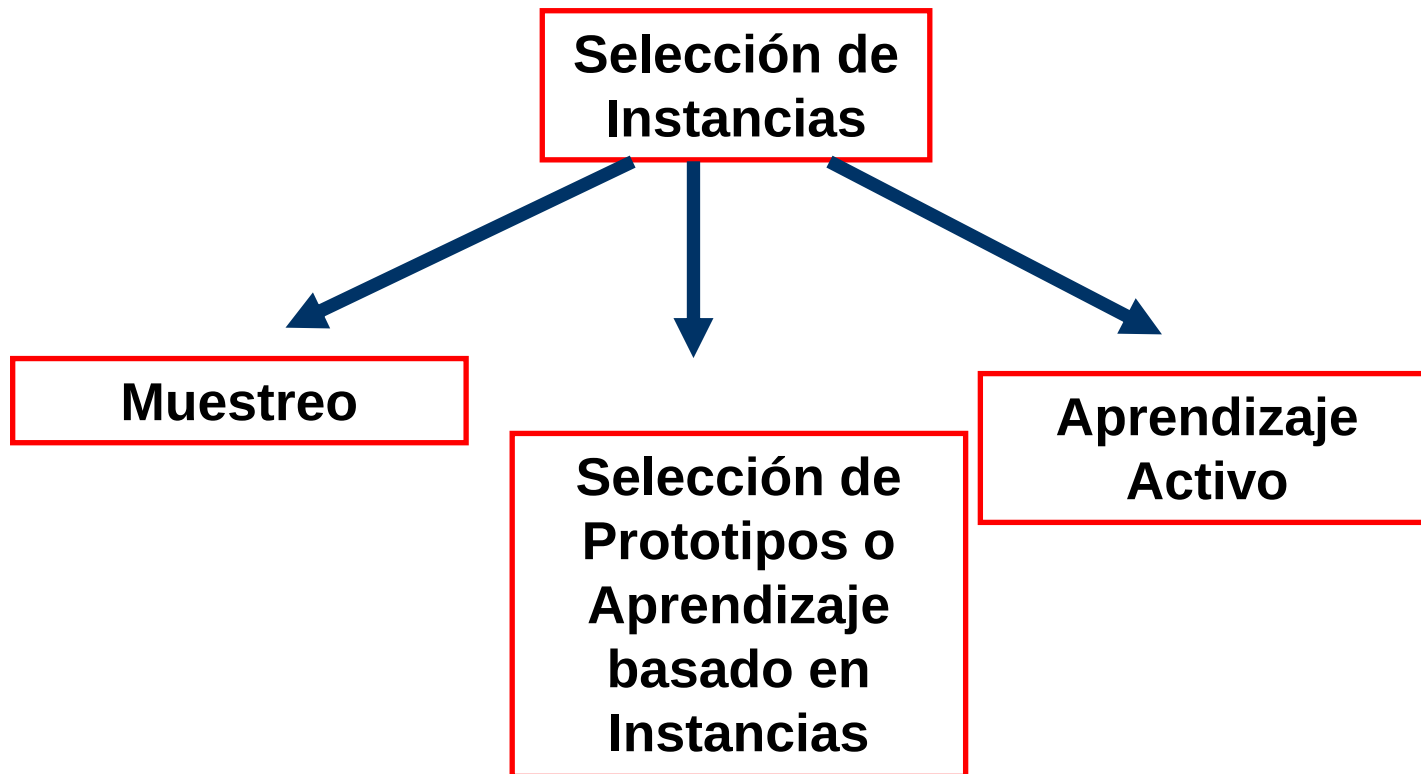


2000 puntos

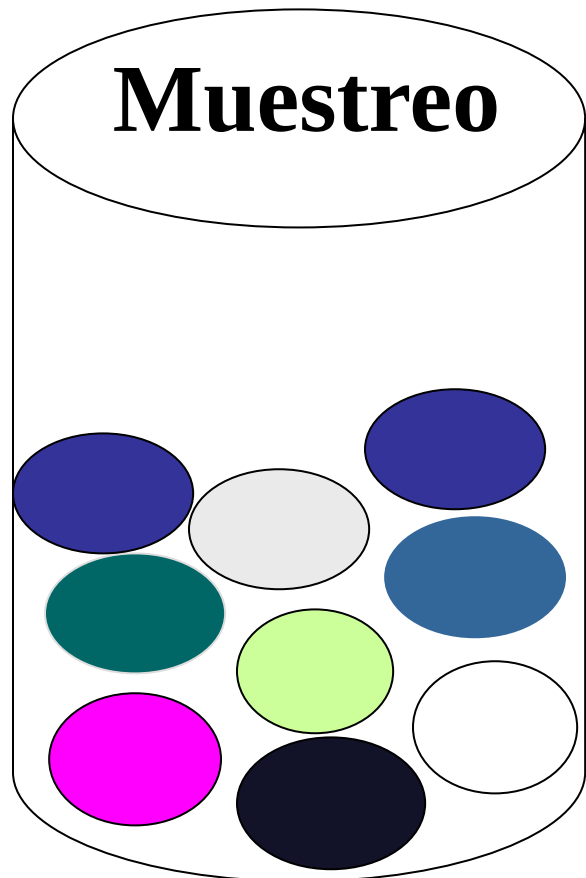


500 puntos

Selección de Instancias

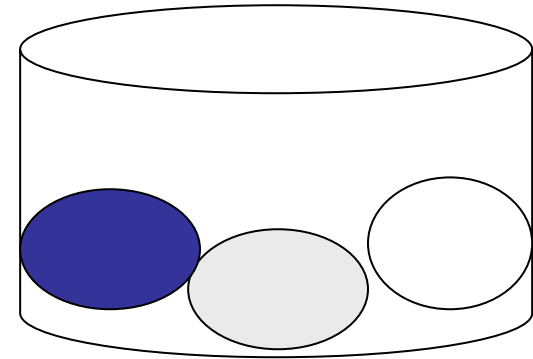


Selección de Instancias

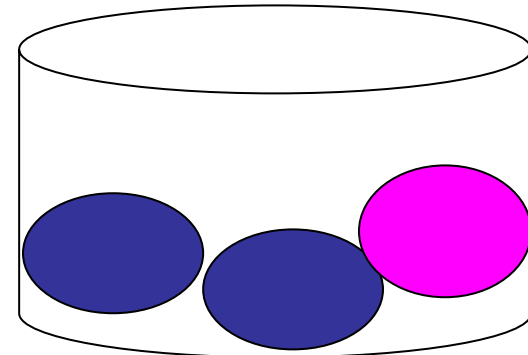


Datos sin refinar

SRSWOR
(muestreo
Simple sin
Reemplazamiento)



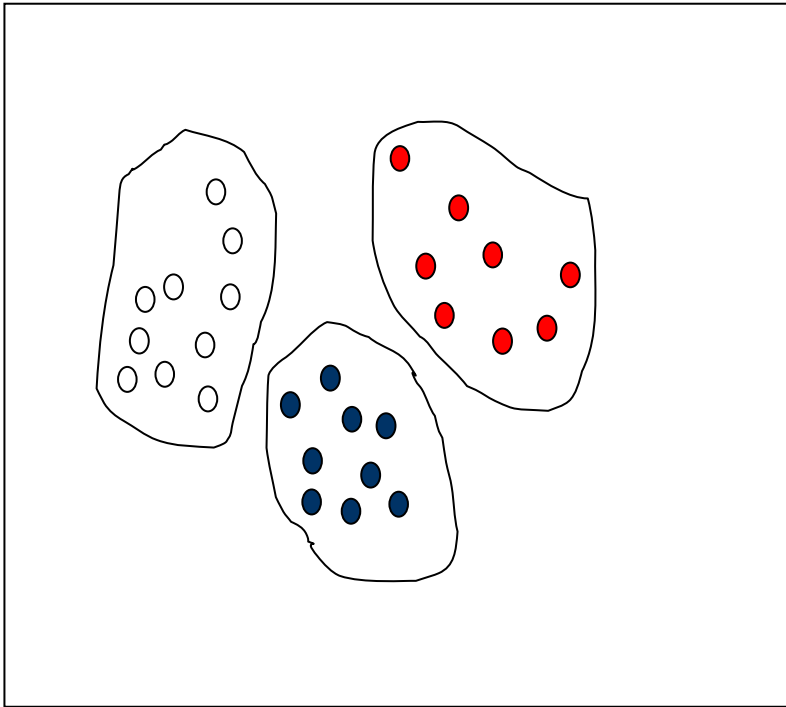
SRSWR



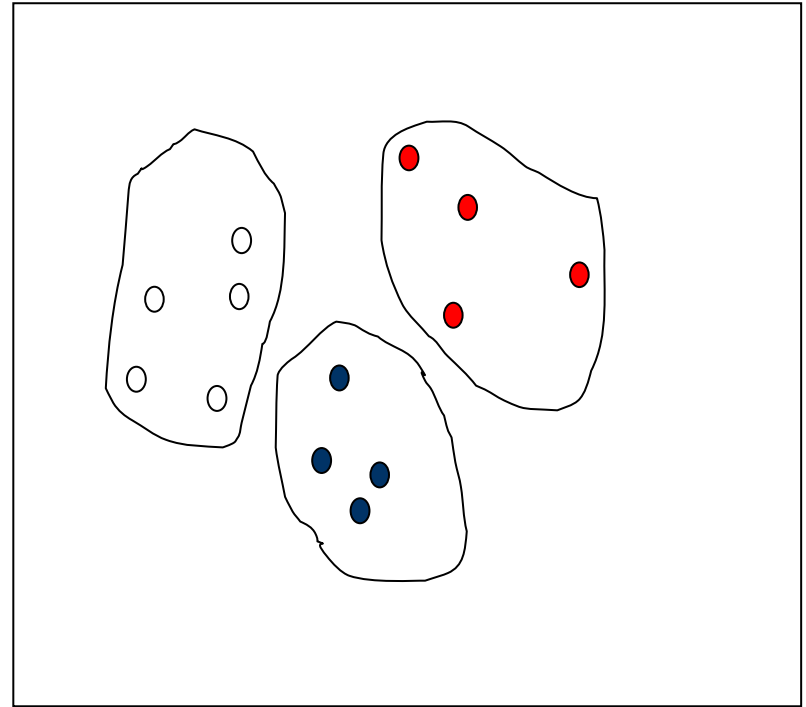
Selección de Instancias

Muestreo

Datos sin refinar



Reducción simple



Selección de Instancias

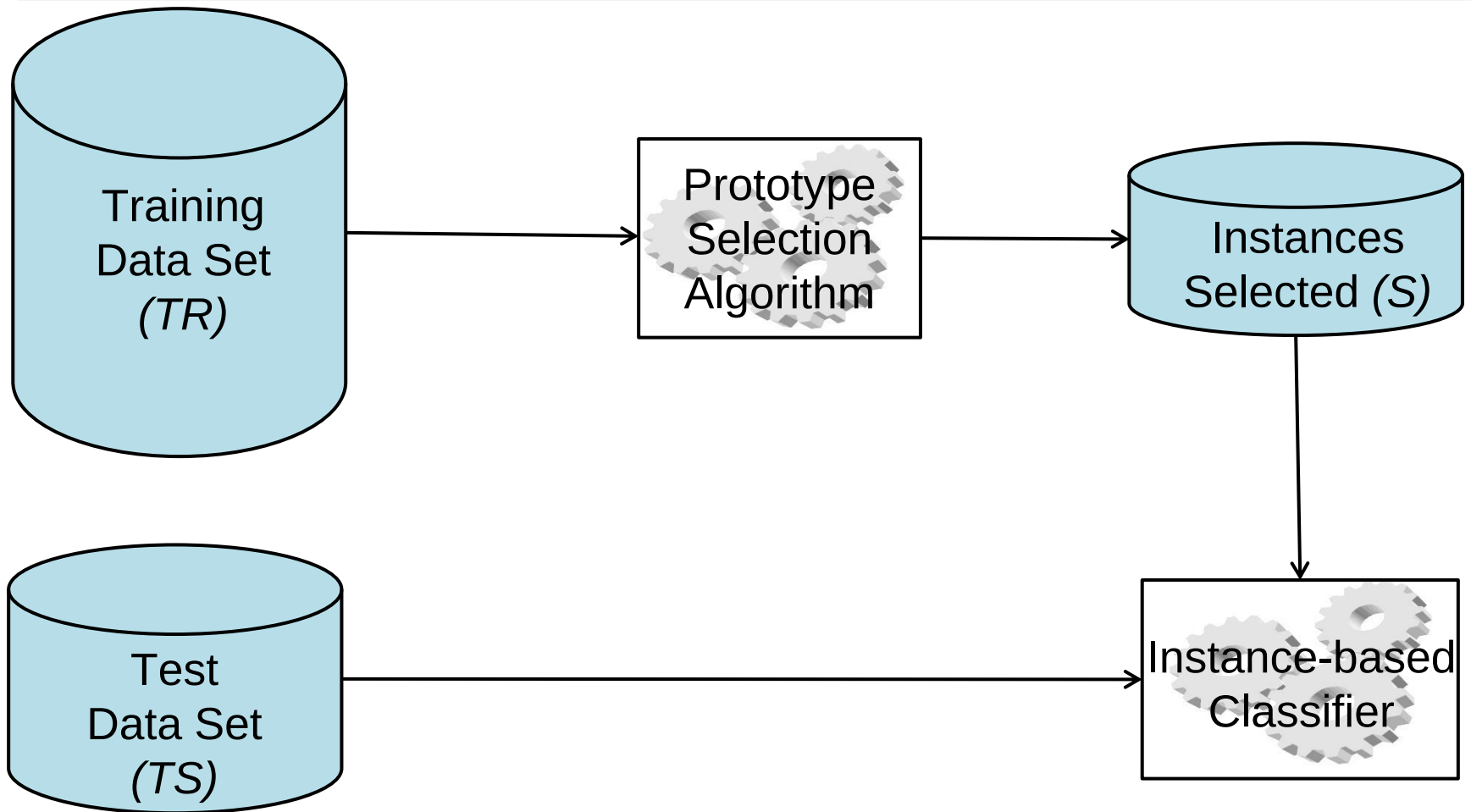
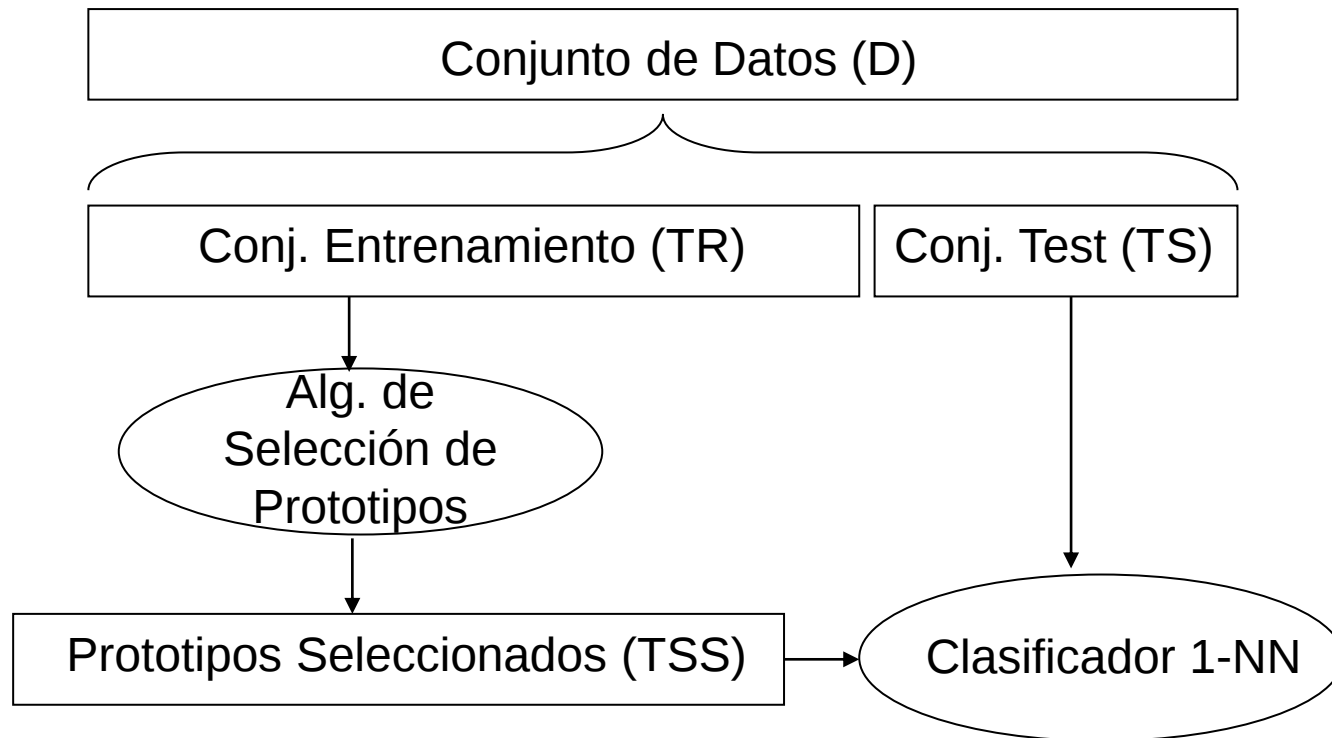


Fig. 8.1 PS process

Selección de Instancias

Selección de Prototipos para Clasificación con 1-NN



Selección de Instancias

Selección de Prototipos

Propiedades:

- Dirección de la búsqueda: Incremental, decremental, por lotes, mezclada y fijada.
- Tipo de selección: Condensación, Edición, Híbrido.
- Tipo de evaluación: Filtrada o envolvente.

Selección de Instancias

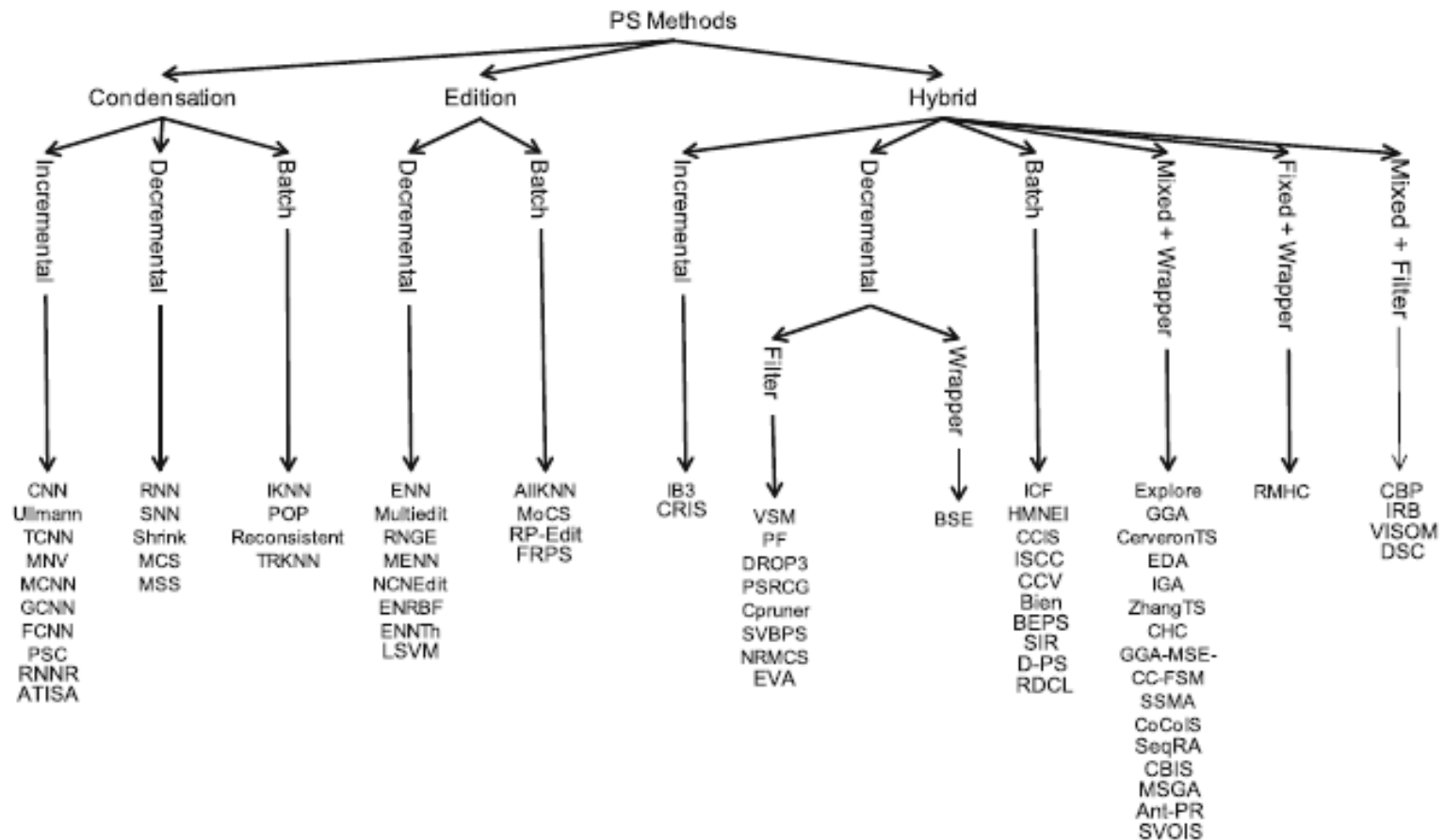


Fig. 8.3 PS taxonomy

Selección de Instancias

Selección de Prototipos o Aprendizaje basado en Instancias

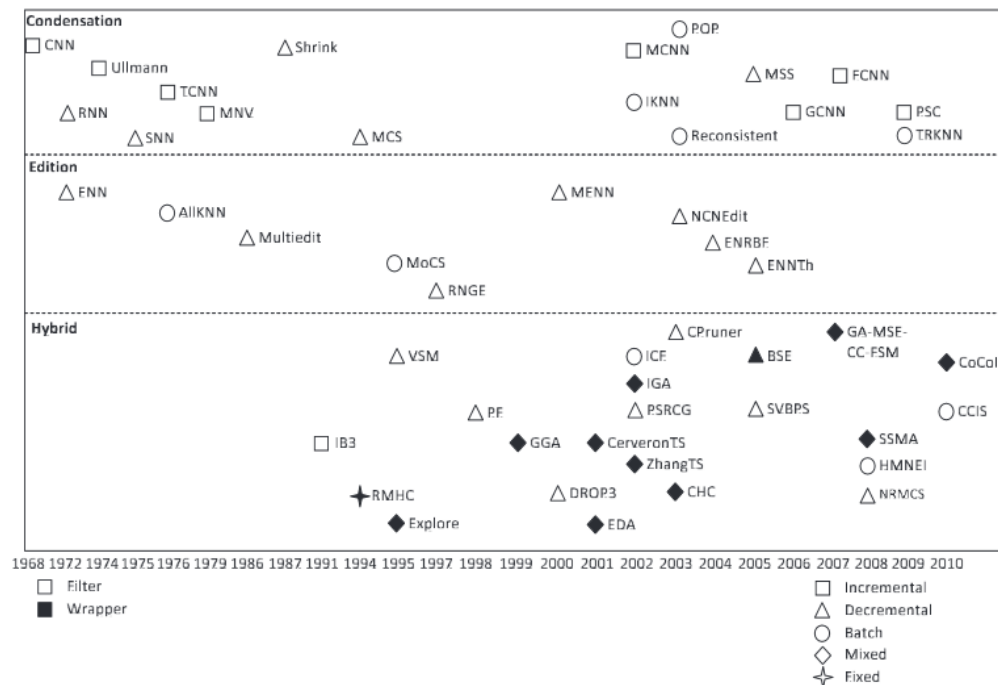


Fig. 2. Prototype selection map.

Ref. S. García, [J. Derrac](#), J.R. Cano and [F. Herrera](#), **Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study**, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34:3 (2012) 417-435 [doi: 10.1109/TPAMI.2011.142](#)

Selección de Instancias

Formas de evaluar un algoritmo de Selección de instancias en k-NN:

- Reducción del espacio de almacenamiento
- Tolerancia al ruido
- Precisión en la generalización del aprendizaje
- Requerimientos de cómputo

Selección de Instancias

Un par de algoritmos clásicos:

- Algoritmo clásico de Condensación: Condensed Nearest Neighbor (CNN)
 - Incremental
 - Inserta solo las instancias mal clasificadas a partir de una selección aleatoria de una instancia de cada clase.
 - Dependiente del orden de presentación
 - Solo retiene puntos pertenecientes al borde

Algorithm 10 CNN algorithm.

```
function CNN( $T$  - training data)
  initialize:  $S = \emptyset$ 
  repeat
    for all  $x \in T$  (in random order) do
      Find  $x' \in S$  s.t.  $\|x - x'\| = \min_{x' \in S} \|x - x'\|$ 
      if  $class(x) \neq class(x')$  then
         $S = S \cup \{x\}$ 
      end if
    end for
  until  $S$  does not change
  return  $S$ 
end function
```

Selección de Instancias

Un par de algoritmos clásicos:

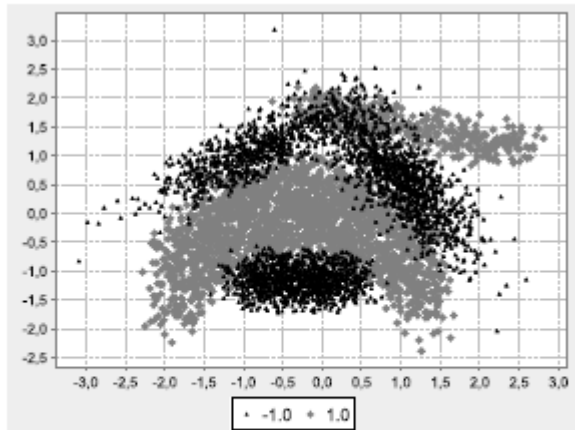
- Algoritmo clásico de Edición: Edited Nearest Neighbor (ENN)
 - Por lotes
 - Se eliminan aquellas instancias que se clasifican incorrectamente usando sus k vecinos más cercanos ($K = 3, 5$ ó 9).
 - “Suaviza” las fronteras, pero retiene el resto de puntos (muchos redundantes)

Algorithm 11 ENN algorithm.

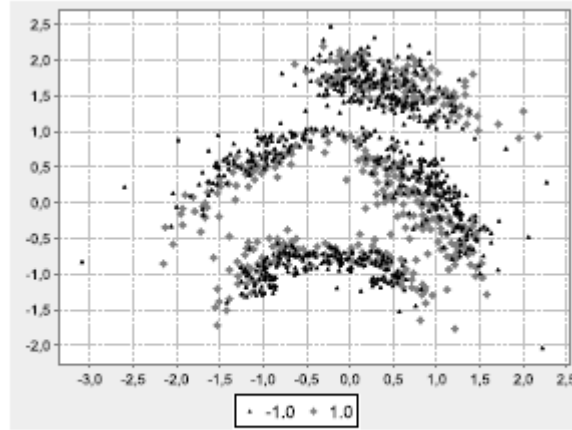
```
function ENN( $T$  - training data,  $k$  - number of nearest neighbor)
  initialize:  $S = T$ 
  for all  $x \in S$  do
     $X' = \emptyset$ 
    for  $i = 1$  to  $k$  do
      Find  $x'_i \in T$  s.t.  $x \neq x'_i$  and  $\|x - x'_i\| = \min_{x' \in (T \setminus X')} \|x - x'\|$ 
       $X' = X' \cup \{x'_i\}$ 
    end for
    if  $\text{class}(x) \neq \text{majorityClass}(X')$  then
       $S = S \setminus \{x\}$ 
    end if
  end for
  return  $S$ 
end function
```

Selección de Instancias

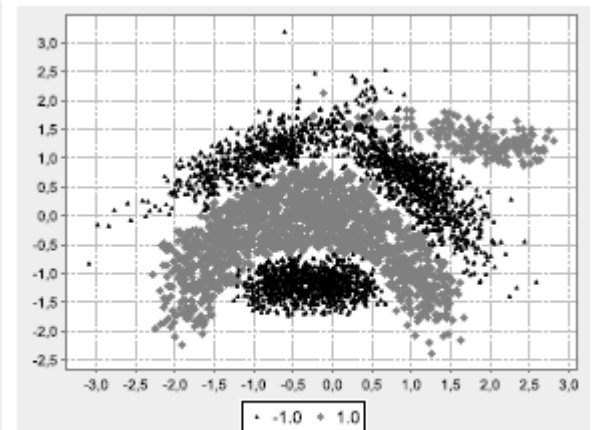
Ejemplos gráficos:



(a) Banana Original
(0.8751, 0.7476)



(b) CNN (0.7729, 0.8664, 0.7304)

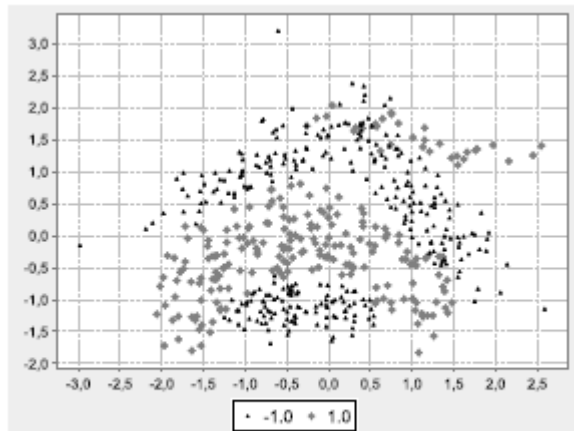


(h) AllKNN (0.1758, 0.8934, 0.7831)

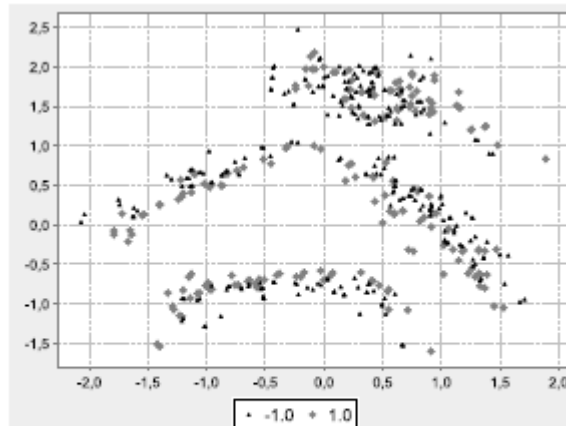
Conjunto banana con 5.300 instancias y dos clases. Conjunto obtenido por CNN y AllKNN (aplicación iterativa de ENN con $k=3$, 5 y 7).

Selección de Instancias

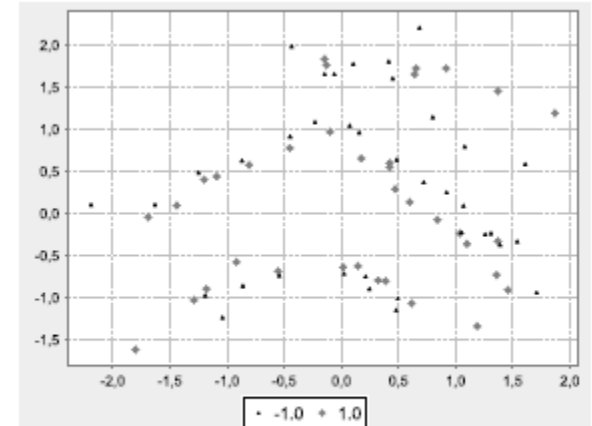
Ejemplos gráficos:



(k) RMHC (0.9000, 0.8972, 0.7915)



(e) DROP3
(0.9151, 0.8696, 0.7356)



(l) SSMA (0.9879, 0.8964, 0.7900)

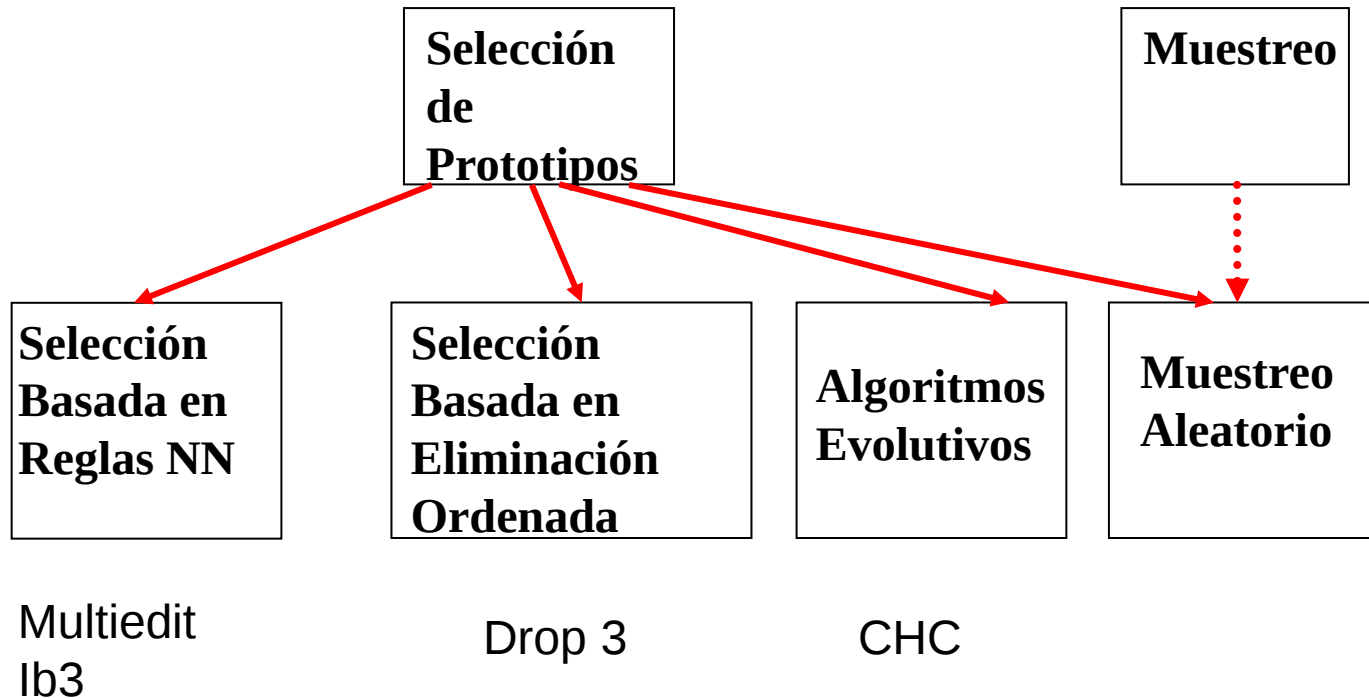
RMHC es una técnica de muestreo adaptativo basa en búsquedas locales con un tamaño final fijo.

DROP3 es la técnica híbrida más conocida y utilizada para NN.

SSMA es una aproximación evolutiva basada en algoritmo meméticos.

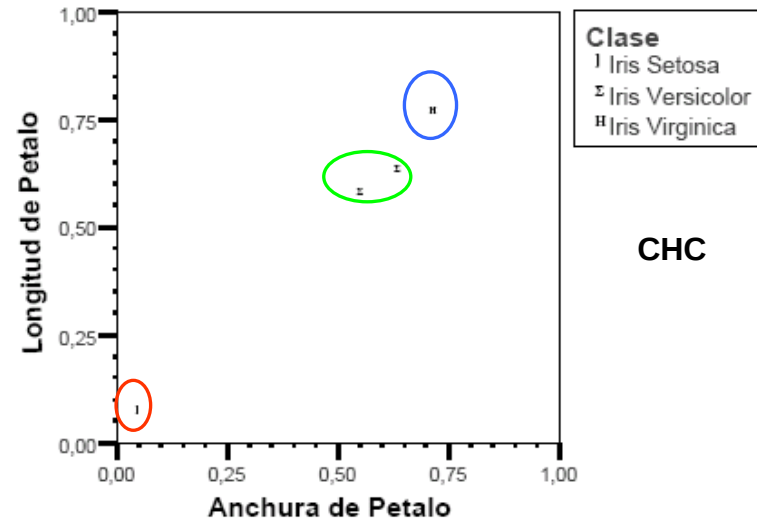
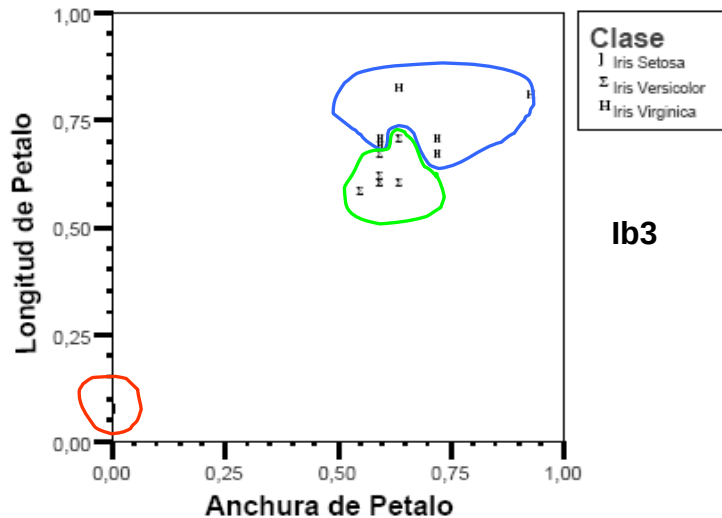
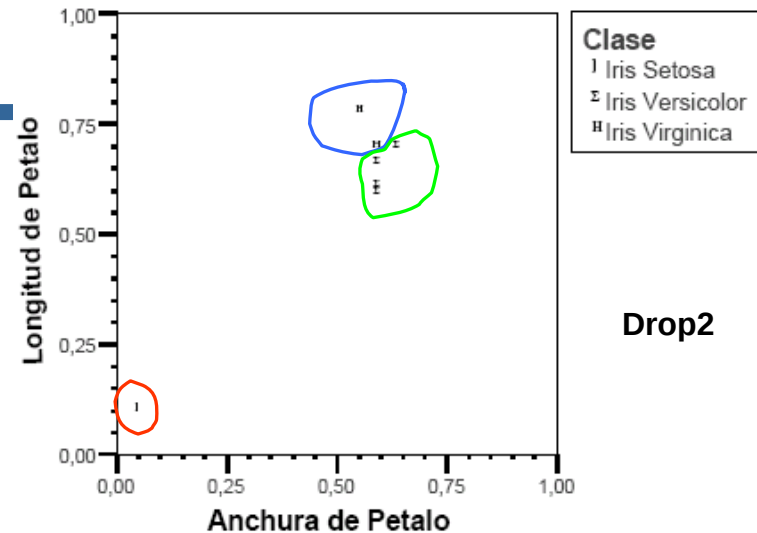
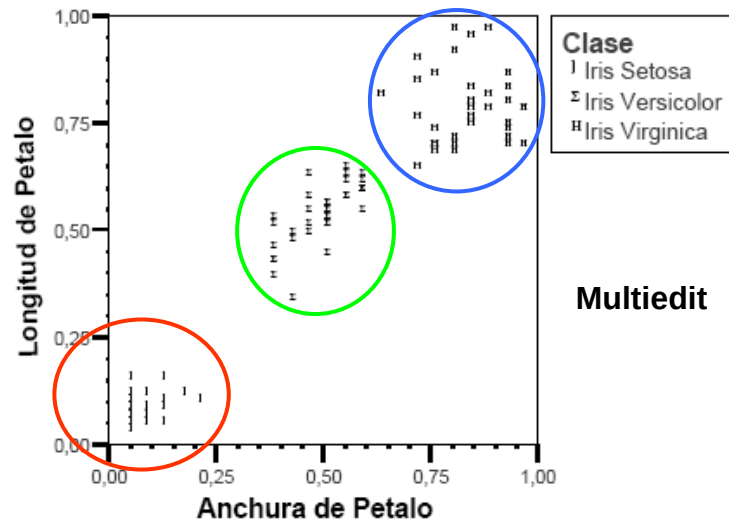
Selección de Instancias

Ejemplos gráficos:



Selección de Instancias

Ejemplos gráficos:



Bibliografía: J.R. Cano, F. Herrera, M. Lozano. Using Evolutionary Algorithms as Instance Selection for Data Reduction in KDD: An Experimental Study. IEEE Trans. on Evolutionary Computation 7:6 (2003) 561-575.

Selección de Instancias

Selección de Instancias. Eficiencia

El orden de los algoritmos es superior a $O(n^2)$ y suele estar en orden $O(n^3)$.

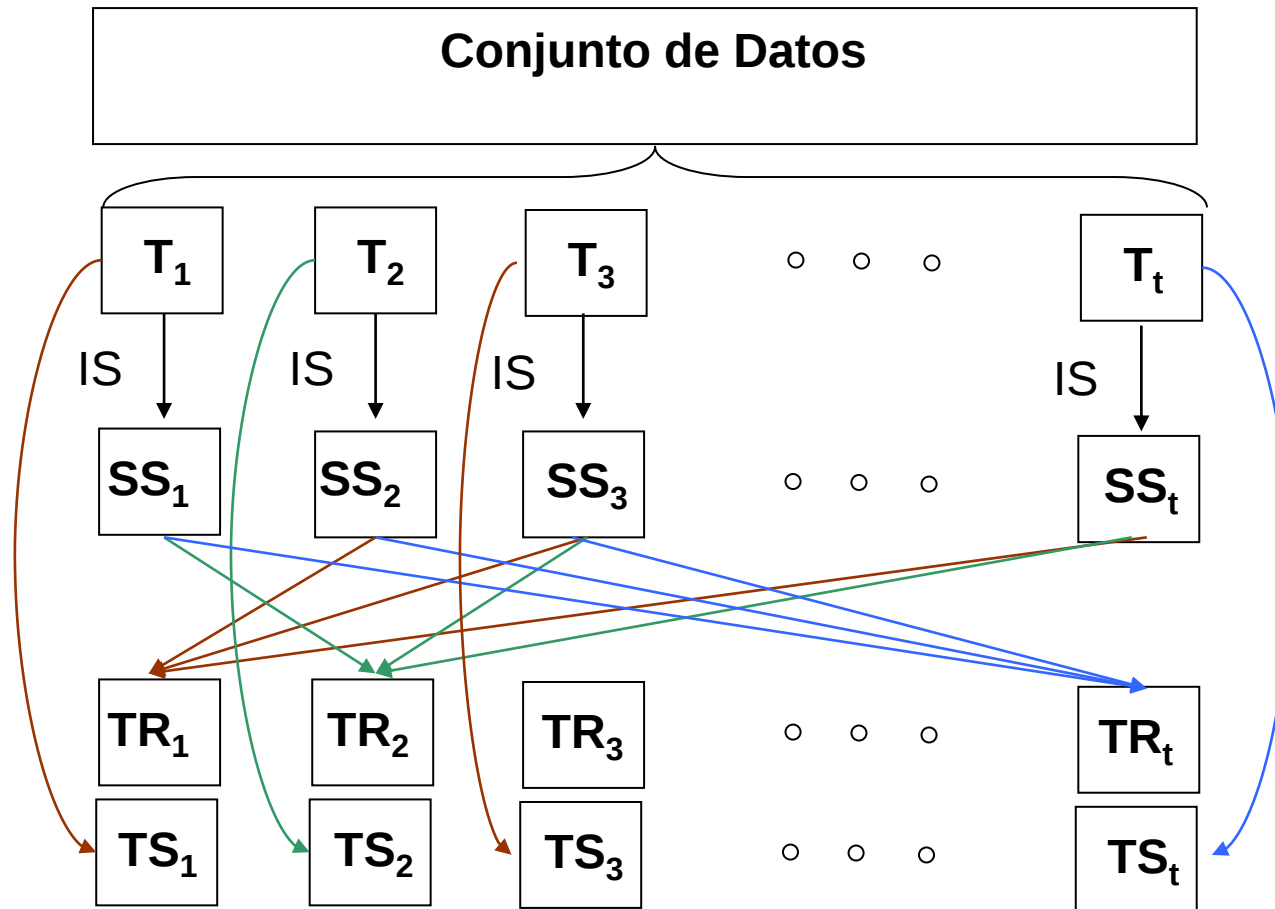
Las principales dificultades que deben afrontar los algoritmos de Selección de Prototipos son: Eficiencia, recursos, generalización, representación.

¿Cómo realizar la selección de instancias con grandes bases de datos?

Combinamos una estrategia de estratificación con los algoritmos de selección de instancias.

Selección de Instancias

Grandes Bases de Datos. Estrategia de Estratificación.



Selección de Instancias

Selección de Instancias. Ejemplo – Kdd Cup'99

Nombre	Número de Instancias	Número de Atributos	Número de Clases
Kdd Cup'99	494022	41	23

Selección de Instancias

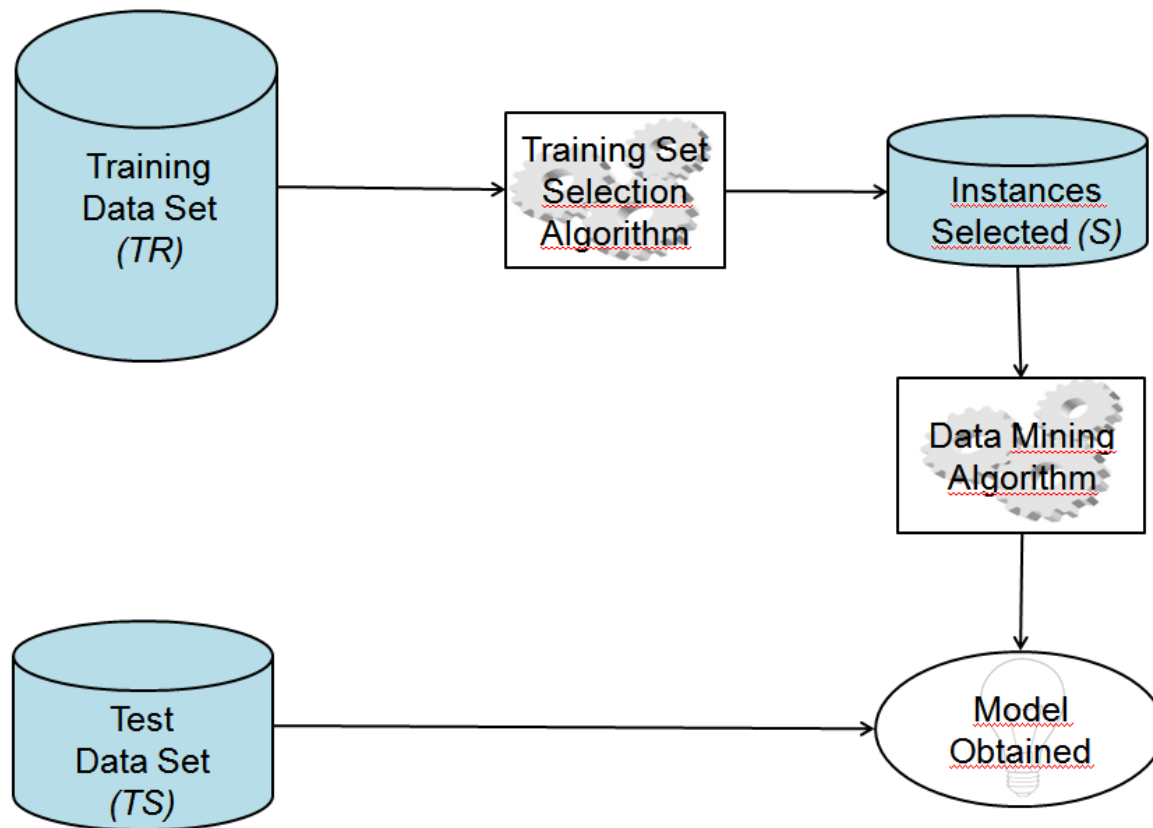
Selección de Instancias. Ejemplo – Kdd Cup'99

	Tiempo	% Red	% Ac. Trn	% Ac Test
1-NN cl	18568		99.91	99.91
Cnn st 100	8	81.61	99.30	99.27
Cnn st 200	3	65.57	99.90	99.15
Cnn st 300	1	63.38	99.89	98.23
Ib2 st 100	0	82.01	97.90	98.19
Ib2 st 200	3	65.66	99.93	98.71
Ib2 st 300	2	60.31	99.89	99.03
Ib3 st 100	2	78.82	93.83	98.82
Ib3 st 200	0	98.27	98.37	98.93
Ib3 st 300	0	97.97	97.92	99.27
CHC st 100	1960	99.68	99.21	99.43
CHC st 200	418	99.48	99.92	99.23
CHC st 300	208	99.28	99.93	99.19

J.R. Cano, F. Herrera, M. Lozano, **Stratification for Scaling Up Evolutionary Prototype Selection**. *Pattern Recognition Letters*, 26, (2005), 953-963.

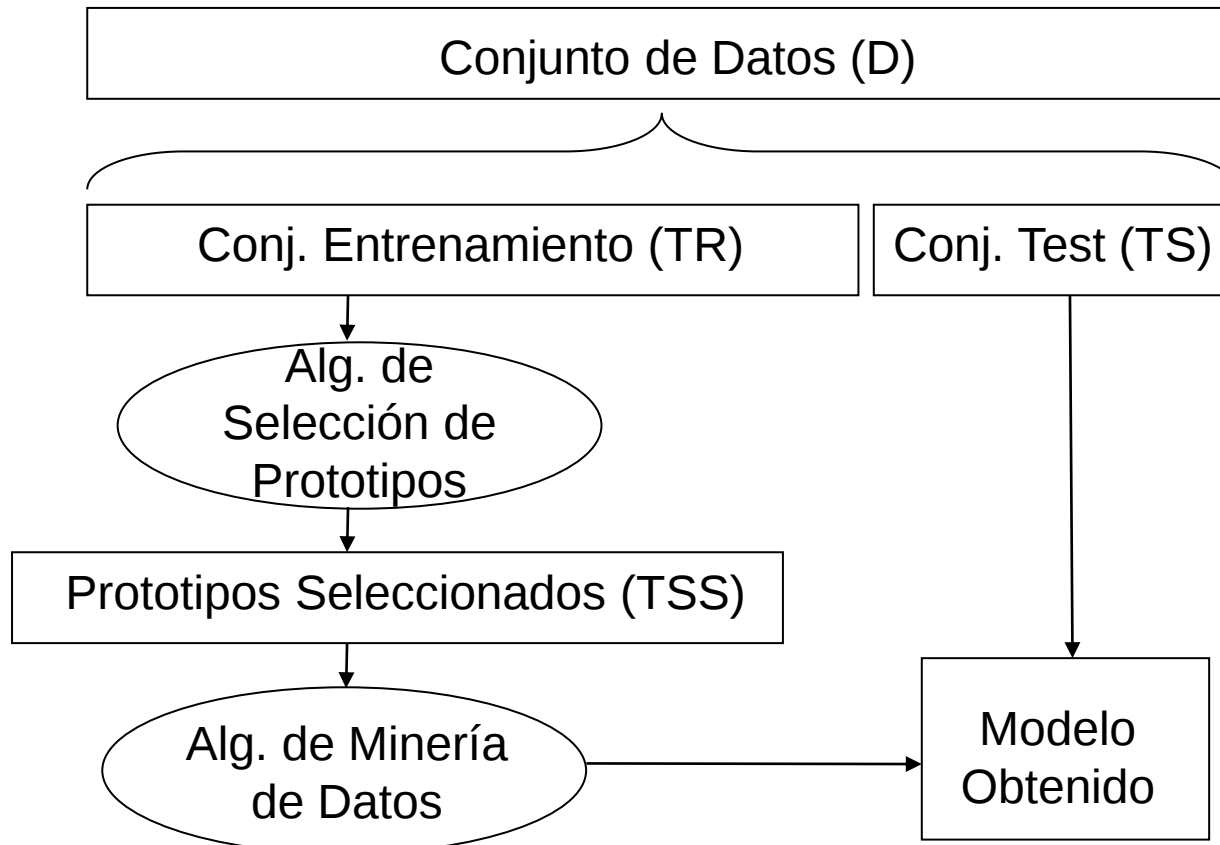
Selección de Instancias

Selección de Prototipos la Selección de Conjuntos de Entrenamiento



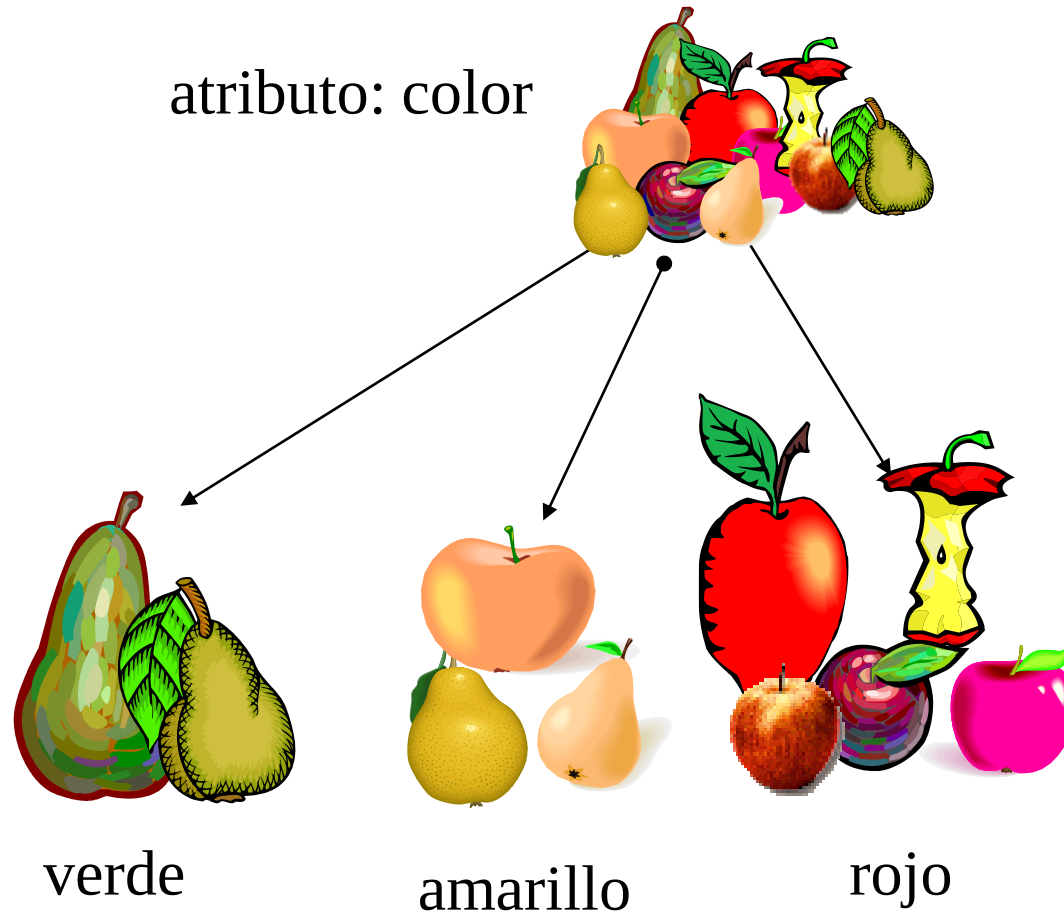
Selección de Instancias

Selección de Prototipos la **Selección de Conjuntos de Entrenamiento**



Ej. Selección de Instancias y Extracción de Árboles de Decisión

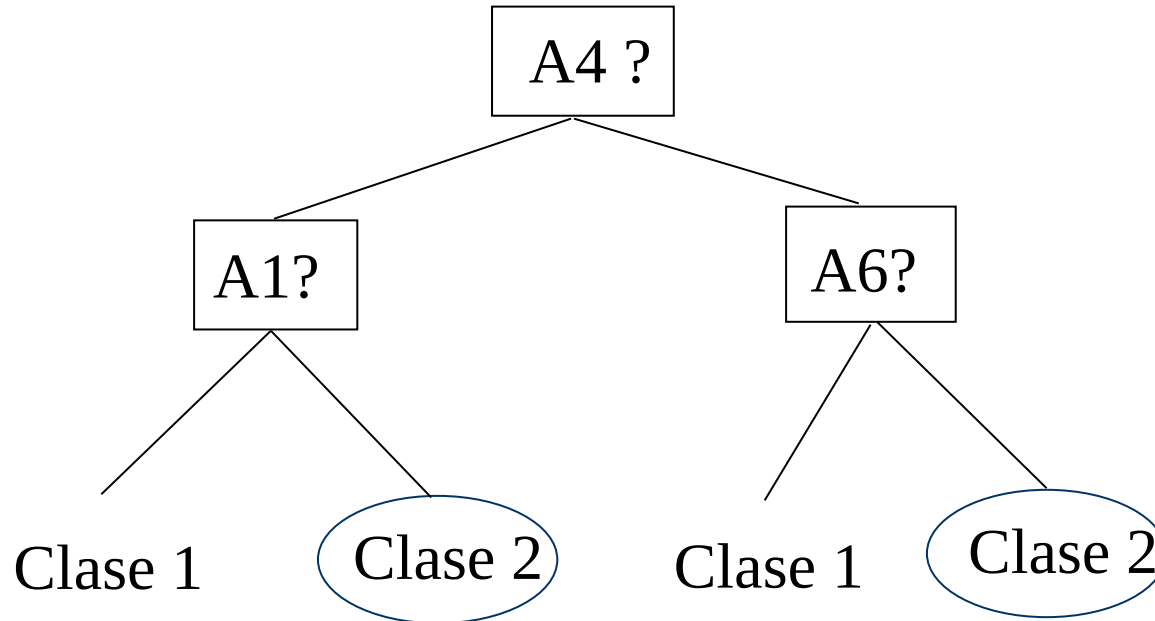
Selección de Instancias



Ej. Selección de Instancias y Extracción de Árboles de Decisión

Selección de Instancias

Conjunto inicial de atributos:
 $\{A1, A2, A3, A4, A5, A6\}$



-----> Conjunto reducido de atributos: $\{A1, A4, A6\}$

Los árboles de decisión seleccionan características

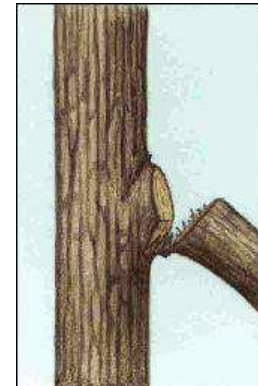
Ej. Selección de Instancias y Extracción de Árboles de Decisión

Selección de Instancias



**Comprehensibilidad:
Árboles de Tamaño reducido**

**Se utilizan técnicas de poda
eliminación de nodos**



Las estrategias de selección de instancias permiten construir árboles de decisión para grandes bases de datos reduciendo el tamaño de los árboles.

Aumentan su interpretabilidad.

Ej. Selección de Instancias y Extracción de Árboles de Decisión

Selección de Instancias

Kdd Cup'99. Número de estratos: 100

	No. Reglas	% Reducción	C4.5	
			%Ac Trn	%Ac Test
<i>C4.5</i>	252		99.97%	99.94%
<i>Cnn Strat</i>	83	81.61%	98.48%	96.43%
<i>Drop1 Strat</i>	3	99.97%	38.63%	34.97%
<i>Drop2 Strat</i>	82	76.66%	81.40%	76.58%
<i>Drop3 Strat</i>	49	56.74%	77.02%	75.38%
<i>Ib2 Strat</i>	48	82.01%	95.81%	95.05%
<i>Ib3 Strat</i>	74	78.92%	99.13%	96.77%
<i>Icf Strat</i>	68	23.62%	99.98%	99.53%
<i>CHC Strat</i>	9	99.68%	98.97%	97.53%

Ej. Selección de Instancias y Extracción de Árboles de Decisión

Selección de Instancias

La selección de instancias nos permite obtener conjuntos de reglas más interpretables y con aporte de mayor información.

	No. Instan- cias - N	No. Varia- bles	No. Reglas		No. Variables/ regla		Confidencia de las Reglas $N(\text{Cond}, \text{Clas})/N$	
Adult 2 clases	30132	14	C4.5	IS-CHC/ C4.5	C4.5	IS-CHC/ C4.5	C4.5	IS-CHC/ C4.5
			359	5	14	3	0.003	0.167

Bibliografía: J.R. Cano, [F. Herrera](#), [M. Lozano](#), **Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade-off Precision-Interpretability**. *Data and Knowledge Engineering* 60 (2007) 90-108, [doi:10.1016/j.datak.2006.01.008](https://doi.org/10.1016/j.datak.2006.01.008).

Selección de Instancias

Conjuntos de datos no balanceados

- Algunos problemas tienen una presencia de las clases desigual
 - Diagnóstico médico: 90% sin-enfermedad, 10% enfermedad
 - e-comercio: 99% no-compra, 1% compra
 - seguridad: >99.99% de conexiones no son ataques
- La situación es similar con múltiples clases
- La mayoría de los clasificadores obtienen un 90 o 99% de clasificación correcta en los casos anteriores, pero no son útiles

Selección de Instancias

Conjuntos de datos no balanceados

¿Cómo se procesan las clases no balanceadas?

Selección de Instancias

Conjuntos de datos no balanceados

¿Cómo se procesan las clases no balanceadas?

- a. Utilizar técnicas de reducción de datos para balancear las clases reduciendo las clases mayoritarias.
- b. Realizar sobremuestreo para balancear aumentar el tamaño de las clases minoritarias.

Selección de Instancias

Algunos otros aspectos a destacar

Generación de prototipos: Creación de prototipos artificiales para mejorar el comportamiento de los algoritmos.

Table 8.2 Some of the most important prototype generation methods

Complete name	Abbr. name	Reference
Prototype nearest neighbor	PNN	[27]
Generalized editing using nearest neighbor	GENN	[99]
Learning vector quantization	LVQ	[98]
Chen algorithm	Chen	[29]
Modified Chang's algorithm	MCA	[12]
Integrated concept prototype learner	ICPL	[102]
Depuration algorithm	Depur	[140]
Hybrid LVQ3 algorithm	HYB	[90]
Reduction by space partitioning	RSP	[141]
Evolutionary nearest prototype classifier	ENPC	[54]
Adaptive condensing algorithm based on mixtures of Gaussians	MixtGauss	[113]
Self-generating prototypes	SGP	[52]
Adaptive Michigan PSO	AMPSO	[25]
Iterative prototype adjustment by differential evolution	IPADE	[151]
Differential evolution	DE	[152]

Selección de Instancias

Algunos otros aspectos a destacar

Hibridación entre selección de instancias y características

Table 8.3 IS combined with FS and weighting

Description	Reference
Random mutation hill climbing for simultaneous instance and feature selection	[147]
Review of feature weighting methods for lazy learning algorithms	[163]
Distance functions for instance-based learning methods	[166]
Prototype reduction for sublinear space methods	[92]
Prototype reduction for sublinear space methods using ensembles	[93]
Learning feature weighting schemes for KNN	[129]
PS and feature weighting	[128]
PS for dissimilarity-based classifiers	[131]
Prototype reduction (selection and generation) for dissimilarity-based classifiers	[95]
Optimization of feature and instance selection with co-evolutionary algorithms	[59]
Prototype reduction and feature weighting	[55]
Instance and feature selection with cooperative co-evolutionary algorithms	[40]
Genetic algorithms for optimizing dissimilarity-based classifiers	[134]
Learning with weighted instances	[169]
Experimental review on prototype reduction for dissimilarity-based classifiers	[97]
Unification of feature and instance selection	[172]
Evolutionary IS with fuzzy rough FS	[43]
IS, instance weighting and feature weighting with co-evolutionary algorithms	[44]
Fuzzy rough IS for evolutionary FS	[45]
Feature and instance selection with genetic algorithms	[155]
Multi-objective genetic algorithm for optimizing instance weighting	[124]

Selección de Instancias

Algunos otros aspectos a destacar

Hibridación con técnicas de aprendizaje y multclasificadores

Table 8.4 Hybridizations with other learning approaches and ensembles

Description	Reference
First approach for nested generalized examples learning (hyperrectangle learning): EACH	[138]
Experimental review on nested generalized examples learning	[162]
Unification of rule induction with instance-based learning: RISE	[50]
Condensed nearest neighbour (CNN) ensembles	[5]
Inflating instances to obtain rules: INNER	[114]
Bagging for lazy learning	[174]
Evolutionary ensembles for classifiers selection	[142]
Ensembles for weighted IS	[71]
Bootstrapping for KNN	[148]
Evolutionary optimization in hyperrectangles learning	[67]
Evolutionary optimization in hyperrectangles learning for imbalanced problems	[69]
Review of ensembles for data preprocessing in imbalanced problems	[60]
Boosting by warping of the distance metric for KNN	[121]
Evolutionary undersampling based on ensembles for imbalanced problems	[61]

Selección de Instancias

Algunos otros aspectos a destacar **Estudios sobre escalabilidad**

Table 8.5 Scaling-up and distributed approaches

Description	Reference
Recursive subdivision of prototype reduction methods for tackling large data sets	[91]
Stratified division of training data sets to improve the scaling-up of PS methods	[20]
Usage of KD-trees for prototype reduction schemes	[120]
Distributed condensation for large data sets	[6]
Divide-and-conquer recursive division of training data for speed-up IS	[81]
Division of data based of ensembles with democratic voting for IS	[70]
Usage of stratification for scaling-up evolutionary algorithms for IS	[41]
Distributed implementation of the stratification process combined with k-means for IS	[33]
Scalable divide-and-conquer based on bookkeeping for instance and feature selection	[74]
Scaling-up IS based on the parallelization of small subsets of data	[82]

Selección de Instancias

WEBSITE:

<http://sci2s.ugr.es/pr/index.php>

Bibliografía:

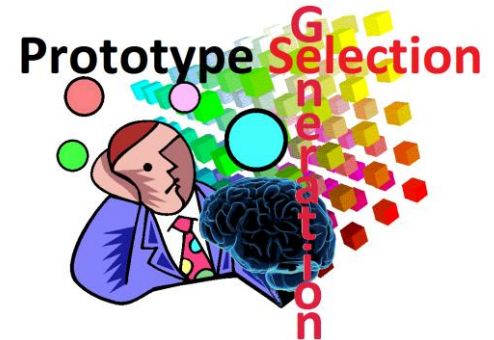
S. García, [J. Derrac](#), J.R. Cano and [F. Herrera](#),

Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study.

IEEE Transactions on Pattern Analysis and Machine Intelligence 34:3 (2012) 417-435 [doi:](#)

[10.1109/TPAMI.2011.142](https://doi.org/10.1109/TPAMI.2011.142)

S. García, J. Luengo, F. Herrera. **Data Preprocessing in Data Mining**, Springer, 2015



Códigos (Java):

KEEL

Selección de Instancias (website)

<http://sci2s.ugr.es/pr/>



Thematic Public Websites

Prototype Reduction in Nearest Neighbor Classification:
Prototype Selection and Prototype Generation



This **Website** contains SCI²S research material on Prototype Reduction in Nearest Neighbor Classification. This research is related to the following SCI²S surveys published recently:

- S. García, [J. Derrac](#), J.R. Cano and [F. Herrera](#), **Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study.**

IEEE Transactions on Pattern Analysis and Machine Intelligence 34:3 (2012) 417-435 [doi: 10.1109/TPAMI.2011.142](#)



COMPLEMENTARY MATERIAL to the paper [here](#): datasets, experimental results and source codes.

- [I. Triguero](#), [J. Derrac](#), S. García and [F. Herrera](#), **A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification** . *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews* 42:1 (2012) 86-100, [doi: 10.1109/TSMCC.2010.2103939](#)



COMPLEMENTARY MATERIAL to the paper [here](#): datasets, experimental results and source codes.

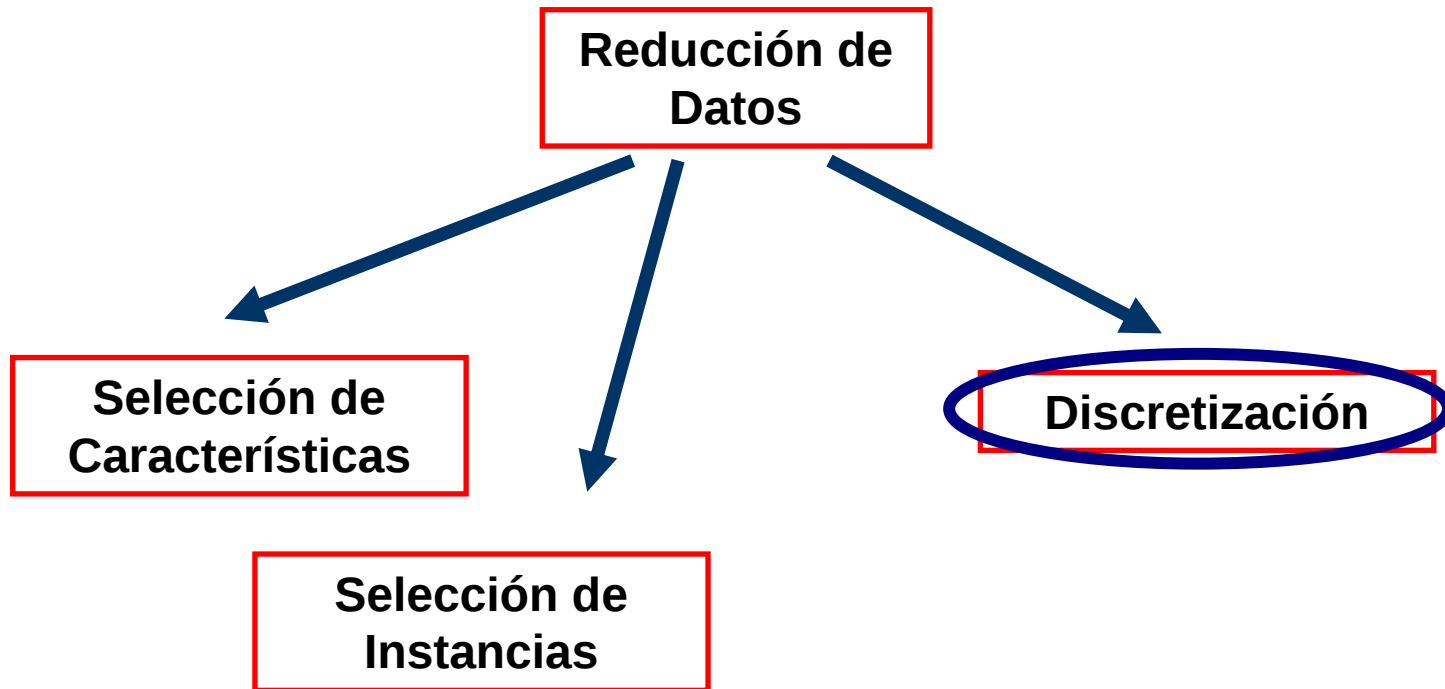
Selección de Instancias (website)

<http://sci2s.ugr.es/pr/>

The web is organized according to the following **Summary**:

- 1 [Introduction to Prototype Reduction](#)
- 2 [Prototype Selection](#)
 - 2.1 [Background](#)
 - 2.2 [Taxonomy](#)
 - 2.3 [Prototype Selection Methods](#)
 - 2.4 [Experimental Analyses](#)
 - 2.5 [SCI²S Approaches on Prototype Selection](#)
- 3 [Prototype Generation](#)
 - 3.1 [Background](#)
 - 3.2 [Taxonomy](#)
 - 3.3 [Prototype Generation Methods](#)
 - 3.4 [Experimental Analyses](#)
 - 3.5 [SCI²S Approaches on Prototype Generation](#)
- 4 [Prototype Reduction Outlook](#)
 - 4.1 [Key Milestones & Surveys](#)
 - 4.2 [Evolutionary Proposals](#)
 - 4.3 [SCI²S Related Approaches](#)
 - 4.4 [Prototype Reduction Visibility at the Web of Science](#)
 - 4.5 [Software and Algorithm Implementations](#)
 - 4.6 [Recent and forthcoming approaches](#)

Reducción de Datos



Bibliografía:

S. García, [J. Luengo](#), [José A. Sáez](#), [V. López](#), [F. Herrera](#), A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning.

IEEE Transactions on Knowledge and Data Engineering, [doi: 10.1109/TKDE.2012.35](#).

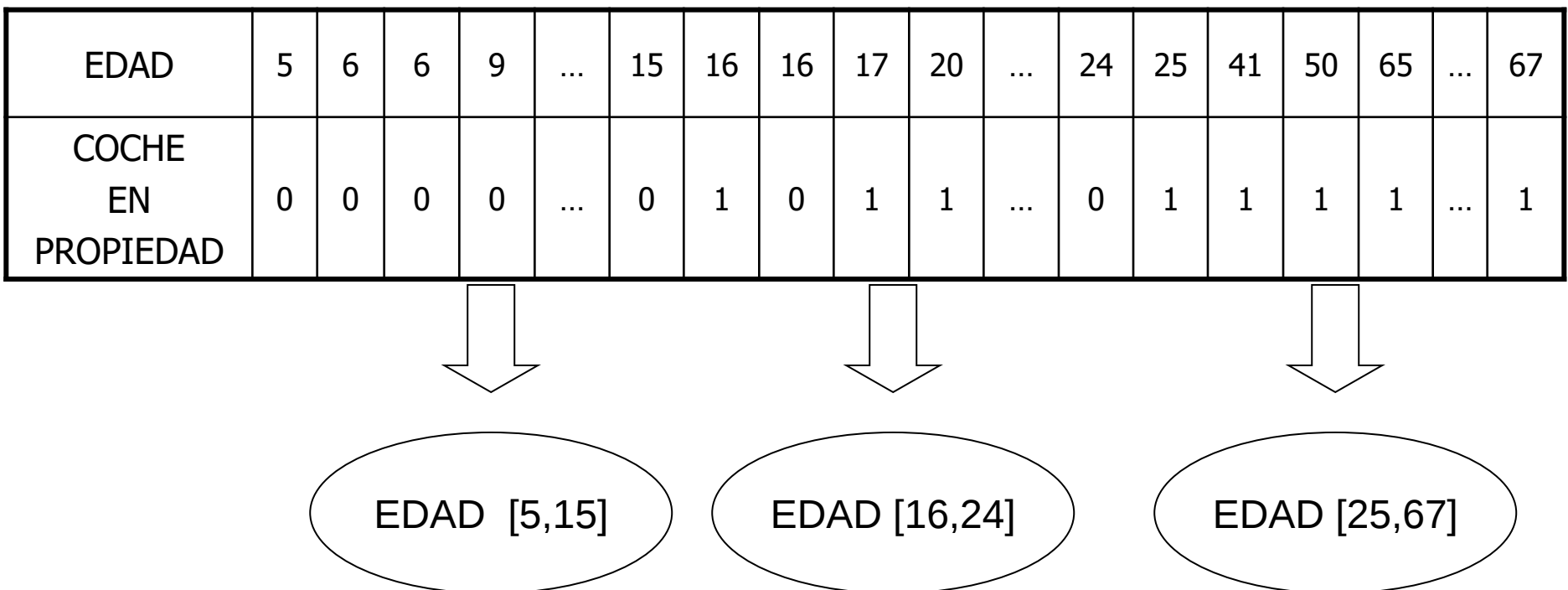
WEBSITE: <http://sci2s.ugr.es/discretization/>

Discretización

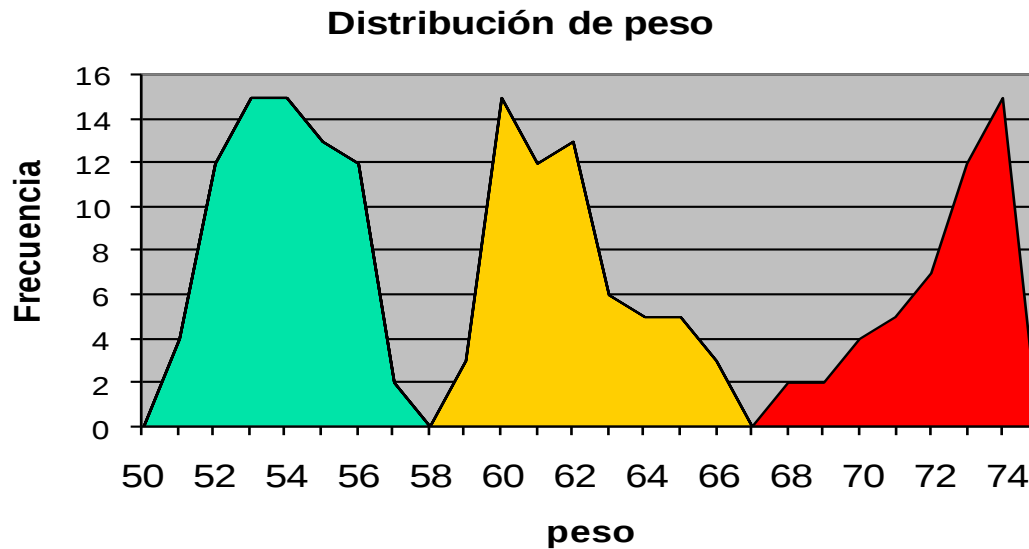
- Los valores discretos son muy útiles en Minería de Datos.
- Representan información más concisa, son más fáciles de entender más cercanos a la representación a nivel de conocimiento.
- La discretización busca transformar los valores continuos/discretos que se encuentran ordenados en valores nominales que no están ordenados. Proceso de cuantificación de atributos numéricos.
- Los valores nominales tienen un dominio finito, por lo que también se considera una técnica de reducción de datos.
- La discretización puede hacerse antes de la obtención de conocimiento o durante la etapa de obtención de conocimiento.

Discretización

- Divide el rango de atributos continuos (numéricos) en intervalos
- Almacena solo las etiquetas de los intervalos
- Importante para reglas de asociación y clasificación, algunos algoritmos solo aceptan datos discretos.



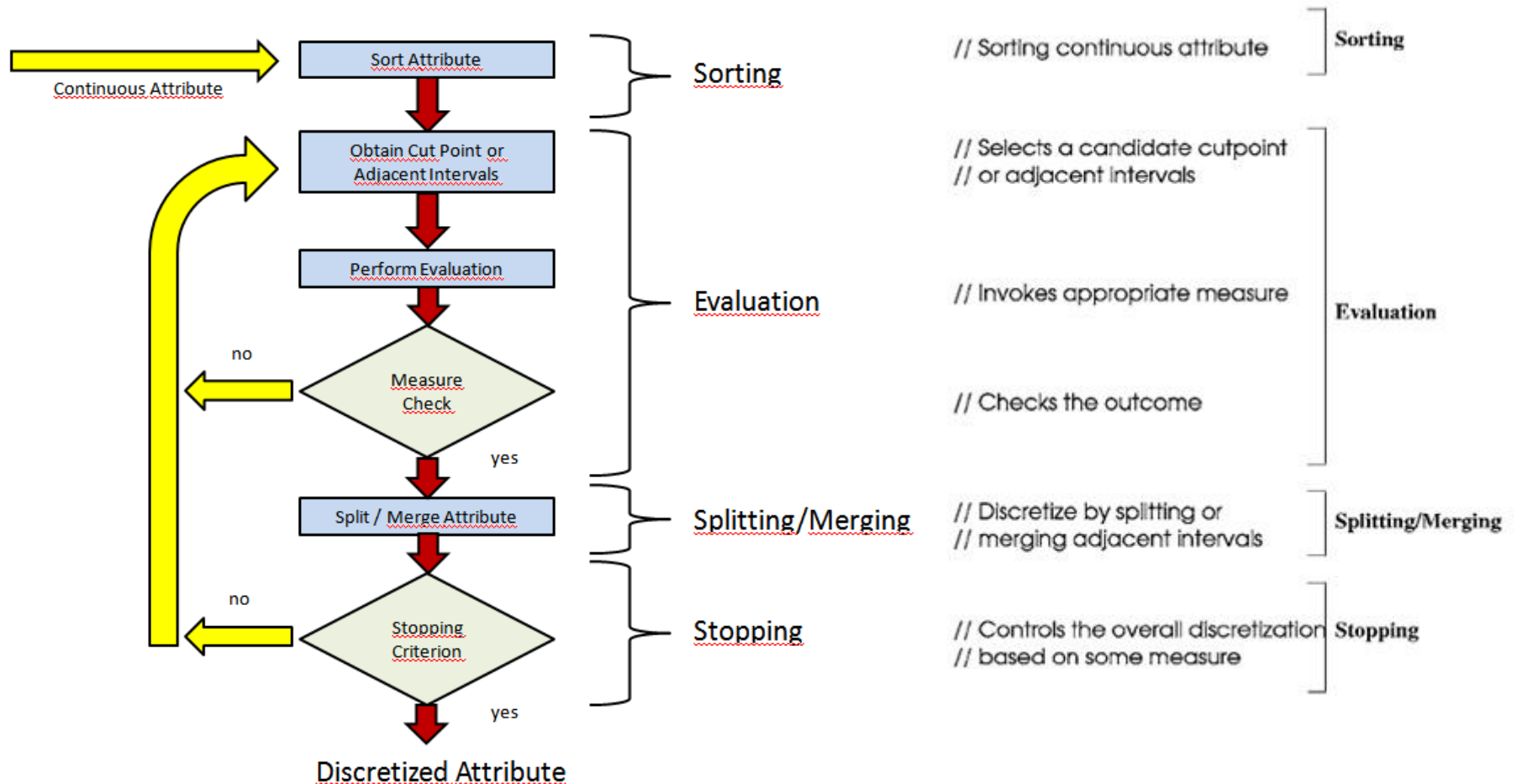
Discretización



50 - 58 kg
59-67 kg
> 68 kg

Discretización

Etapas en el proceso de discretización



Discretización

- **La discretización se ha desarrollado a lo largo de diferentes líneas según las necesidades:**
- **Supervisados vs. No supervisados:** Consideran o no el atributo objetivo.
- **Dinámicos vs. estáticos:** Mientras se construye o no el modelo.
- **Locales vs. Globales:** Centrados en una subregión del espacio de instancias o considerando todas ellas.
- **Top-down vs. Bottom-up:** Empiezan con una lista vacía o llena de puntos de corte.
- **Directos vs. Incrementales:** Usan o no un proceso de optimización posterior.

Discretización

■ Algoritmos no supervisados:

- Intervalos de igual amplitud
- Intervalos de igual frecuencia
- Clustering

■ Algoritmos supervisados:

- Basados en Entropía [Fayyad & Irani 93 and others]

[Fayyad & Irani 93] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. 13th Int. Joint Conf. AI (IJCAI-93)*, 1022-1027. Chamberry, France, Aug./Sep. 1993.

- Metodos Chi-square [Kerber 92]

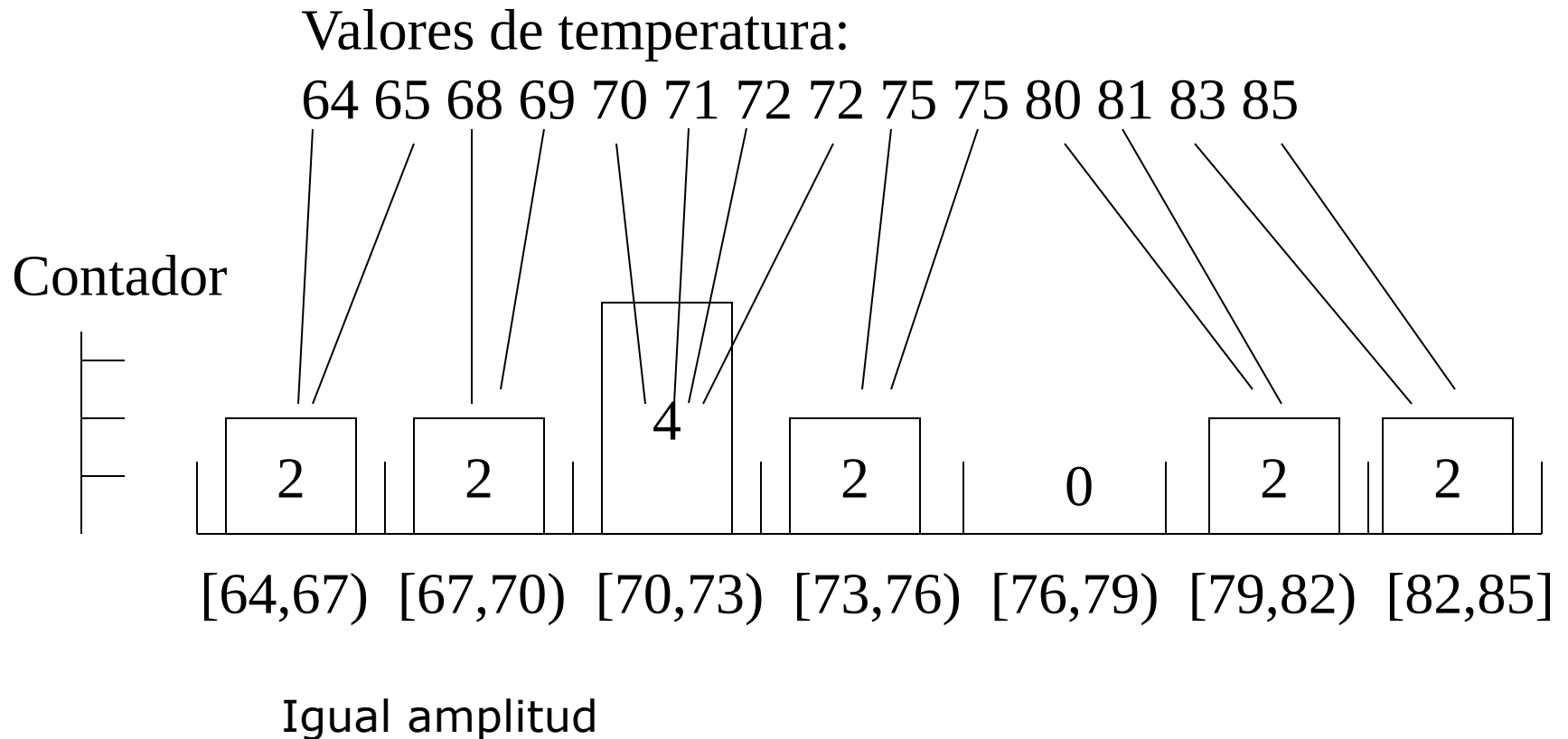
[Kerber 92] R. Kerber. ChiMerge: Discretization of numeric attributes. *Proc. 10th Nat. Conf. AAAI*, 123-128. 1992.

- ... (múltiples propuestas)

Bibliografía: S. García, J. Luengo, José A. Sáez, V. López, F. Herrera, A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. *IEEE Transactions on Knowledge and Data Engineering* 25:4 (2013) 734-750, [doi: 10.1109/TKDE.2012.35](https://doi.org/10.1109/TKDE.2012.35).

Discretización

Ejemplo Discretization: Igual amplitud



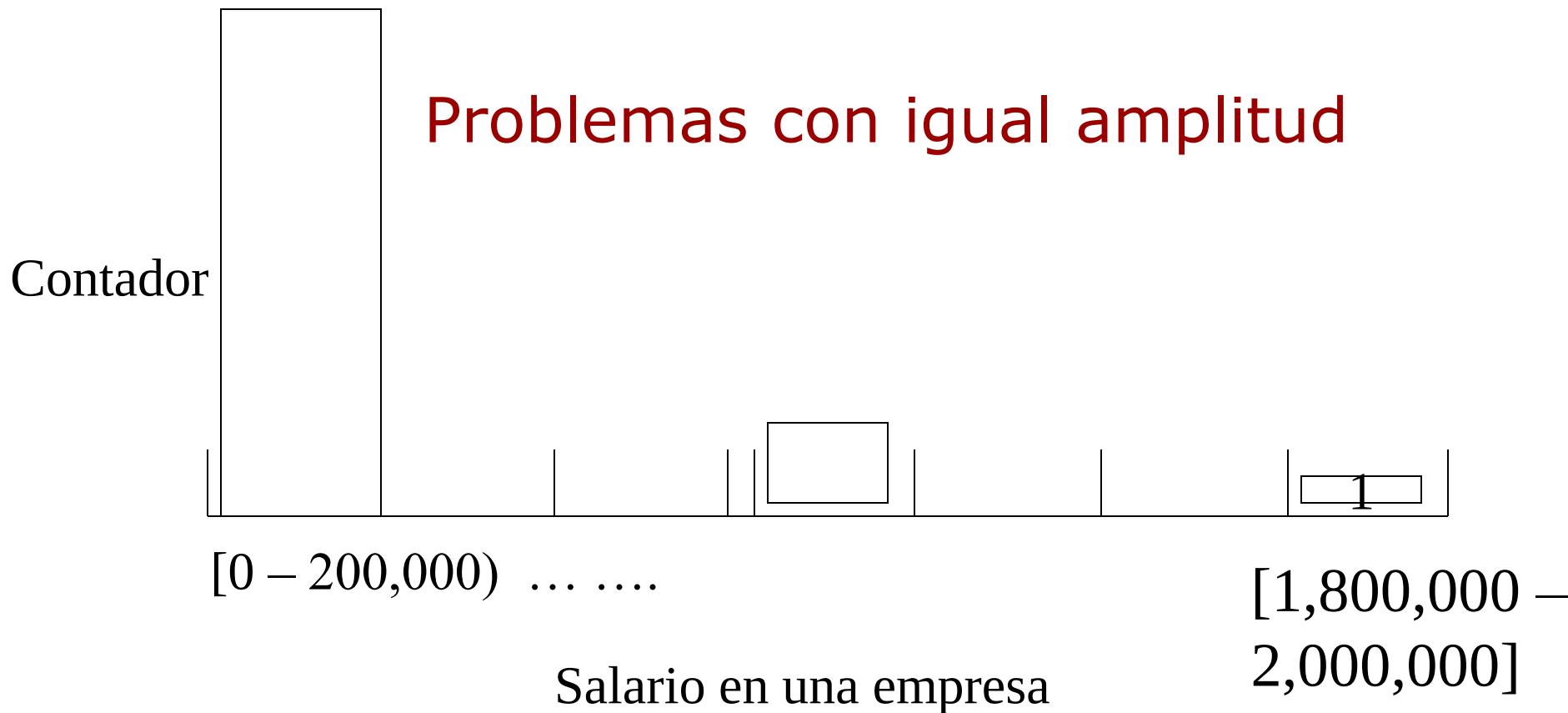
Discretización

Discretización:
igual amplitud puede producir
desequilibrios

Contador



Discretización



¿Qué podemos hacer para conseguir una mejor distribución?

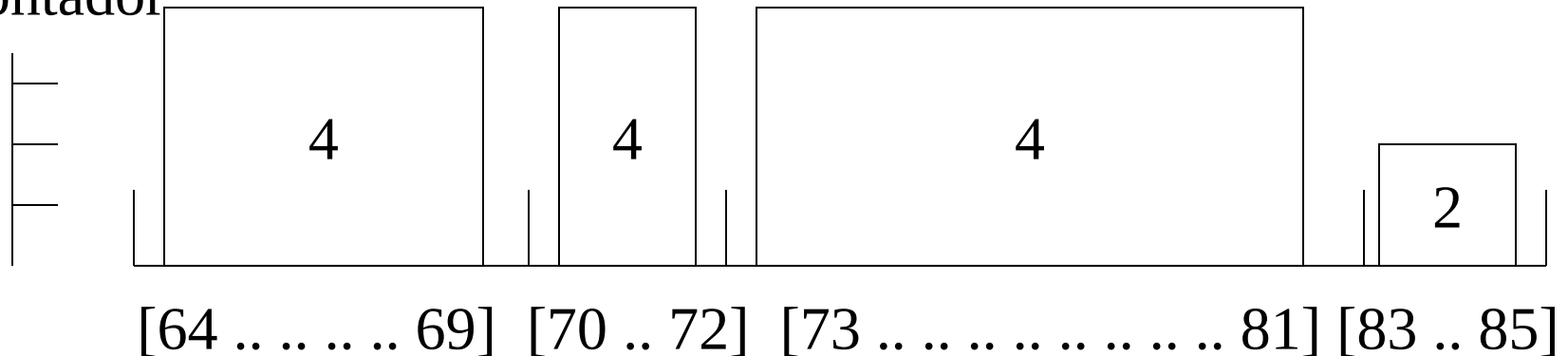
Discretización

Ejemplo Discretización: Igual frecuencia

Valores de la temperatura

64 65 68 69 70 71 72 72 75 75 80 81 83 85

Contador



Igual frecuencia (altura) = 4, excepto para la última caja

Discretización

Ejemplo: Discretización: Ventajas de la igualdad en frecuencia

- Generalmente es preferible porque evita desequilibrios en el balanceo entre valores
- En la práctica permite obtener puntos de corte mas intuitivos.
- Consideraciones adicionales:
 - Se deben crear cajas para valores especiales
 - Se deben tener puntos de corte interpretables

Discretización

■ Algoritmos no supervisados:

- Intervalos de igual amplitud
- Intervalos de igual frecuencia
- Clustering

■ Algoritmos supervisados:

- Basados en Entropía [Fayyad & Irani 93 and others]

[Fayyad & Irani 93] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. 13th Int. Joint Conf. AI (IJCAI-93)*, 1022-1027. Chamberry, France, Aug./Sep. 1993.

- Metodos Chi-square [Kerber 92]

[Kerber 92] R. Kerber. ChiMerge: Discretization of numeric attributes. *Proc. 10th Nat. Conf. AAAI*, 123-128. 1992.

- ... (múltiples propuestas)

Bibliografía: S. García, J. Luengo, José A. Sáez, V. López, F. Herrera, A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. *IEEE Transactions on Knowledge and Data Engineering* 25:4 (2013) 734-750, [doi: 10.1109/TKDE.2012.35](https://doi.org/10.1109/TKDE.2012.35).

Discretización

Discretizador Entropy MDLP (Fayyad)

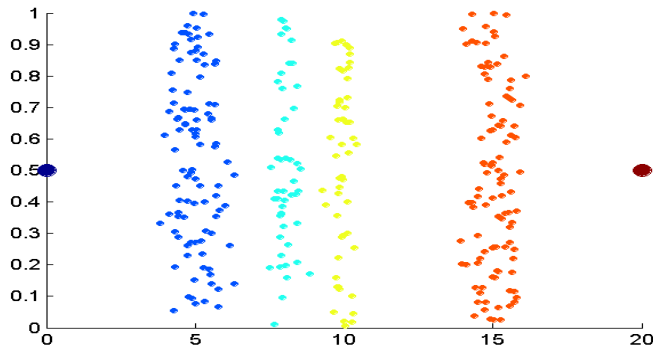
- Comienzan con los puntos de corte dados entre ejemplos de **diferentes** clases:

11	14	15	18	19	20	21	22	23	25	30	31	33	35	36
R	C	C	R	C	R	C	R	C	C	R	C	R	C	R

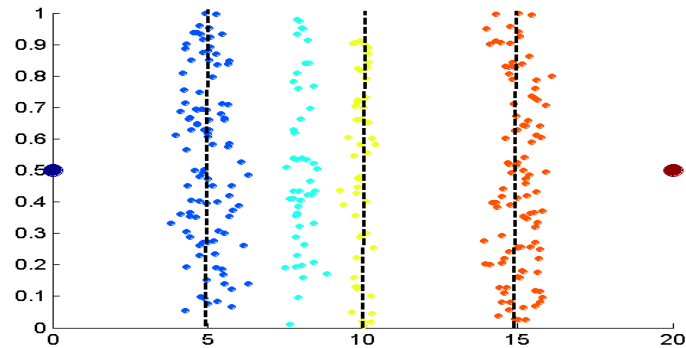
- Minimum Description Length Principle (MDLP), basado en entropía, se utiliza para escoger los puntos de corte útiles entre los anteriores.
- El criterio de parada se basa también en MDLP.
- MDLP se formula como el problema de encontrar el coste de comunicación entre un emisor y un receptor. Se asume que el emisor tiene el conjunto de instancias mientras que el receptor tiene las etiquetas de clase.
- Se dice que una partición inducida por un punto de corte es aceptada si y solo si el coste del mensaje requerido para enviar antes de particionar es mayor que el requerido después de particionar.

Discretización

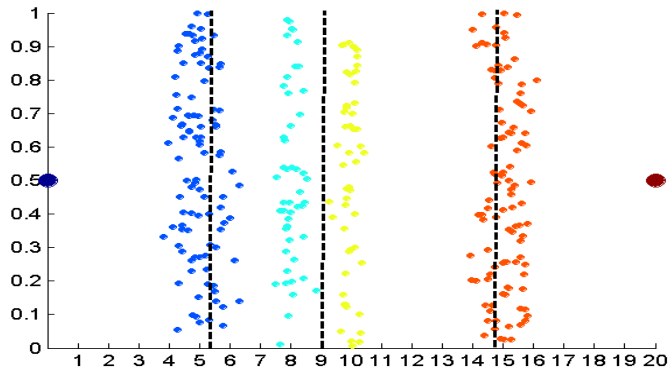
Discretización sin utilizar las clases



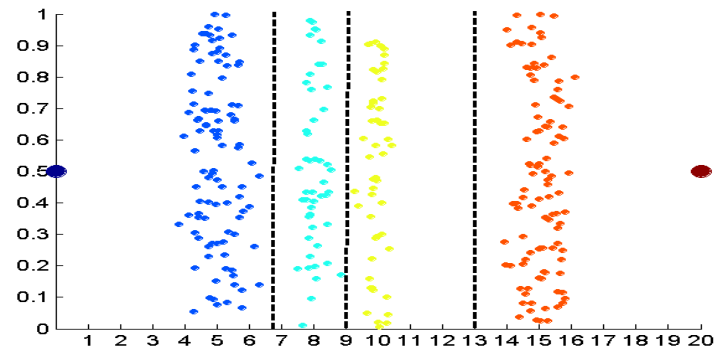
Datos



Igual anchura de intervalo



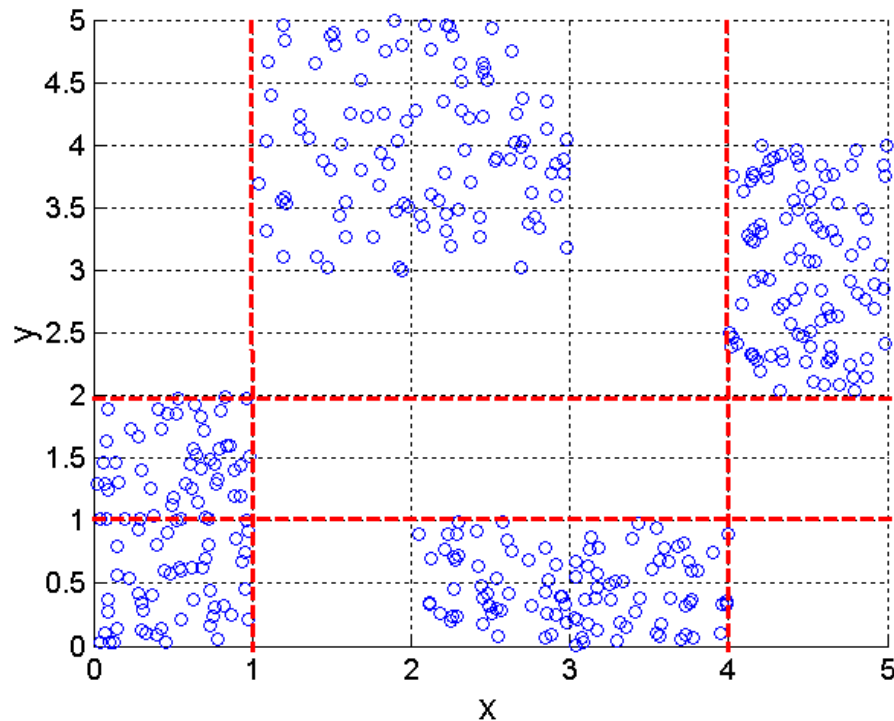
Igual frecuencia



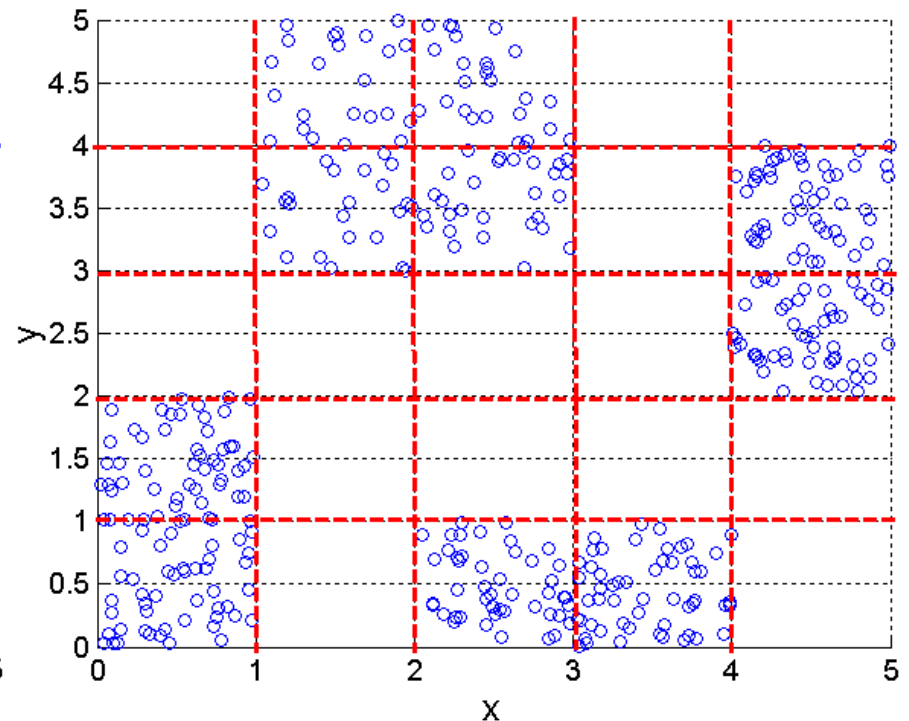
K-medias

Discretización

Discretización utilizando clases (basado en entropía)



3 categorías para ambas variables x e y



5 Categorías para ambas variables

Discretización

- *¿Qué discretizador será mejor?.*
- Como siempre, dependerá de la aplicación, necesidades del usuario, etc...
- Formas de evaluación:
 - Número total de intervalos
 - Número de inconsistencias causadas
 - Tasa de acierto predictivo

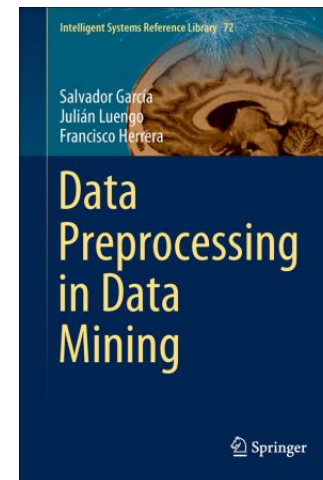
Inteligencia de Negocio

TEMA 4. Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

Bibliografía:

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



Comentarios Finales

El preprocesamiento de datos es una necesidad cuando se trabaja con una aplicación real, con datos obtenidos directamente del problema.

Comentarios Finales

Datos sin refinar

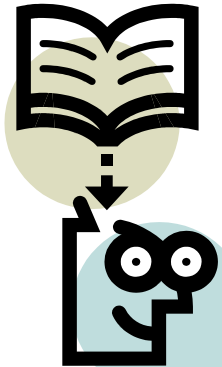


**Preprocesa-
miento
de
Datos**

**Obtención
de
Patrones**

**Interpretación
de
Resultados**

Conocimiento



- Preparación de Datos
- Reducción

- Reglas de asociación
- Clasificación / predicción
- Análisis de cluster

- Visualización
- Validación

Comentarios Finales

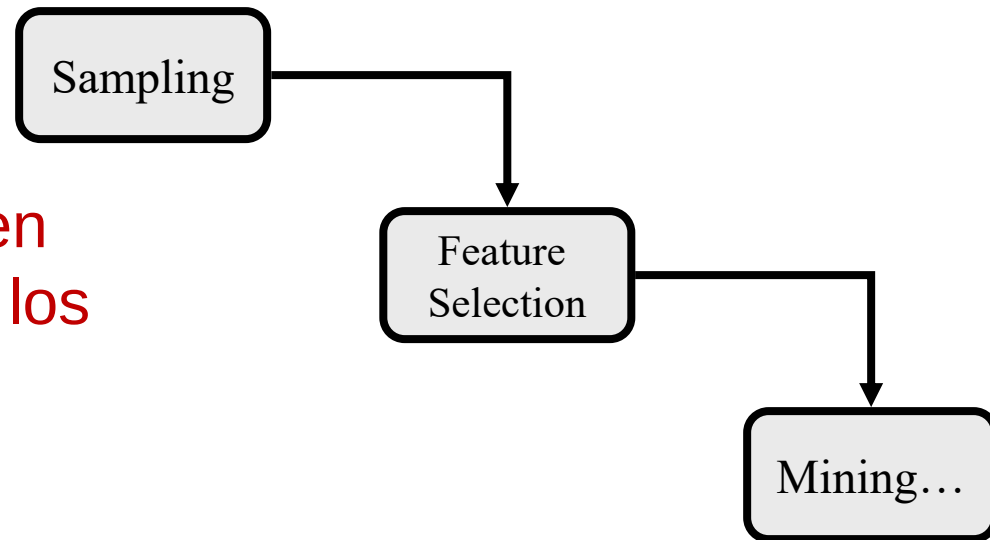
Una ventaja: El preprocesamiento de datos permite aplicar los modelos de Aprendizaje/Minería de Datos de forma más rápida y sencilla, obteniendo modelos/patrones de más calidad: precisión e/o interpretabilidad.

Un inconveniente: El preprocesamiento de datos no es un área totalmente estructurada con una metodología concreta de actuación para todos los problemas. Cada problema puede requerir una actuación diferente, utilizando diferentes herramientas de preprocesamiento.

Comentarios Finales

Un inconveniente: El preprocesamiento de datos no es un área totalmente estructurada con una metodología concreta de actuación para todos los problemas.

El diseño de procesos automáticos de uso de las diferentes etapas/técnicas en minería de datos es uno de los nuevos retos existentes.



Q. Yang, X. Wu

10 Challenging problems in data mining research.

International Journal of Information Technology & Decision Making 5:4 (2006) 597-604

Comentarios Finales

Las Técnicas de Reducción de Datos pueden permitir mejorar la precisión/interpretabilidad de los métodos de extracción de conocimiento, además de reducir el tamaño de la BD y el tiempo de los algoritmos de aprendizaje.

Para cada método de aprendizaje/problema puede ser necesario diseñar un mecanismo de reducción de datos que sea cooperativo con el propio método de aprendizaje.

“Good data preparation is key to producing valid and reliable models”

Snapshot on Data Preprocessing

Every problem can need a different preprocessing process, using different tools.



Knowledge-Based Systems

Volume 98, 15 April 2016, Pages 1–29



<http://sci2s.ugr.es/most-influential-preprocessing>

Tutorial on practical tips of the most influential data preprocessing algorithms in data mining

Salvador García^{a, b, *}, Julián Luengo^{a, *}, Francisco Herrera^{a, b, *}

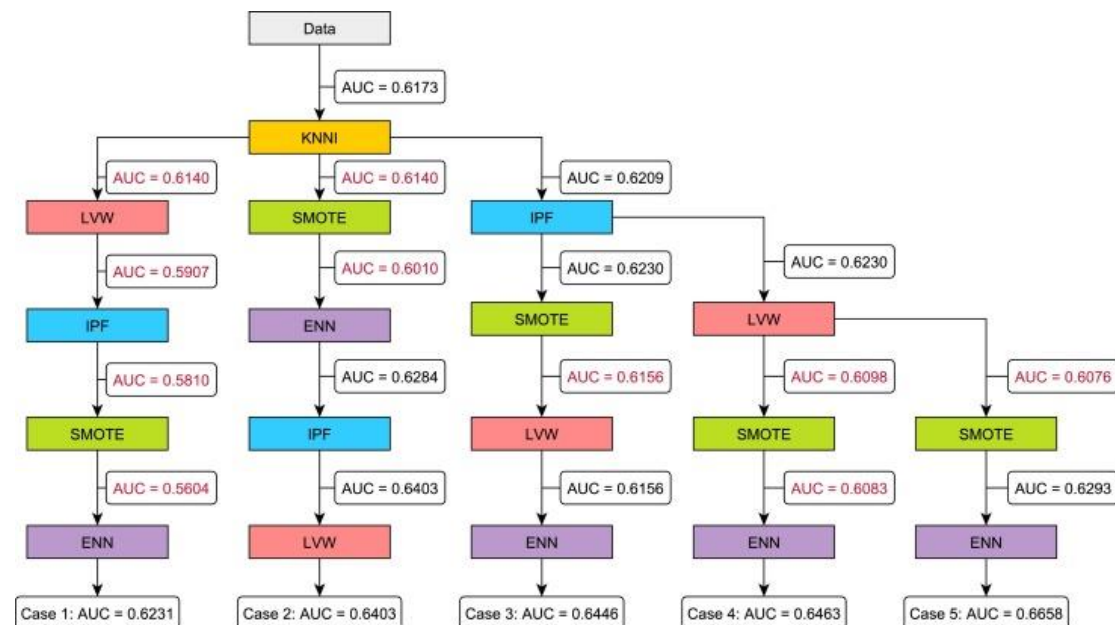
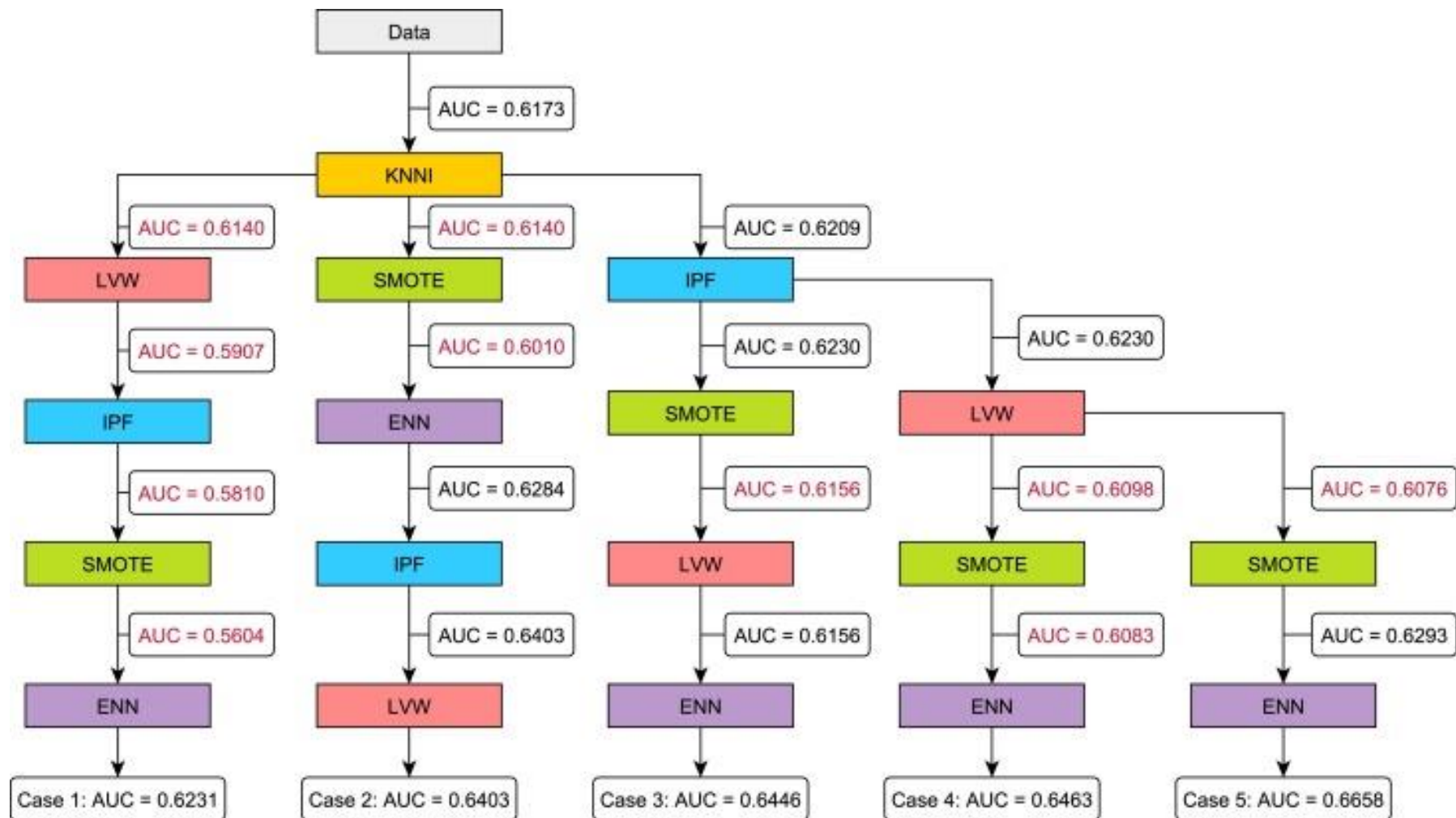


Fig. 29. Depiction of the 5 Cases, hierarchically distributed by the preprocessing application order, showing the AUC values obtained in each step

Snapshot on Data Preprocessing



Comentarios Finales

El software de minería de datos KEEL (knowledge extraction based on evolutionary learning) incluye un módulo de preparación de datos de datos (selección de características, imputación de valores perdidos, selección de instancias, discretización, ...)



KEEL

<http://www.keel.es/>

Comentarios Finales

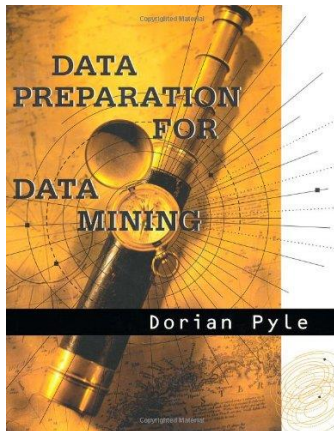
El software de minería de datos KEEL (knowledge extraction based on evolutionary learning) incluye un módulo de preparación de datos de datos (selección de características, imputación de valores perdidos, selección de instancias, discretización, ...

Algorithms included in KEEL (484)		
Family	Subfamily	
Data Preprocessing (98)	Discretization (30)	
	Feature Selection (25)	Feature Selection (22)
		Evolutionary Feature Selection (3)
	Training Set Selection (16)	Training Set Selection (12)
		Evolutionary Training Set Selection (4)
	Missing Values (15)	
	Transformation (4)	
	Data Complexity (1)	
	Noisy Data Filtering (7)	

Bibliografía

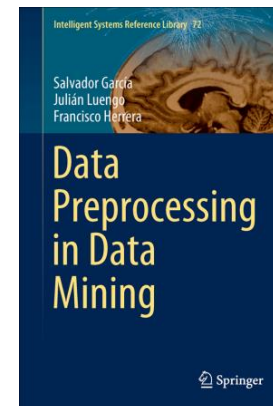


Bibliografía – Minería de Datos y Preprocesamiento



Dorian Pyle
Morgan Kaufmann, 1999

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015



Competición

[Sign up](#)[Login](#)

Academic Machine Learning Competitions

Theory, meet practice.

Kaggle hosts free projects for hundreds of universities around the globe. Engage students with an opportunity to apply machine learning to real problems.

[Learn about hosting](#)

Berkeley
UNIVERSITY OF CALIFORNIA



Cornell

ERASMUS
UNIVERSITY



THE UNIVERSITY OF
MELBOURNE

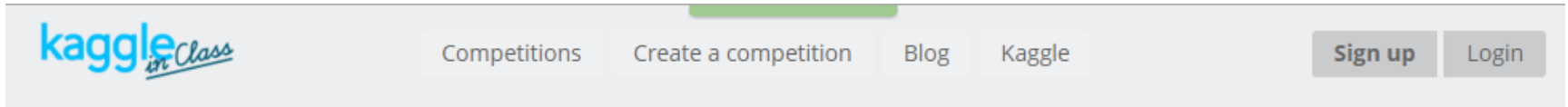
MICHIGAN

UNIVERSITY OF
OXFORD

Stanford
University

UNIVERSITY
OF TORONTO

Ej. Competición 2015

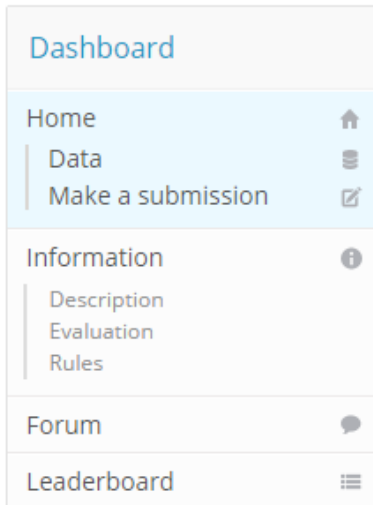


Knowledge • 16 teams

UGR15: Preprocessing for Protein Structure Prediction

Thu 12 Nov 2015

Wed 6 Jan 2016 (50 days to go)



Competition Details » [Get the Data](#) » [Make a submission](#)



This competition is private-entry. You can view but not participate.

Deal with a dataset on Protein Structure Prediction by preprocessing methods (feature/instance selection, noise filters, missing values)

INTELIGENCIA DE NEGOCIO

2020-2021



- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- Tema 7. Modelos Avanzados de Minería de Datos.
- Tema 8. Big Data