

PREPROCESAMIENTO DE DATOS

Minería de Datos: Preprocesamiento y clasificación

Julián Luengo

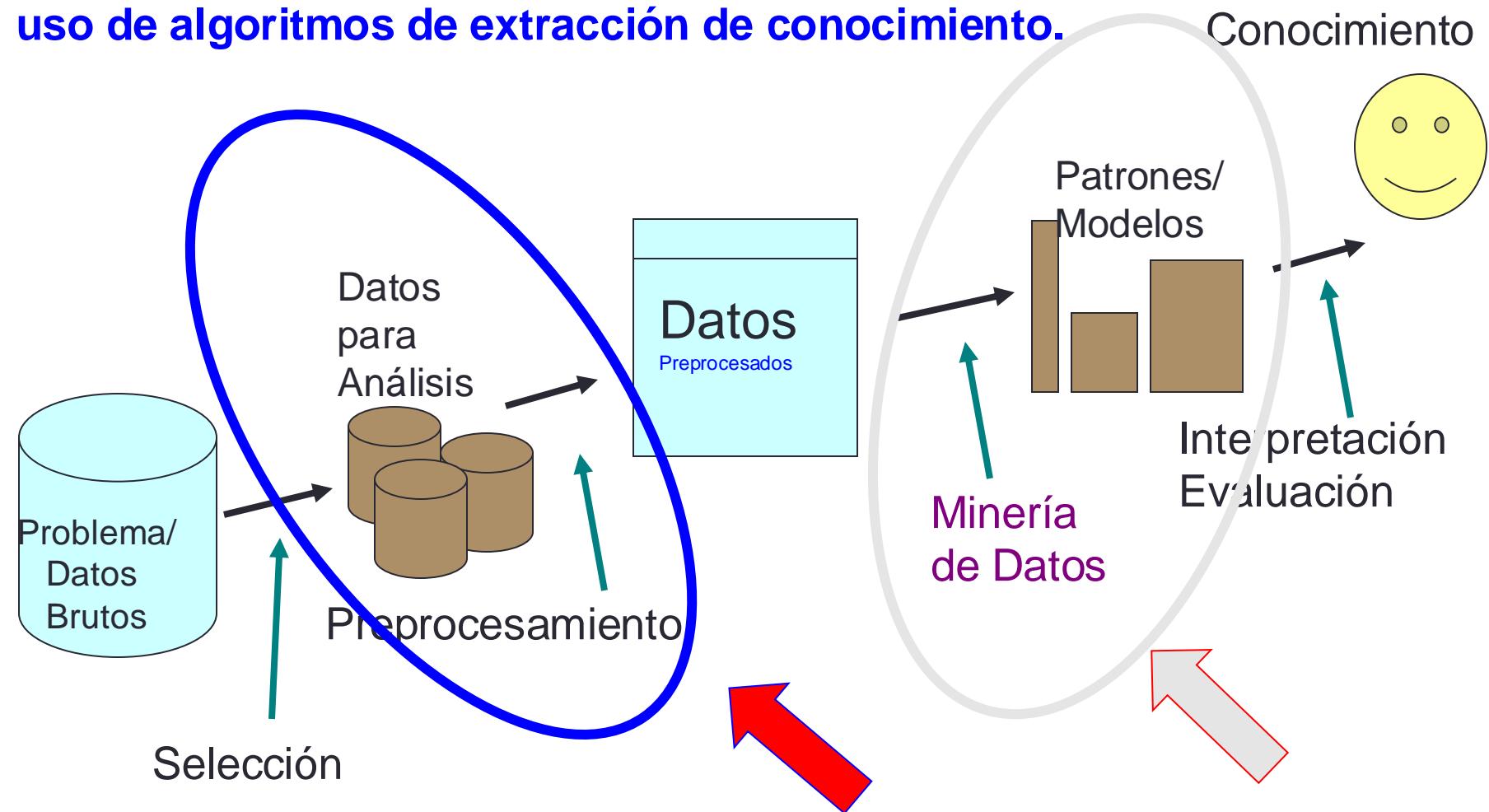
julianlm@ugr.es

Bibliografía.

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

Preprocesamiento de Datos

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



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.

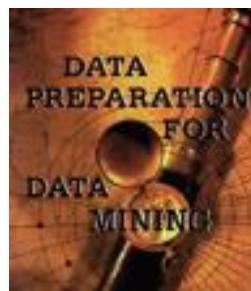
Preprocesamiento de Datos

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

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

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, ...

Reducción 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

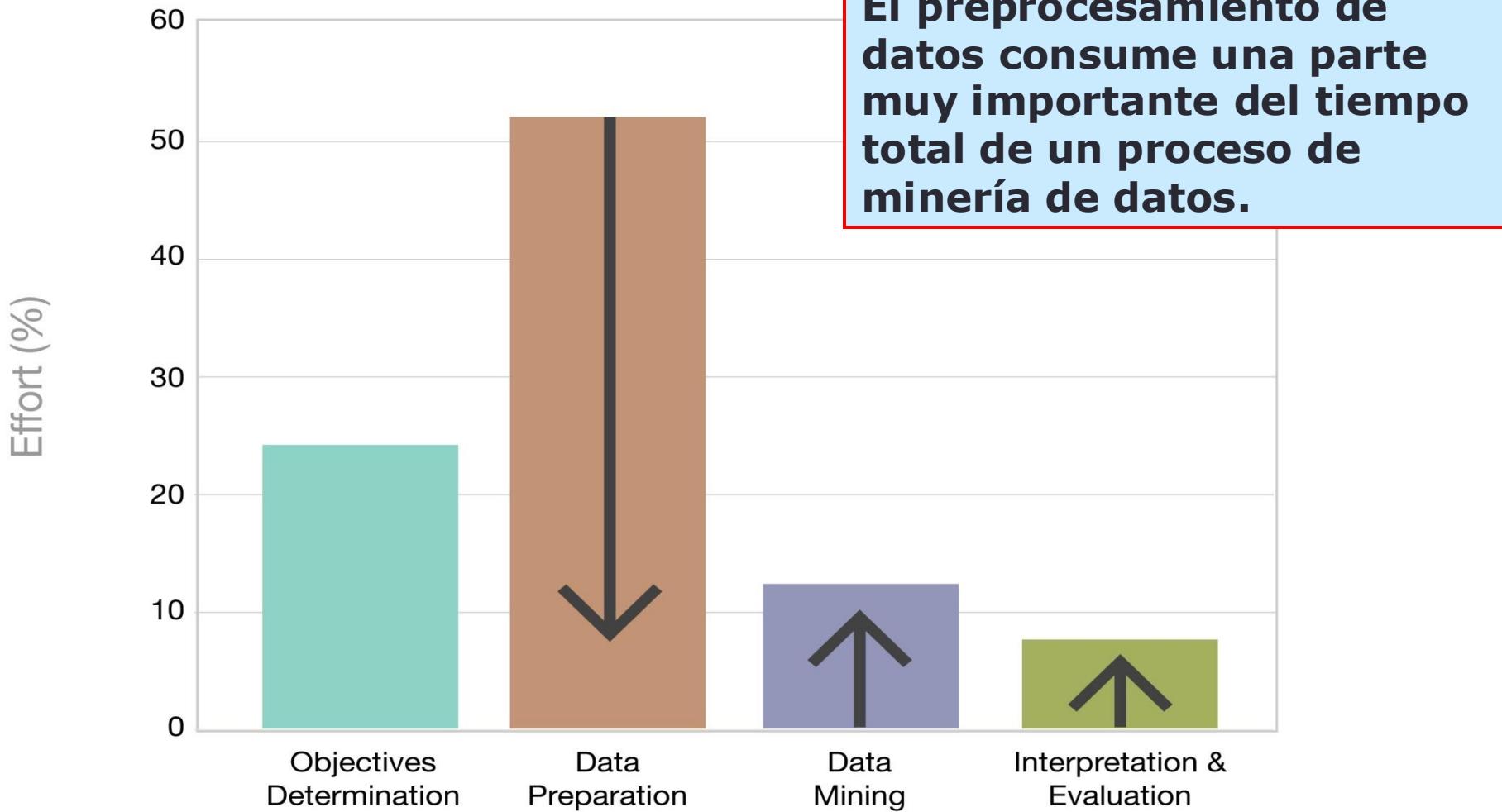
Importancia del Preprocesamiento de Datos

- 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 (90%).

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?

Las BBDD reales en la actualidad suelen contener datos ruidosos, perdidos y/o inconsistentes, fundamentalmente por su gran tamaño.

¿Cómo se pueden procesar los datos para mejorar la eficiencia y la facilidad de aplicación del proceso de minería de datos?

1. Integración de datos. Fusión de múltiples fuentes en un DW.
2. Limpieza de datos. Eliminación de ruido e inconsistencias.
3. Transformación de datos. Normalización, Discretización, ...
4. Reducción de la dimensionalidad. Disminuir el tamaño de las BBDD mediante agregación y/o eliminación de variables redundantes, ...

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

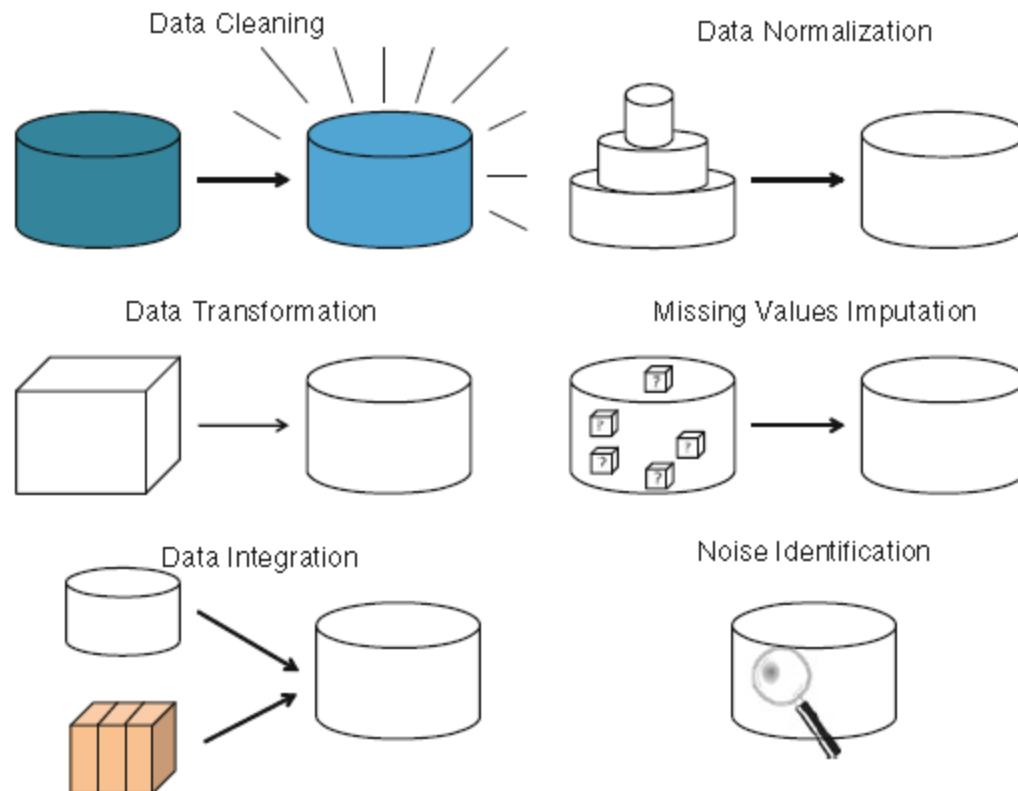


Fig. 1.3 Forms of data preparation

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

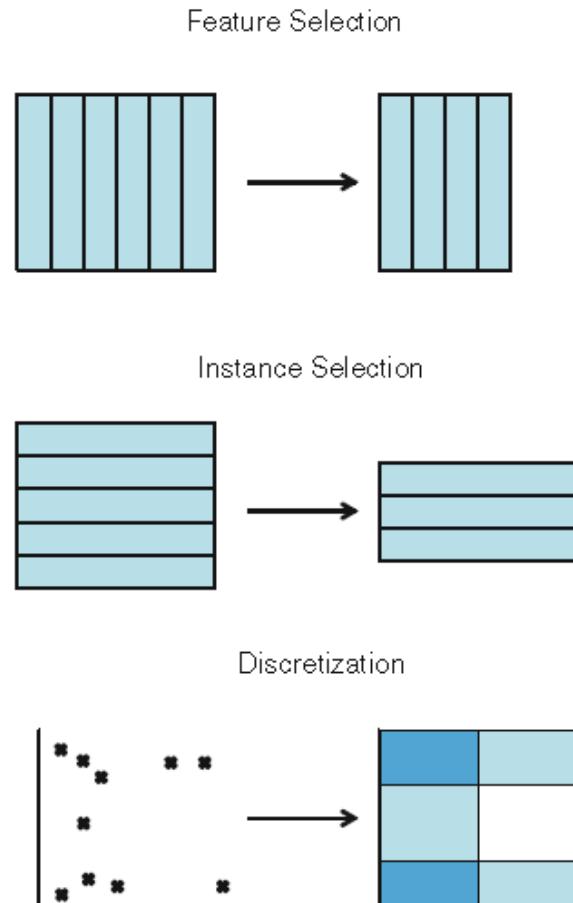


Fig. 1.4 Forms of data reduction

Preprocesamiento de Datos

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

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



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

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 $1/x$ ($x=1, \dots, 5$)

	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.

Limpieza de datos

Limpieza de Datos: Ejemplo de Inconsistencia

Presencia de discrepancias en datos

Edad=“42”

Fecha de Nacimiento=“03/07/1997”

Normalización (MinMaxScaling)

- **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.

$$\begin{aligned} & [\min_A, \max_A] \rightarrow [nuevo_{\min_A}, nuevo_{\max_A}] \\ & v' = \frac{v - \min_A}{\max_A - \min_A} (nuevo_{\max_A} - nuevo_{\min_A}) + nuevo_{\min_A} \end{aligned}$$

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.

Datos Categóricos y Codificación

- **Objetivo:** Utilizar algoritmos de aprendizaje que trabajan bien con datos números en datos categóricos.

- **Medidas de distancia para datos categóricos**

Cosine distance (Angular)

Jaccard distance (Magnitud)

Hamming distance (one-hot encoding)

Value Difference Metric (VDM)

- **Técnicas de Codificación**

Label Encoding

One Hot Encoding / Dummy Encoding

Hash Encoding

Binary Encoding

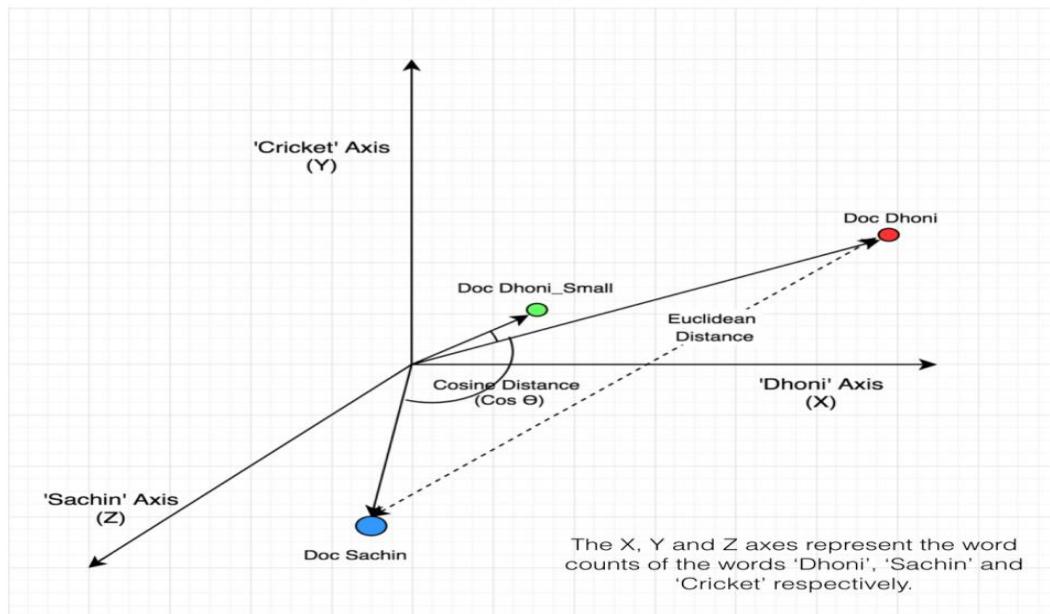
BaseN Encoding

Target Encoding

Datos Categóricos y Codificación

- **Cosine distance:** mide el coseno del ángulo entre dos vectores proyectados en un espacio multidimensional. En este contexto, los dos vectores son matrices que contienen el número de palabras de dos documentos.

Projection of Documents in 3D Space



$$\text{Cos}\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ is the dot product of the two vectors.

Datos Categóricos y Codificación

- **Jaccard distance:** mide las similitudes entre conjuntos. Se define como el tamaño de la intersección dividido por el tamaño de la unión de dos conjuntos. Esta noción se ha generalizado para los conjuntos múltiples, donde los elementos duplicados se cuentan como pesos. Muy usado en entornos multi-atributo.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- **Hamming distance (One-Hot encoding):** Para cada una de las variables nominales, podría crear una representación binaria 1-de-K (donde K es el número de valores que puede tomar la variable nominal). A continuación, se puede calcular la distancia Hamming entre estas representaciones binarias (tenga en cuenta que para los datos binarios la distancia Hamming y la L1 serán equivalentes).

Datos Categóricos y Codificación

- **Value difference Metric:** Una versión simplificada del VDM (sin los esquemas de ponderación) define la distancia entre dos valores x e y de un atributo a como

$$vdm_a(x, y) = \sum_{c=1}^C \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q = \sum_{c=1}^C |P_{a,x,c} - P_{a,y,c}|^q$$

- $N_{a,x}$ is the number of instances in the training set T that have value x for attribute a ;
- $N_{a,x,c}$ is the number of instances in T that have value x for attribute a and output class c ;
- C is the number of output classes in the problem domain;
- q is a constant, usually 1 or 2; and
- $P_{a,x,c}$ is the conditional probability that the output class is c given that attribute a has the value x , i.e., $P(c | x_a)$. As can be seen from (8), $P_{a,x,c}$ is defined as:

$$P_{a,x,c} = \frac{N_{a,x,c}}{N_{a,x}}$$

where $N_{a,x}$ is the sum of $N_{a,x,c}$ over all classes, i.e.,

$$N_{a,x} = \sum_{c=1}^C N_{a,x,c}$$

Datos Categóricos y Codificación

One Hot Encoding /
Dummy Encoding

Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Label Encoding

Degree	
0	1
1	4
2	2
3	3
4	3
5	4
6	5
7	1
8	1

Column	Code
A	100
B	010
C	001

One- Hot Coding

Column	Code
A	10
B	01
C	00

Dummy Code

Datos Categóricos y Codificación

Hash Encoding

Al igual que la codificación one-hot, el codificador Hash representa las características categóricas utilizando las nuevas dimensiones. Aquí, el usuario puede fijar el número de dimensiones después de la transformación utilizando el argumento `n_componente`. Esto es lo que quiero decir - Una característica con 5 categorías puede ser representada usando N nuevas características, de manera similar, una característica con 100 categorías también puede ser transformada usando N nuevas características.

Por defecto, el codificador Hashing utiliza el algoritmo hashing md5 pero el usuario puede pasar cualquier algoritmo de su elección.

	Month	col_0	col_1	col_2	col_3	col_4	col_5
0	January	0	0	0	0	1	0
1	April	1	0	0	0	1	0
2	March	2	0	0	0	0	1
3	April	3	0	0	0	1	0
4	Februay	4	0	0	0	1	0
5	June	5	0	1	0	0	0
6	July	6	1	0	0	0	0
7	June	7	0	1	0	0	0
8	September	8	0	0	0	0	1

Datos Categóricos y Codificación

Binary Encoding

La codificación binaria es una combinación de la codificación Hash y la codificación ordinal. En este esquema de codificación, la característica categórica se convierte primero en numérica utilizando un codificador ordinal. A continuación, los números se transforman en números binarios. A continuación, el valor binario se divide en diferentes columnas. La codificación binaria funciona muy bien cuando hay un gran número de categorías. Por ejemplo, las ciudades de un país donde una empresa suministra sus productos.

BaseN Encoding

Para la codificación binaria, la base es 2, lo que significa que convierte los valores numéricos de una categoría en su respectiva forma binaria. Si desea cambiar la base del esquema de codificación, puede utilizar el codificador Base N. En el caso de que las categorías sean más y la codificación binaria no sea capaz de manejar la dimensionalidad, entonces podemos utilizar una base mayor, como 4 u 8.

	city_0	city_1	city_2
0	0	0	1
1	0	0	2
2	0	0	3
3	0	0	4
4	0	1	0
5	0	0	1
6	0	0	3
7	0	0	2
8	0	1	1

Datos Categóricos y Codificación

Target Encoding

Es una técnica de codificación bayesiana. Los codificadores bayesianos utilizan la información de las variables dependientes/objetivo para codificar los datos categóricos. En la target encoding, calculamos la media de la variable objetivo para cada categoría y sustituimos la variable de categoría por el valor medio. En el caso de las variables objetivo categóricas, la probabilidad posterior del objetivo sustituye a cada categoría.

Más Recursos:

<https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/>

<https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02>

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.3340&rep=rep1&type=pdf>

<https://ieeexplore.ieee.org/document/6889941>

[https://github.com/AjayMukundS/Categorical similarity measures](https://github.com/AjayMukundS/Categorical_similarity_measures)

Preprocesamiento de Datos

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

Datos Imperfectos

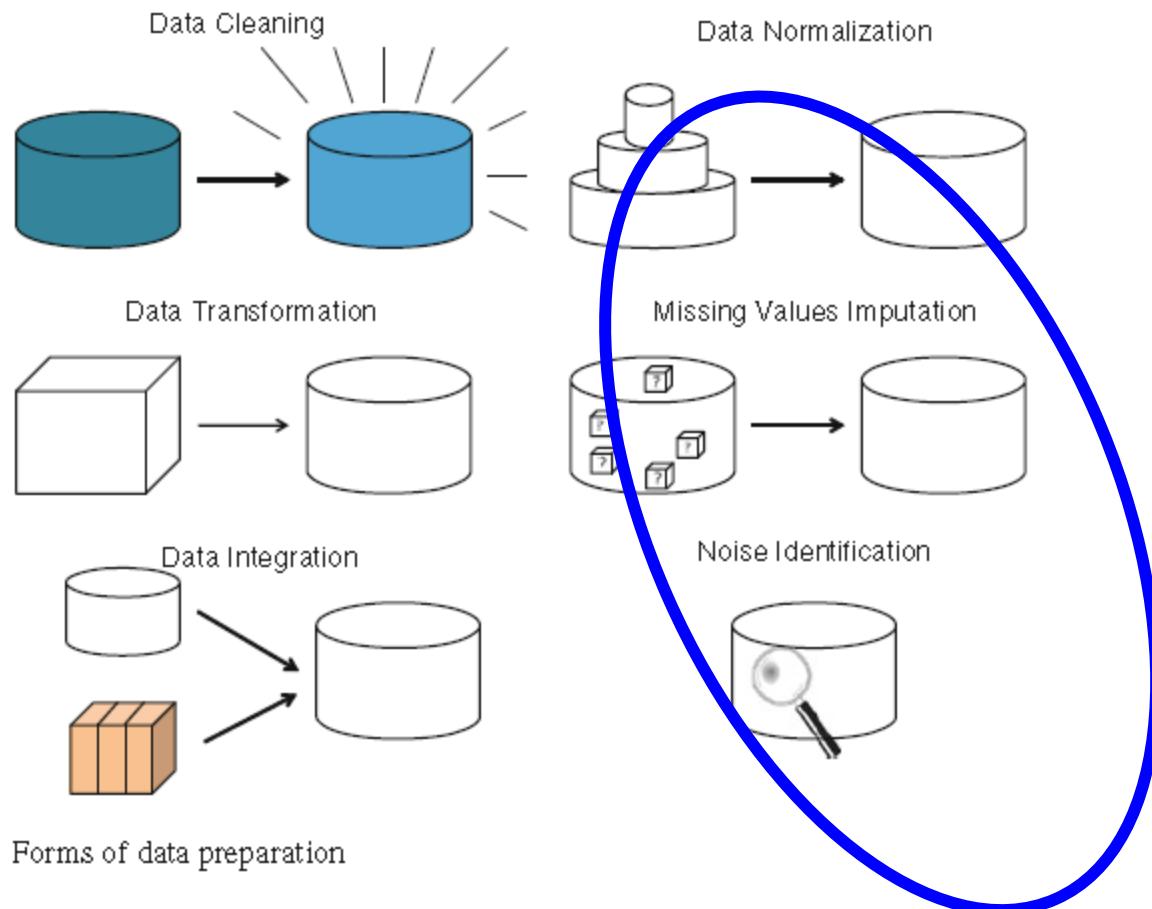


Fig. 1.3 Forms of data preparation

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

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

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

Valores perdidos (ejemplo)

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)
→ No resuelve nada

Suposiciones y mecanismos de Valores Perdidos en los datos

- $X = (X_{\text{obs}}, X_{\text{mis}})$,
 - Llamamos X_{obs} a la parte observada de X
 - Llamamos X_{mis} a la parte de valores de perdidos.
- Supongamos que disponemos de una matriz B del mismo tamaño que X
- Los valores de B son 0 ó 1 cuando los elementos de X son no perdidos o perdidos respectivamente.

Suposiciones y mecanismos de Valores Perdidos en los datos

- La distribución de B debería estar relacionada con X y con algunos parámetros desconocidos ζ , así que disponemos de un modelo de probabilidad para B descrito por $P(B|X, \zeta)$.
- Teniendo suposiciones *missing at random* (**MAR**), significa que esta distribución no depende de X_{mis} :

$$P(B|X_{obs}, X_{mis}, \zeta) = P(B|X_{obs}, \zeta).$$

Suposiciones y mecanismos de Valores Perdidos en los datos

- MAR nos sugiere que los valores no perdidos constituyan otra posible muestra desde la distribución de probabilidad.
 - Esta condición se conoce como *missing completely at random (MCAR)*.
- MCAR es un caso especial de MAR en la que la distribución de un ejemplo que tiene un valor perdido para un atributo no depende de los datos, sean o no datos observados

$$P(B|X_{obs}, X_{mis}, \zeta) = P(B|\zeta)$$

Suposiciones y mecanismos de Valores Perdidos en los datos

- Bajo MCAR, el análisis de sólo estas unidades con los datos completos nos dará inferencias válidas
 - Aunque habrá generalmente algo de pérdida de información en los datos
- MCAR es más restrictivo que MAR
 - MAR requiere solo que los valores perdidos se comporten como una muestra aleatoria de todos los valores de un subconjunto particular de clases definidos por los datos observados.

Suposiciones y mecanismos de Valores Perdidos en los datos

- Un tercer caso ocurre cuando MAR no es aplicable porque los valores perdidos dependen de ambas cosas: el resto de valores observados y el valor propio.

$$P(B|X_{obs}, X_{mis}, \zeta)$$

- Este modelo se llama not missing at random (**NMAR**) o missing not at random (**MNAR**) en la literatura.
- La única forma de obtener un estimador fiable es modelar la propia manera de perder datos.
 - Esta es una tarea muy compleja en la que se debe crear un modelo a cuenta de los valores perdidos que debería ser incorporado después a un modelo aún más complejo que se usará para estimar los valores perdidos.

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 descomposition (SVDI), local least squares imputation (LLSI)

Valores perdidos

MissForest:

- Imputa los valores perdidos usando Bosques aleatorios de manera iterativa.
- Por defecto, el imputador comienza a imputar los valores faltantes de la columna con el menor número de ellos: columna candidata.
- El primer paso consiste en llenar cualquier valor que falte de las columnas restantes, no candidatas, con una estimación inicial, que es la media de la columna para las columnas que representan variables numéricas y la moda para las columnas que representan variables categóricas.
- Despues, el imputador ajusta un modelo de bosque aleatorio con la columna candidata como variable de resultado y las columnas restantes como predictores en todas las filas en las que no faltan los valores de la columna candidata.

Valores perdidos

MissForest:

- Después del ajuste, se imputan las filas faltantes de la columna candidata utilizando la predicción del Bosque Aleatorio ajustado.
- Las filas de las columnas no candidatas actúan como datos de entrada para el modelo ajustado.
- A continuación, se pasa a la siguiente columna de candidata con el segundo menor número de valores faltantes de entre las columnas no candidatas en la primera ronda.
- El proceso se repite para cada columna con un valor que falta, posiblemente en múltiples iteraciones o épocas para cada columna, hasta que se cumpla el criterio de parada.
- El criterio de parada se rige por la "diferencia" entre las matrices imputadas en iteraciones sucesivas

Valores perdidos en clasificación. Ejemplo

Table 4.4 Average ranks for the Rule Induction Learning methods

	C45	Ripper	PART	Slipper	AQ	CN2	SRI	Rito	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. Algoritmos

MISSING VALUES		
Full Name	Short Name	Reference
Delete Instances with Missing Values	Ignore-MV	P.A. Gouraud, 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 en clasificación. Algoritmos

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.

J. Luengo, 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)

Software en R:



robCompositions

missingpy 0.2.0

Limpieza de datos con ruido

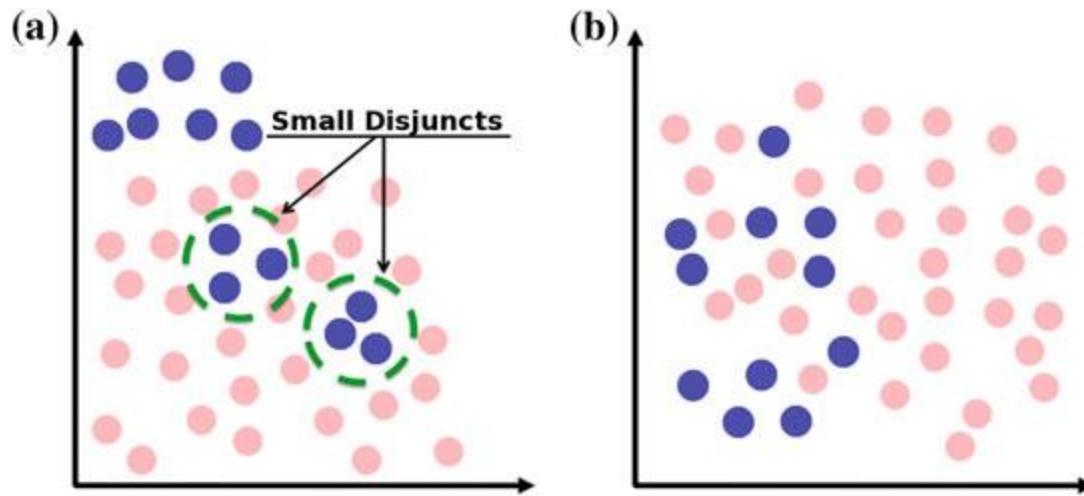


Fig. 5.1 Examples of the interaction between classes: a) small disjuncts and b) overlapping between classes

Limpieza de datos con ruido

Tipos de ejemplos

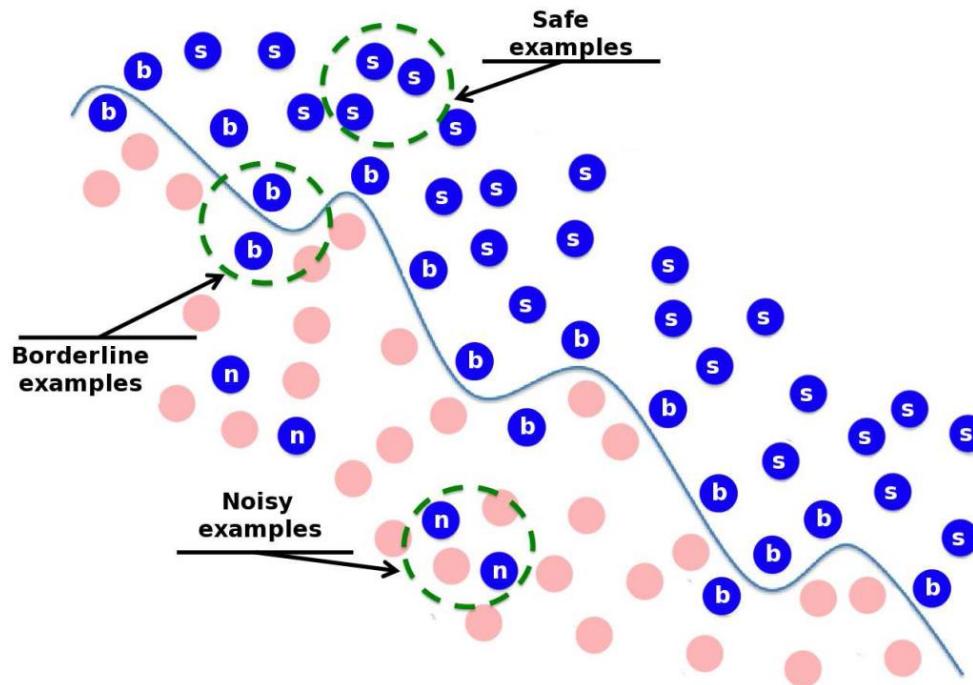


Fig. 5.2 The three types of examples considered in this book: safe examples (labeled as *s*), borderline examples (labeled as *b*) and noisy examples (labeled as *n*). The continuous line shows the decision boundary between the two classes

Tipos de Ruido

- Un gran número de componentes determinan la calidad de un conjunto de datos.
- Entre ellos, las etiquetas de clase y los valores de atributos influencian directamente a dicha calidad
 - La calidad de las etiquetas de clase se refiere a si la clase de cada ejemplo está correctamente asignada.
 - La calidad de los atributos se refiere a su capacidad de caracterizar apropiadamente a los ejemplos con el propósito de clasificarlos.

Tipos de Ruido

- Basándonos en estas dos fuentes de información, se pueden distinguir dos tipos de ruido:
 - **Ruido de clase** (también conocido como *label noise*) ocurre cuando un ejemplo está etiquetado de forma incorrecta. Se pueden distinguir dos tipos de ruido de clase:
 - *Ejemplos contradictorios*
 - *Clasificaciones mal hechas*
 - **Ruido de atributos** se refiere a las corrupciones en los valores de uno o más atributos → valores de atributos erróneos, valores desconocidos o perdidos en atributos, y atributos incompletos o valores “do not care”.

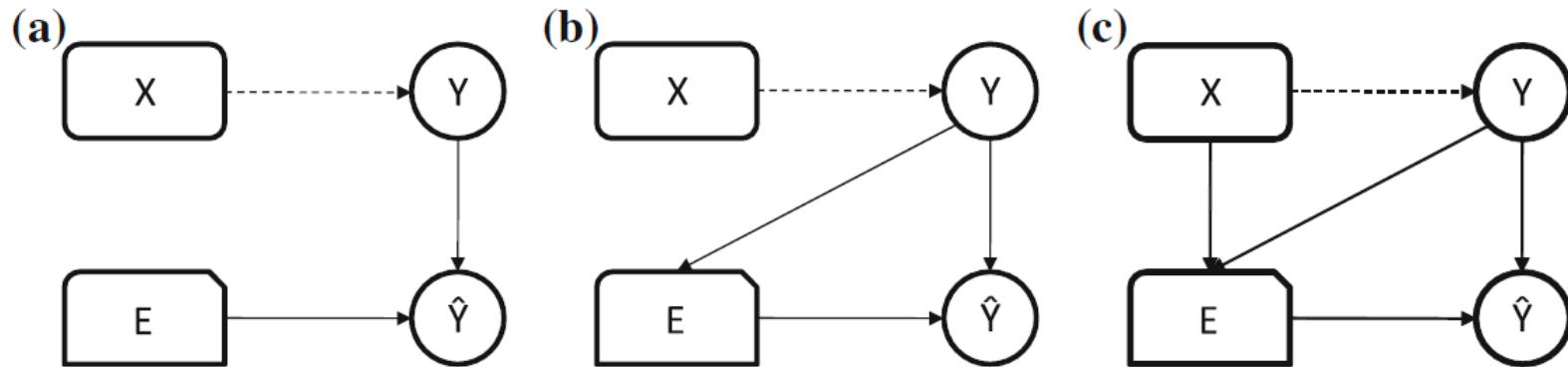
Tipos de Ruido

- El ruido de atributos es más perjudicial que el ruido de clase.
- Eliminar o corregir ejemplos en los conjuntos de datos con ruido de clase o atributos, respectivamente, podría mejorar el rendimiento de clasificación.
- El ruido de atributo es más dañino en aquellos atributos que están muy correlados con la clase.
- La mayoría de los trabajos que podemos encontrar en la literatura se centran solo en ruido de clase.

Mecanismos de introducción de ruido

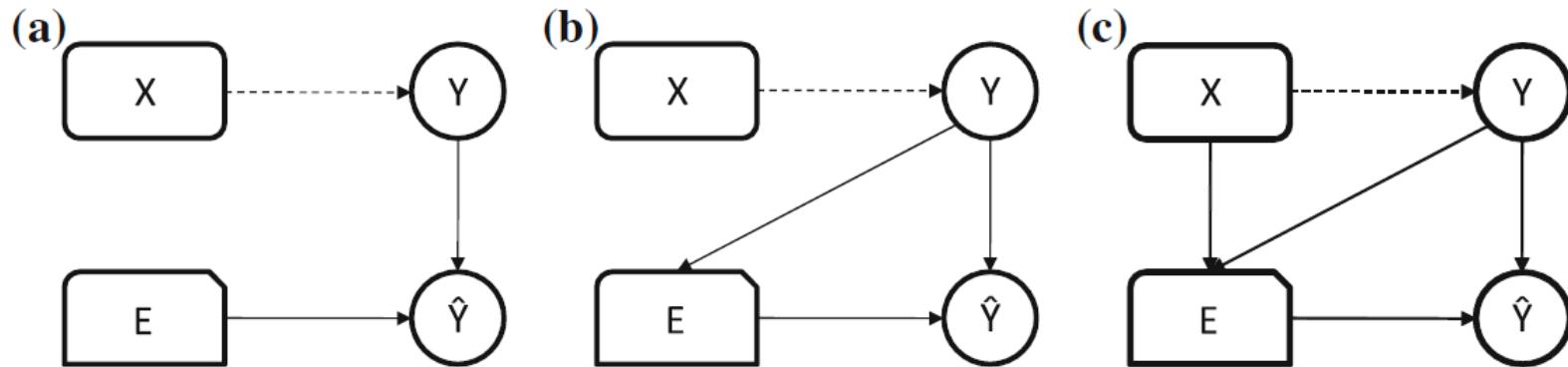
- Tradicionalmente, el mecanismo de introducción de ruido de etiqueta no ha atraído mucho la atención.
- Sin embargo la naturaleza del ruido llega a ser cada vez más importante.
- Los autores distinguen entre tres tipos de modelos estadísticos posibles para el ruido de etiquetas.

Mecanismos de introducción de ruido



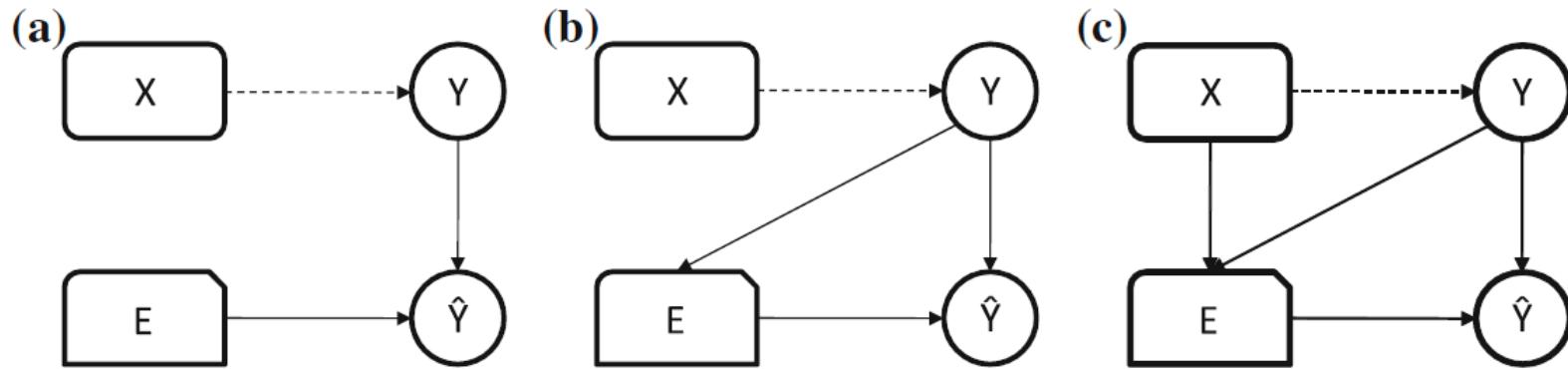
- a) En el caso más simple en el que el procedimiento del ruido no es dependiente del valor real de la clase Y o los valores de atributos de entrada X , el ruido de clase se llama noise completely at random o NCAR.

Mecanismos de introducción de ruido



- b) Las cosas se complican en el modelo noise at random (NAR). Aunque el ruido es independiente de las entradas X, el valor real de la clase lo hace más o menos propenso a ser ruidoso. Este error de etiquetado asimétrico se puede producir por los diferentes costes de extracción de la clase real de las instancias, como en el ejemplo de los estudios sobre casos de control médicos, score financiero, etc.

Mecanismos de introducción de ruido



- c) El tercer y último modelo de ruido es el noisy not at random (NNAR), donde los atributos de entrada afectan de alguna manera a la probabilidad de que las etiquetas de las clases sean erróneas. El modelo NNAR es el más general en el ruido de clase.

Mecanismos de introducción de ruido

- En el caso de ruido de atributo, la modelización descrita anteriormente puede extenderse y adaptarse:
 - Cuando la aparición de ruido no depende en el resto de valores de atributos o en las etiquetas de clases, se aplica el modelo NCAR.
 - Cuando el ruido de atributo depende del valor verdadero pero no en el resto de valores de entrada o el valor observado de clase, el modelo aplicable es NAR.
 - En el último caso (NNAR), la probabilidad de ruido dependerá del valor del atributo pero también del resto de valores de atributos.

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***

Software en R:



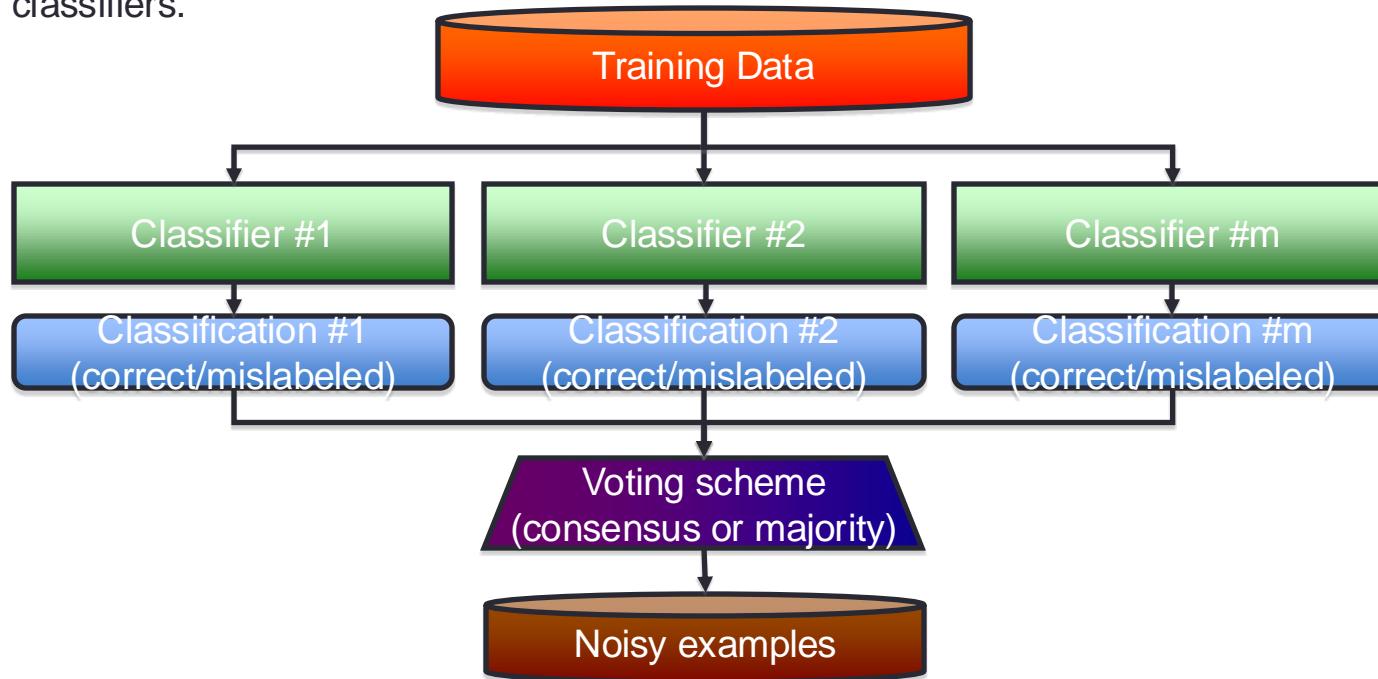
NoiseFiltersR

Package ‘smartdata’

January 19, 2019

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.



Cross-Validated Committees Filter (CVCF)

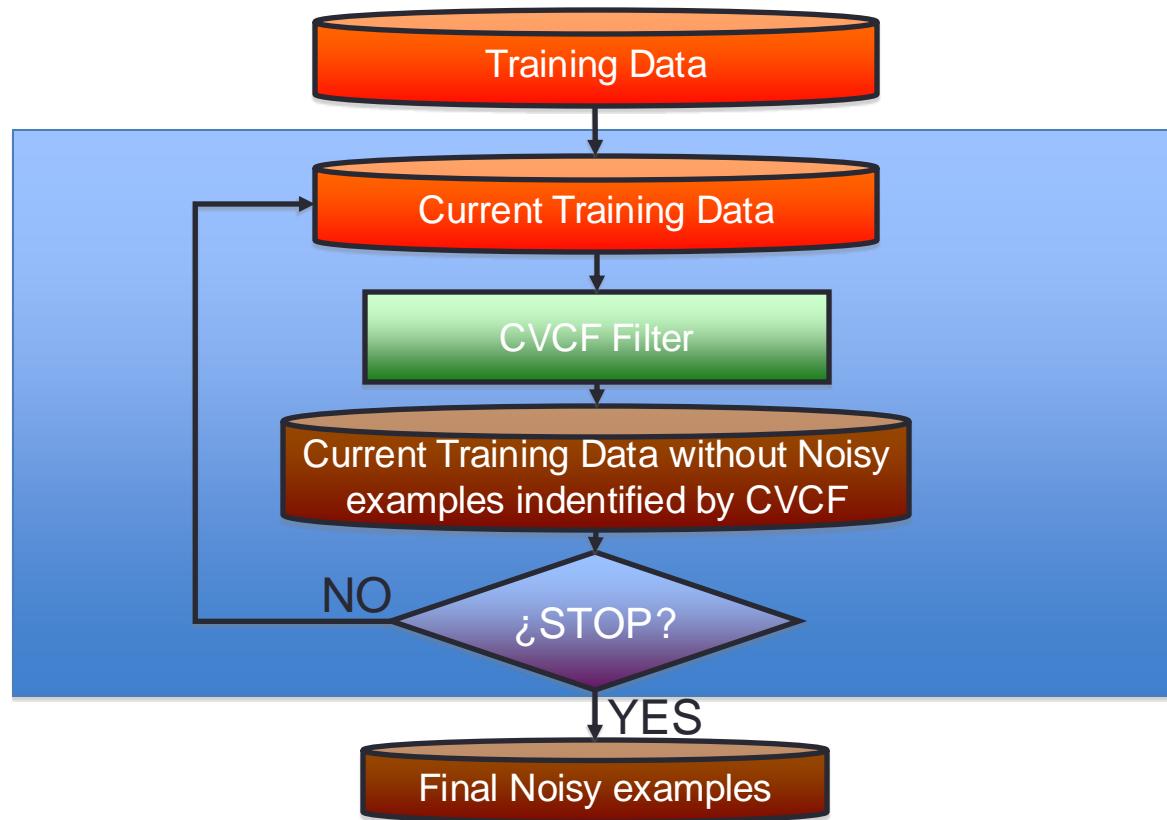
- S. Verbaeten, A.V. Assche. **Ensemble methods for noise elimination in classification problems.** *4th International Workshop on Multiple Classifier Systems (MCS 2003). LNCS 2709, Springer 2003, Guilford (UK, 2003) 317-325.*
- CVCF is similar to EF → two main differences:
 1. **The same learning algorithm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

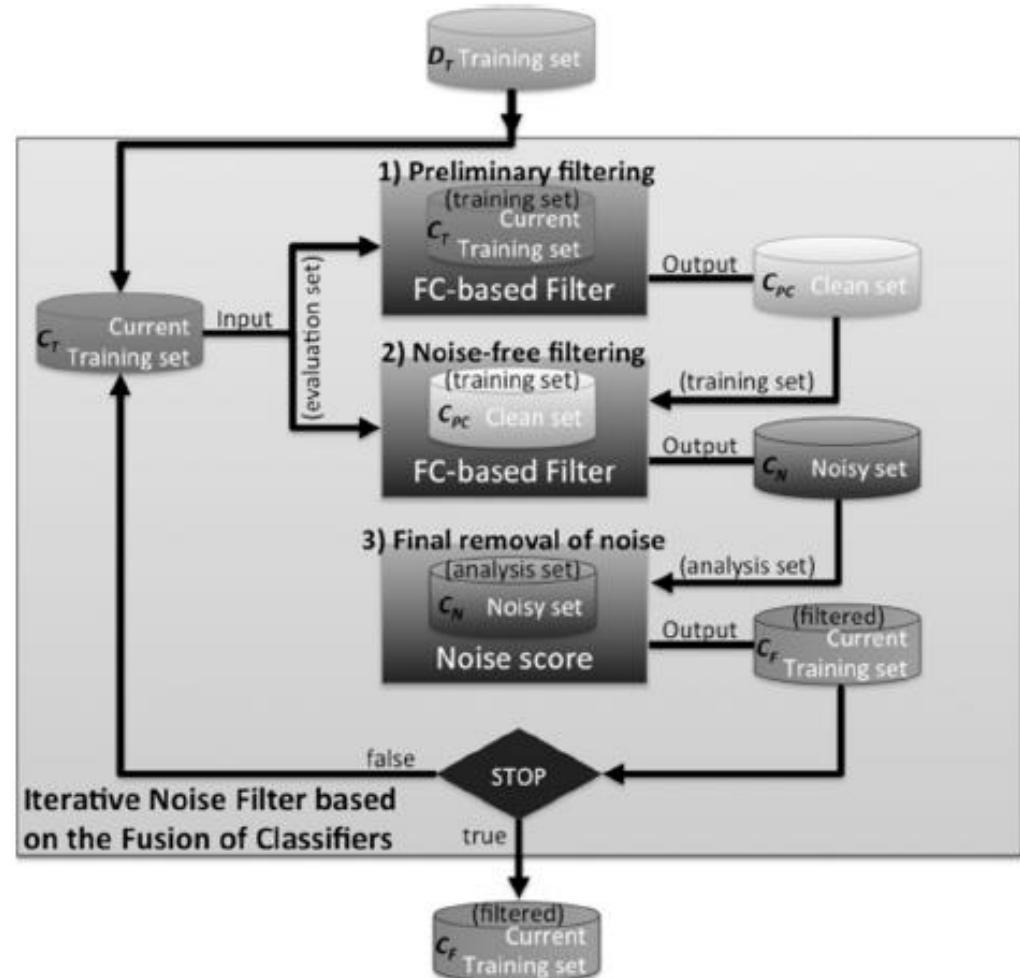
Iterative Partitioning Filter (IPF)

- T.M. Khoshgoftaar, P. Rebours. **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.



INFFC: An iterative class noise filter based on the fusion of classifiers

- INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control. JA Sáez, M Galar, J Luenao, F Herrera. Information Fusion 27, 19-32
- Se realiza un filtrado preliminar con un filtro basado en FC.
- A continuación, se crea otro filtro basado en FC a partir de los ejemplos que no se identifican como ruidosos en el filtrado preliminar para detectar los ejemplos ruidosos en el conjunto completo de instancias de la iteración actual.
- Finalmente, con el fin de controlar la sensibilidad al ruido del filtro, los ejemplos ruidosos sólo se eliminan si superan la medida de la puntuación de ruido.

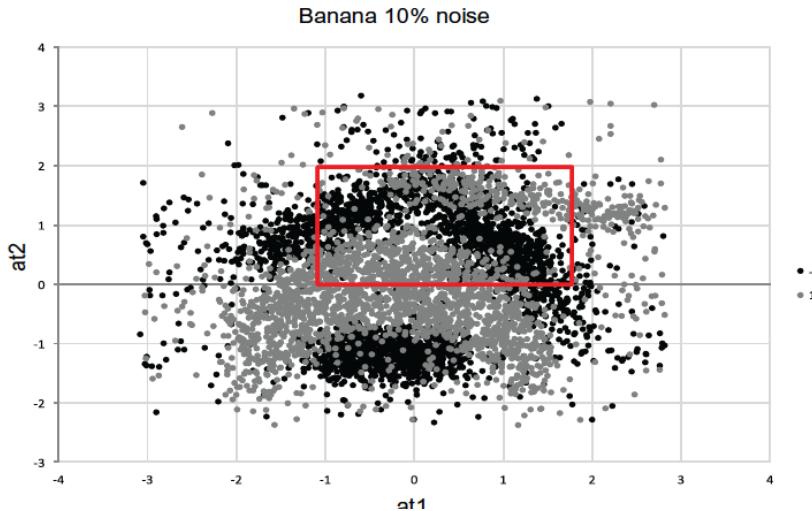


Limpieza de datos con ruido

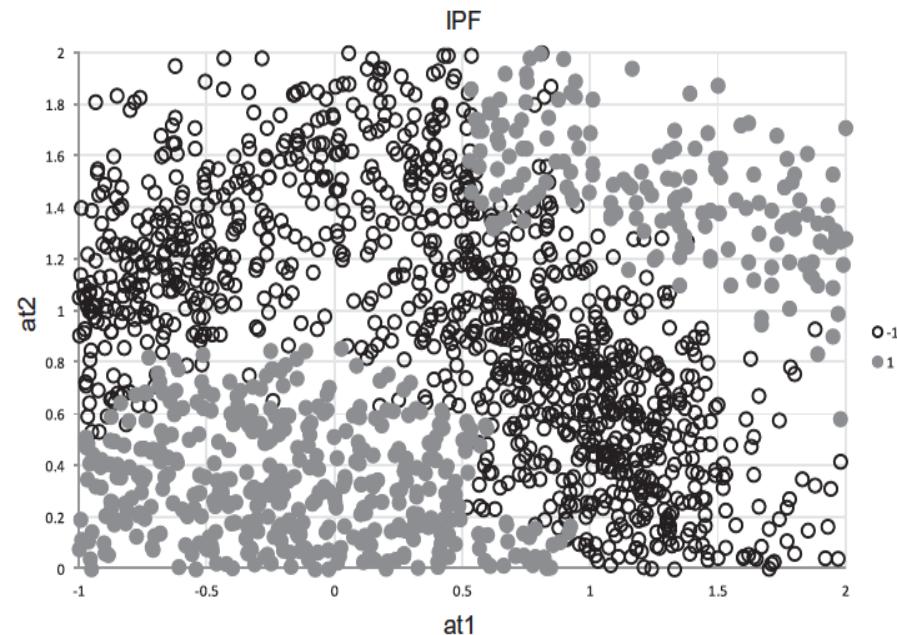
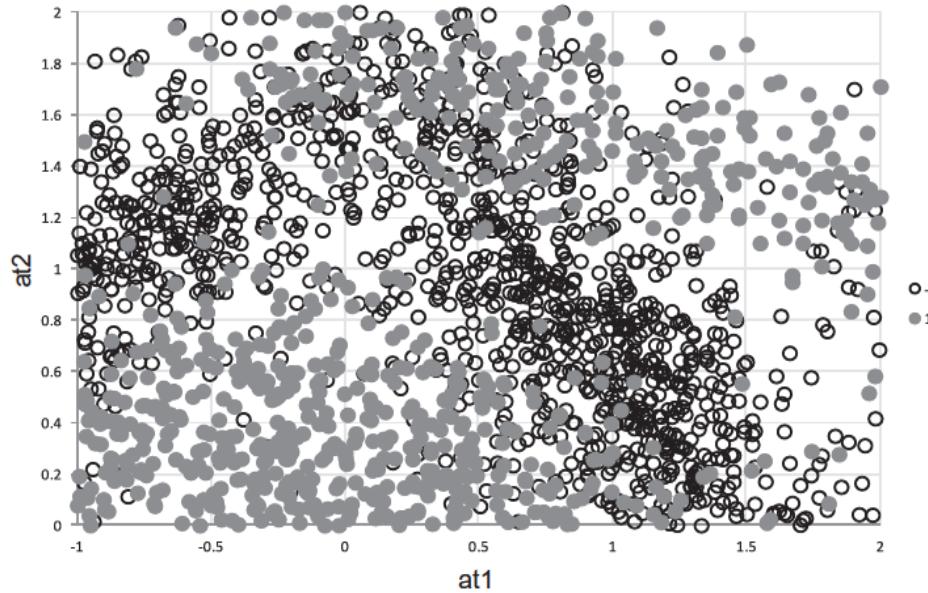
Table 5.2 Filtering of class noise over three classic classifiers

		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

Limpieza de datos con ruido



Banana 10% noise



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 Mislabelled Training Data. Journal of Artificial Intelligence Research 11 (1999) 131-167.
Cross-Validated Committees Filter	CVCommitteesFilter-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. Rebours. Improving software quality prediction by noise filtering techniques. Journal of Computer Science and Technology 22 (2007) 387-396.



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

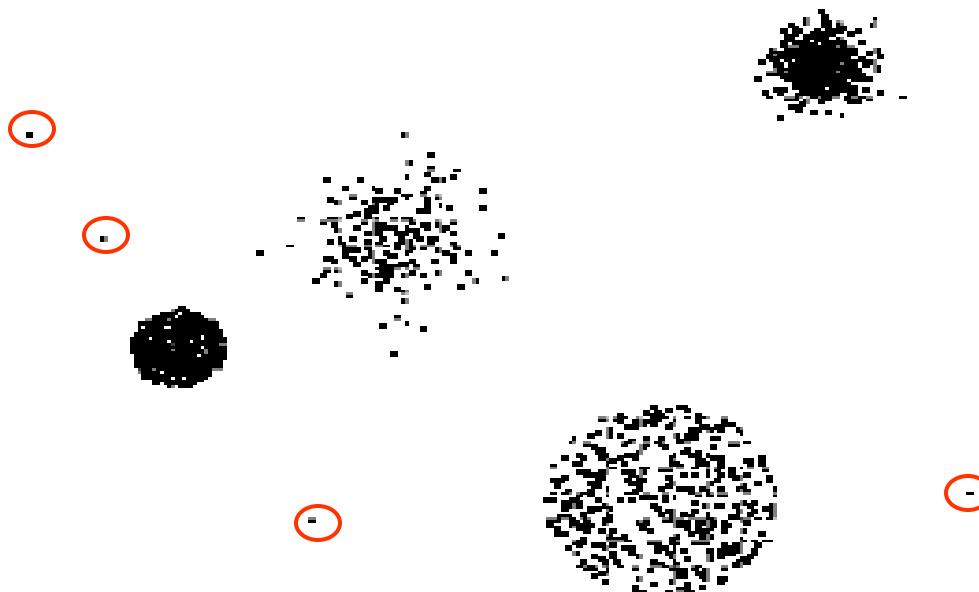
Valores erróneos. Detección de datos anómalos

- Valor erróneo \leftrightarrow valor anómalo.
- El ruido es un error o varianza aleatoria en la medición de una variable.
- ¿Cómo se detectan datos erróneos? Depende del formato y del origen del campo.
 - Nominales \rightarrow valor fuera del formato o rango.
 - Numéricos \rightarrow Buscar datos anómalos y estudiar si son realmente anómalos o erróneos.
- ¿Cómo se elimina el ruido? Mediante técnicas de suavizado.
 - *Binning*: Se suavizan valores ordenados consultando sus vecinos. Los valores se distribuyen en un conjunto de cajas o intervalos (*bins*). Realiza un suavizado local.
Variantes:
 - *Binning* uniforme en los intervalos (*equiwidth*) o en el contenido (*equidepth*)
 - Suavizar por la media o mediana.
 - Suavizar por las fronteras.
 - Regresión: Los datos se suavizan ajustándolos a una función con técnicas de regresión.

Valores erróneos. Detección de datos anómalos

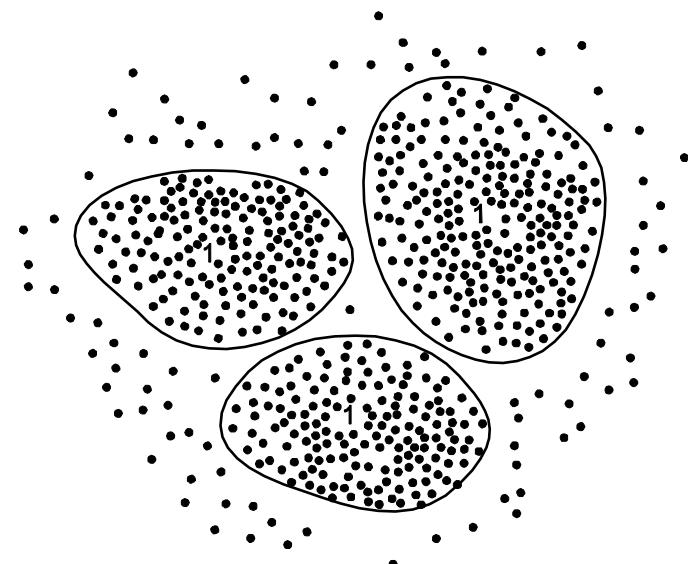
Outliers

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



Valores erróneos. 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.



Valores erróneos. Detección de datos anómalos

- Combinación de inspección humana y automática.
Utilizar técnicas automatizadas (p.e. basadas en la teoría de la información) para identificar casos “extraños” y el experto humano trabaja sólo sobre estos datos.
- La no detección de un dato anómalo puede ser un problema importante si el atributo se normaliza posteriormente, ya que la mayoría de datos estarán en un rango pequeño y puede haber poca precisión o sensibilidad para algunos métodos de DM.

Valores erróneos. Detección de datos anómalos

- Tratamiento de valores anómalos o erróneos:
 - 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.

Preprocesamiento de Datos

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

Reducción de Datos

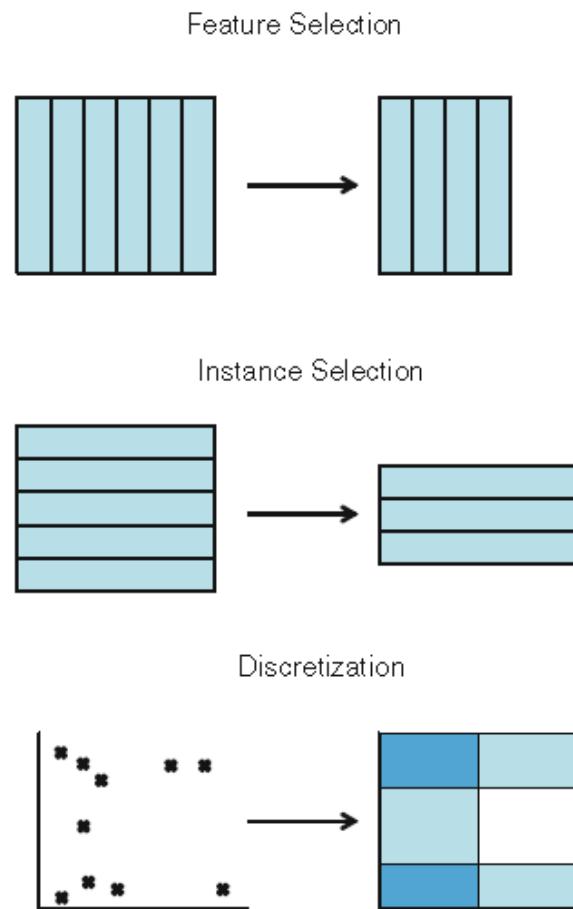


Fig. 1.4 Forms of data reduction

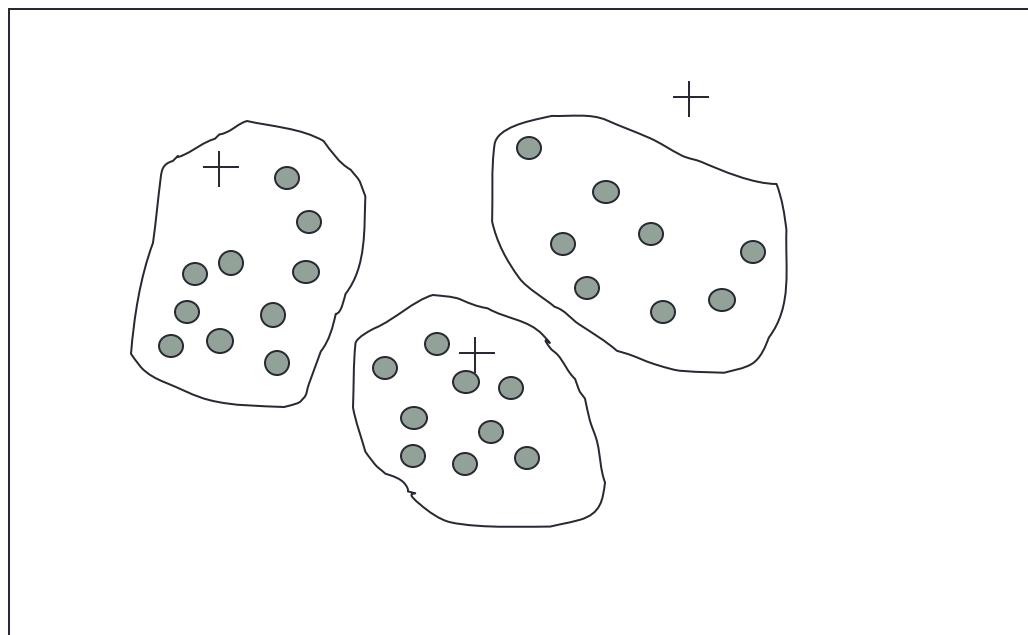
Reducción de Datos

- ✿ Diferentes vías para la Reducción de Datos:
 - ❖ Selección/Extracción de Características
 - ❖ Selección de Instancias
 - ❖ Discretización

Agrupamiento/Compactación/Condensación

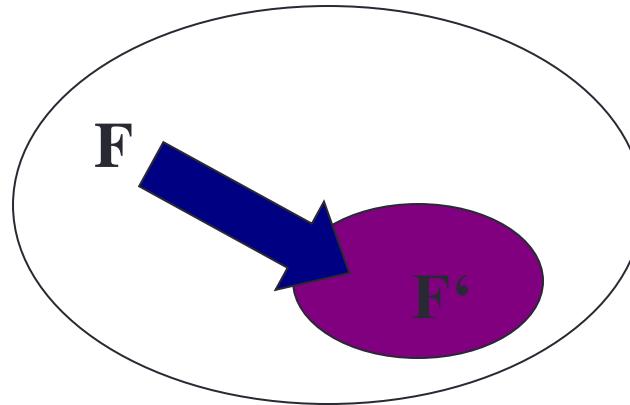
Condensación y Compactación de datos

Por Agrupamiento: Ejemplo.



Feature Selection / - Extraction

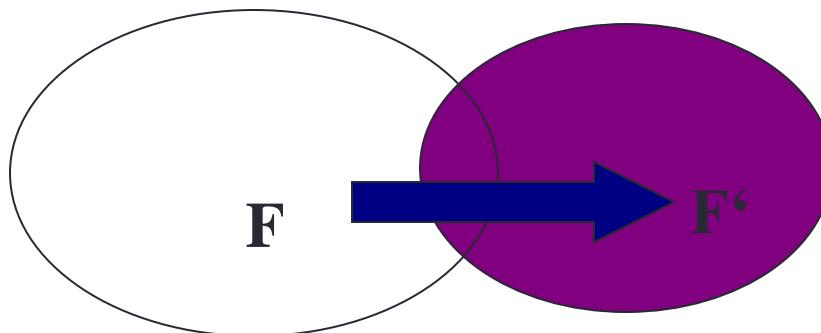
- Feature Selection:



$$\{f_1, \dots, f_i, \dots, f_n\} \xrightarrow{f.selection} \{f_{i_1}, \dots, f_{i_j}, \dots, f_{i_m}\}$$

$i_j \in \{1, \dots, n\}; j = 1, \dots, m$
 $i_a = i_b \Rightarrow a = b; a, b \in \{1, \dots, m\}$

- Feature Extraction/Creation



$$\{f_1, \dots, f_i, \dots, f_n\} \xrightarrow{f.extraction} \{g_1(f_1, \dots, f_n), \dots, g_j(f_1, \dots, f_n), \dots, g_m(f_1, \dots, f_n)\}$$

Reducción de la Dimensionalidad

La “Maldición” de la Dimensionalidad

La dimensionalidad de los datos llega a ser un obstáculo serio para la eficiencia de la mayoría de los algoritmos de aprendizaje y minería de datos.

Se estima que conforme el número de dimensionales (variables) crece, el tamaño de la muestra de datos requiere también crecer exponencialmente para tener un estimador efectivo de las densidades multivariadas.

Reducción de la Dimensionalidad: Se responsabiliza de la reducción del número de variables en los conjuntos de datos evitando importantes pérdidas de información.

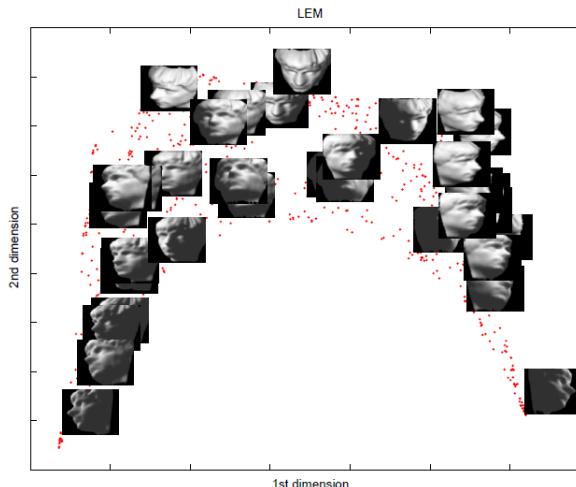
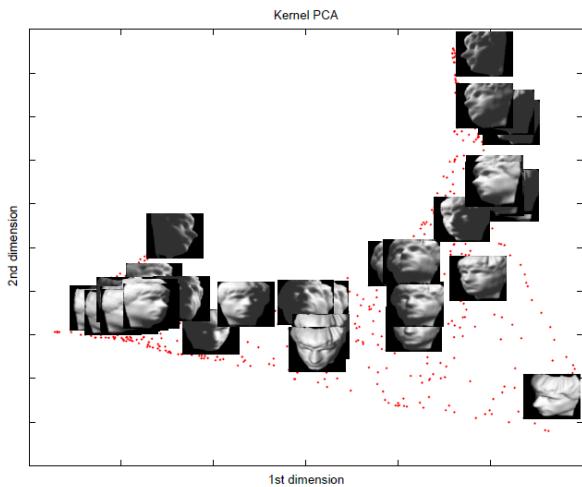
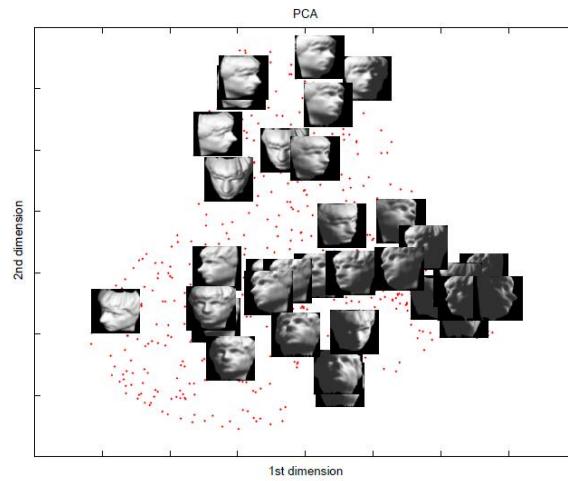
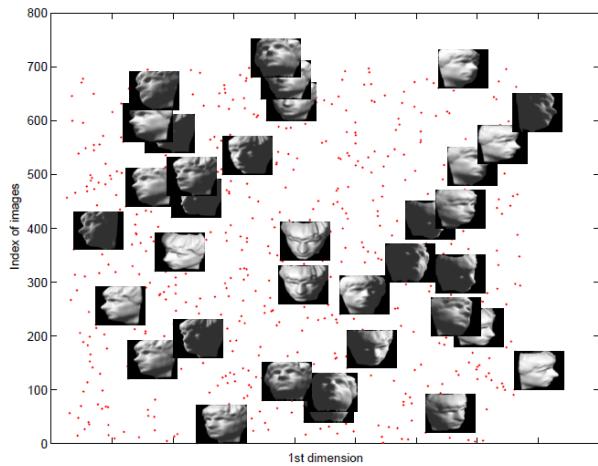
Reducción de la Dimensionalidad

- Dimensionality reduction vs. manifold learning
- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- Multidimensional Scaling (MDS)
- Deep Learning - Autoencoders

Dimensionality Reduction vs. Manifold Learning

- Nombres intercambiables
- Representan datos en un espacio de menor dimensión
- Aplicaciones
 - Visualización de datos
 - Preprocesamiento para aprendizaje supervisado

Ejemplos

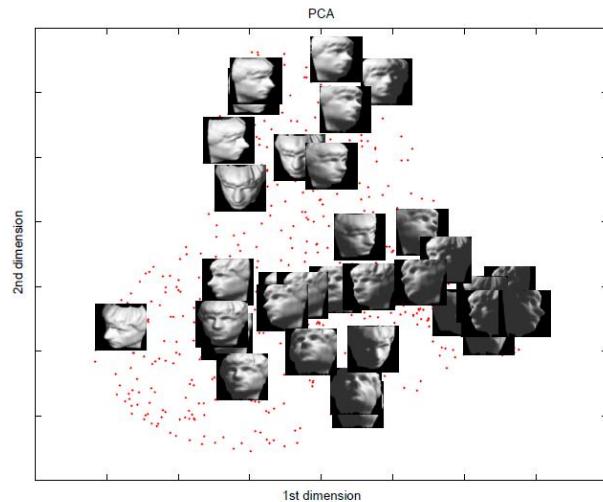
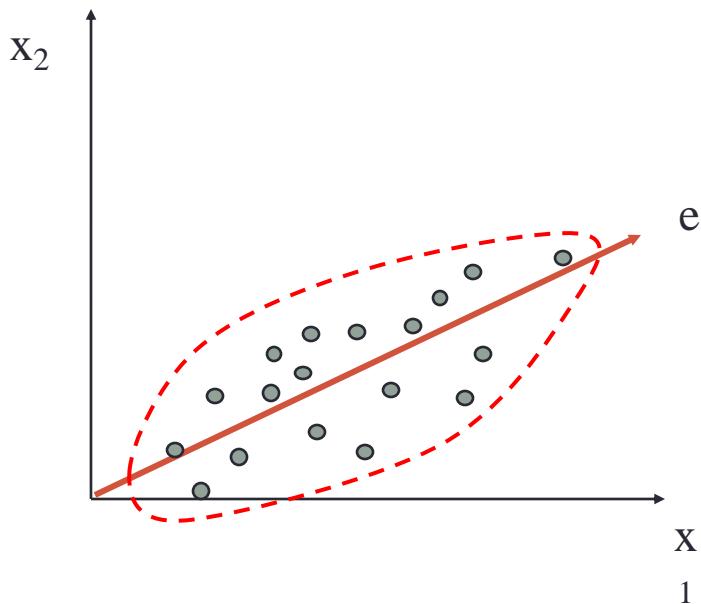


Modelos

- Métodos Lineales
 - Principal component analysis (PCA)
 - Multidimensional scaling (MDS)
 - *Independent component analysis (ICA)*
- Métodos no Lineales
 - Kernel PCA
 - Locally linear embedding (LLE)
 - *Laplacian eigenmaps (LEM)*
 - *Semidefinite embedding (SDE)*
 - Autoencoders

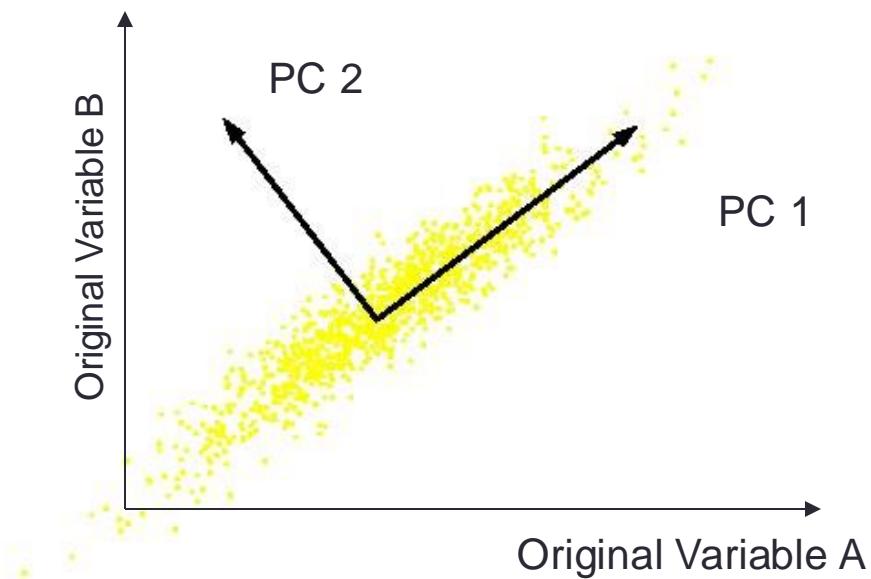
Principal Component Analysis (PCA)

- Historia: Karl Pearson, 1901
- Encontrar las proyecciones que capturen las mayores cantidades de variación en los datos
- Encontrar los autovectores de la matriz de covarianza, y utilizarlos para definir el nuevo espacio,



PCA

- Definición: Dado un dataset $X \in R^{d \times N}$, encontrar los ejes principales que serán aquellos ejes ortogonales en los que la varianza observada en la proyección es máxima.



PCA: Algoritmo

Algorithm 1 Direct PCA Algorithm

Input: Given data $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^d$;

Recover basis: Calculate $XX^\top = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top$ and U as eigenvectors of XX^\top for the top k eigenvalues.

Encode training data: $Y = U^\top X$, where Y is a $k \times N$ matrix of encodings of the original data.

Reconstruct training data: $\hat{X} = UY = UU^\top X$.

Encode test data: $y = U^\top x$, where y is a k -dimensional encoding of x .

Reconstruct test data: $\hat{x} = Uy = UU^\top x$.

Kernel PCA

- Historia: S. Mika et al, NIPS, 1999
- *Data may lie on or near a nonlinear manifold, not a linear subspace*
- Encontrar componentes principales que no son lineales al espacio de entrada a través de un mapeado linear

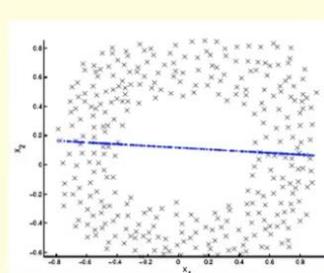
$$\Phi : x \rightarrow \mathcal{H} \quad x \mapsto \Phi(x)$$

- Objetivo

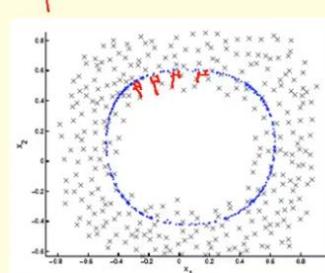
$$\min_{U_k} \sum_{i=1}^N \|\Phi(\mathbf{x}_i) - U_k U_k^T \Phi(\mathbf{x}_i)\|^2$$

- Solución encontrada por SVD: $\Phi(X) = U\Sigma V^T$

U contiene los autovectores de $\Phi(X)\Phi(X)^T$



PCA



KPCA

Locally Linear Embedding (LLE)

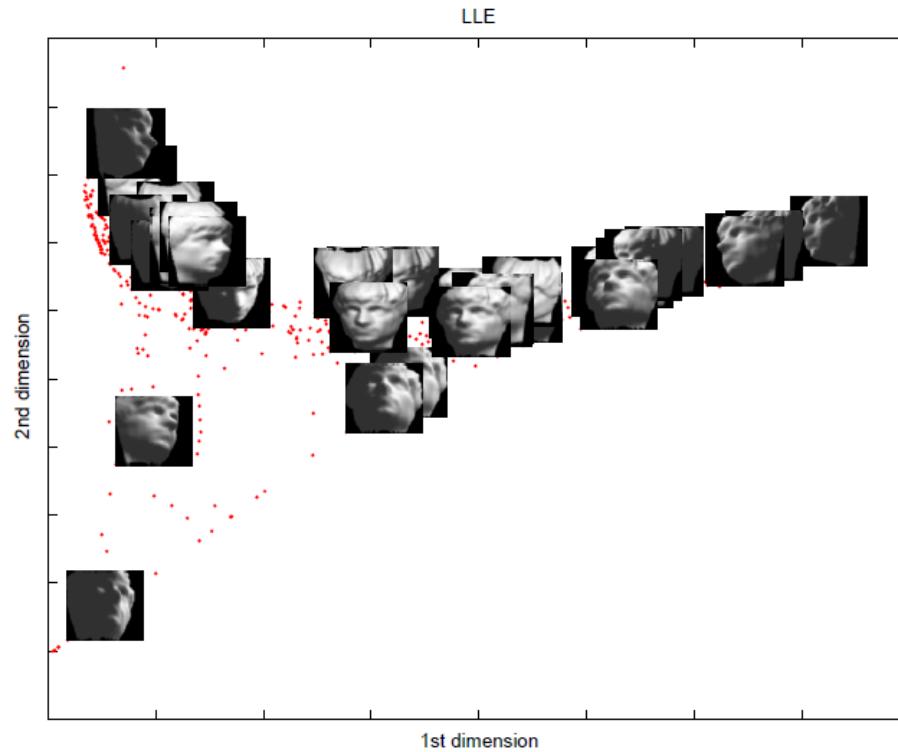
- Historia: S. Roweis and L. Saul, Science, 2000
- Procedimiento
 1. Identificar los vecinos de cada punto
 2. Calcular los pesos que mejor reconstruye linealmente el punto desde su propios vecinos

$$\min_{\mathbf{w}} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{j=1}^k w_{ij} \mathbf{x}_{N_i(j)} \right\|^2$$

- 3. Encontrar el vector dimensional reducido que es mejor reconstruido por los pesos del paso 2

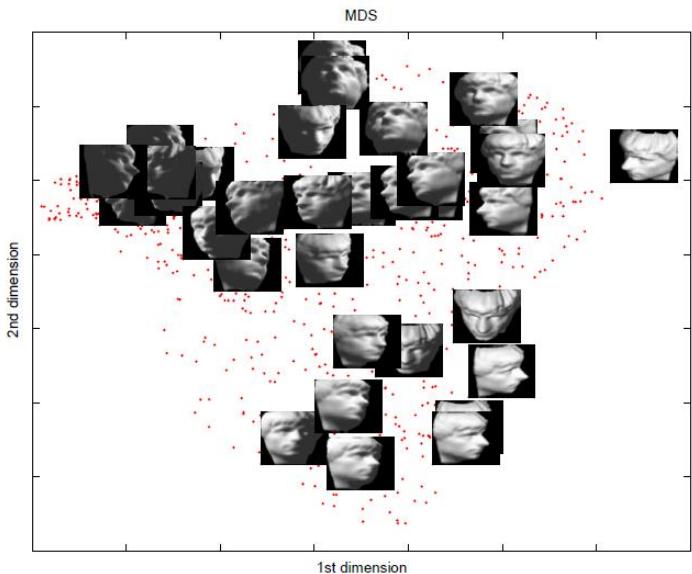
$$\min_Y \sum_{i=1}^N \left\| \mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_{N_i(j)} \right\|^2 \iff \min_Y \text{tr}(Y^\top Y L)$$

LLE Example



Multidimensional Scaling (MDS)

- Historia: T. Cox and M. Cox, 2001
- Procede a preservar las distancias por parejas de puntos
- Diferente formulación que PCA, pero produce resultados muy similares



$$\min_Y \sum_{i=1}^N \sum_{j=1}^N (d_{ij}^{(X)} - d_{ij}^{(Y)})^2$$

where $d_{ij}^{(X)} = \|x_i - x_j\|^2$ and $d_{ij}^{(Y)} = \|y_i - y_j\|^2$.

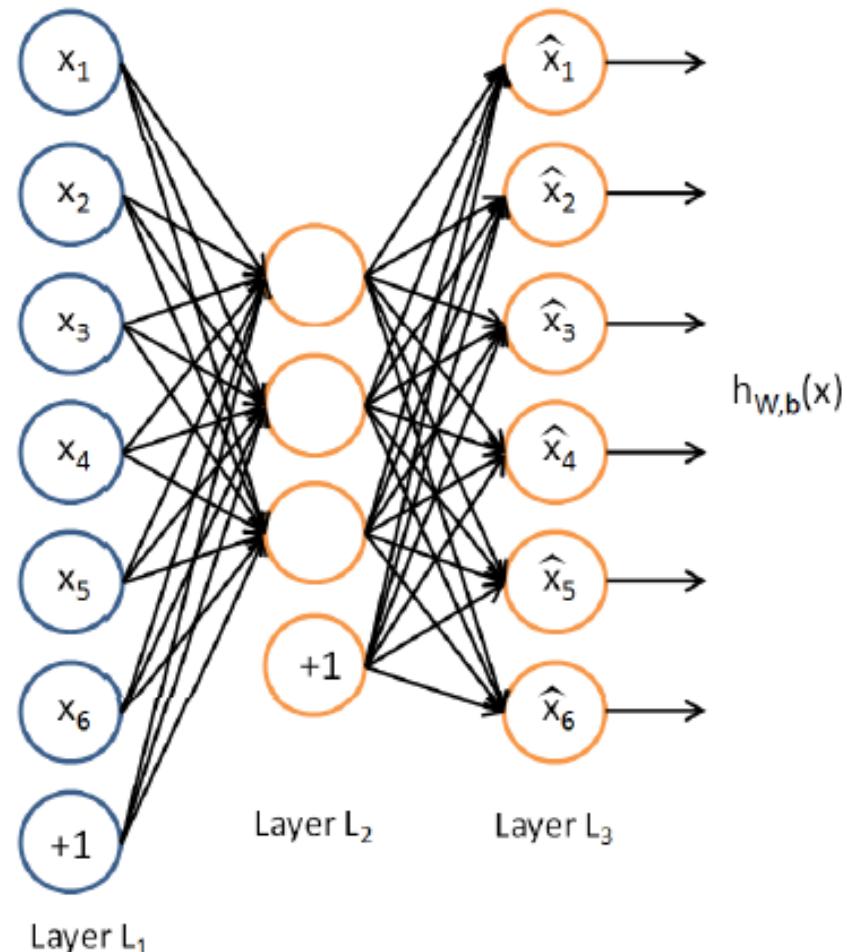
$$X^\top X = -\frac{1}{2} H D^{(X)} H \quad \text{where } H = I - \frac{1}{N} \mathbf{1} \mathbf{1}^\top.$$

$$\min_Y \sum_{i=1}^N \sum_{j=1}^N (x_i^\top x_j - y_i^\top y_j)^2$$

AutoEncoders

An **autoencoder** neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.

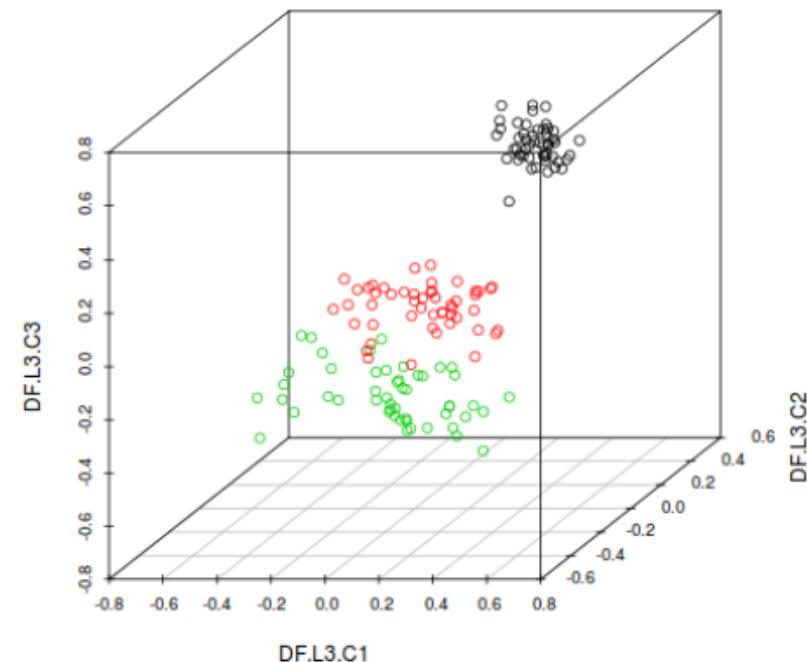
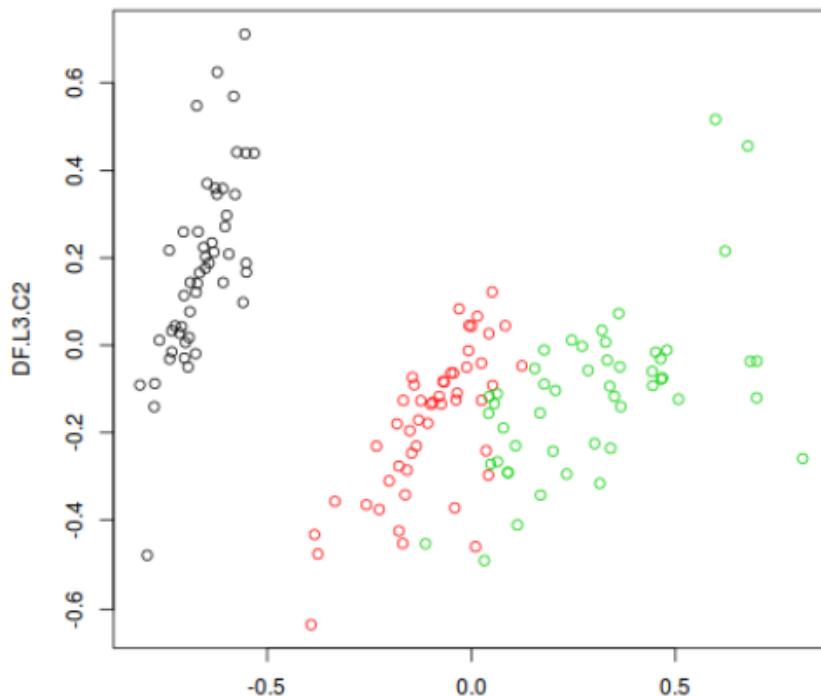
The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction.



AutoEncoders

- **Autoencoder** (autoencoder de h2o, salida de la capa intermedia capas internas de [8, 5, 3, 5, 8] neuronas y 100 "epochs" (el tridimensional), y [8, 5, 2, 5, 8] neuronas con 1000 "epochs" (el bidimensional)).

setosa, versicolor y virginica



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.

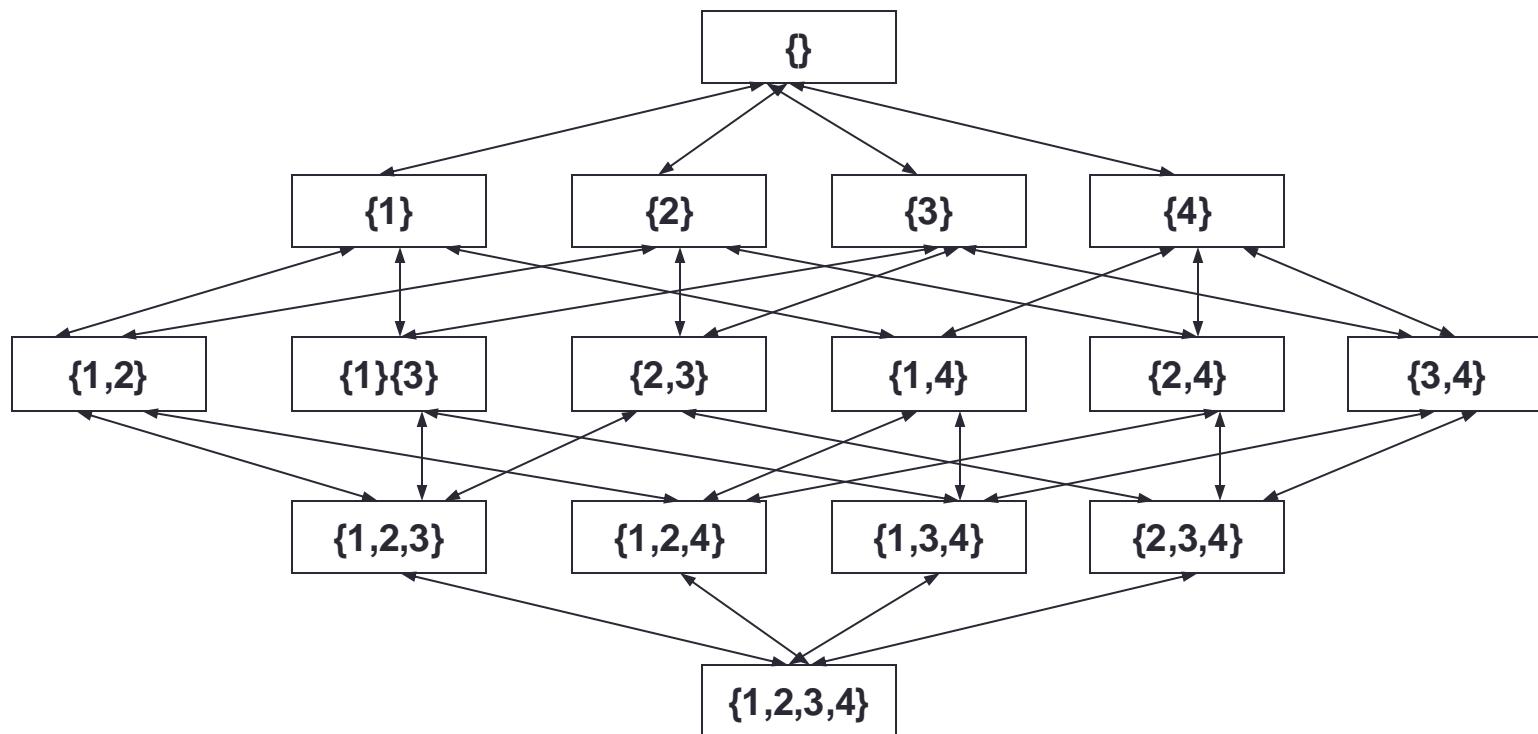
Var. 5

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	0	1	0

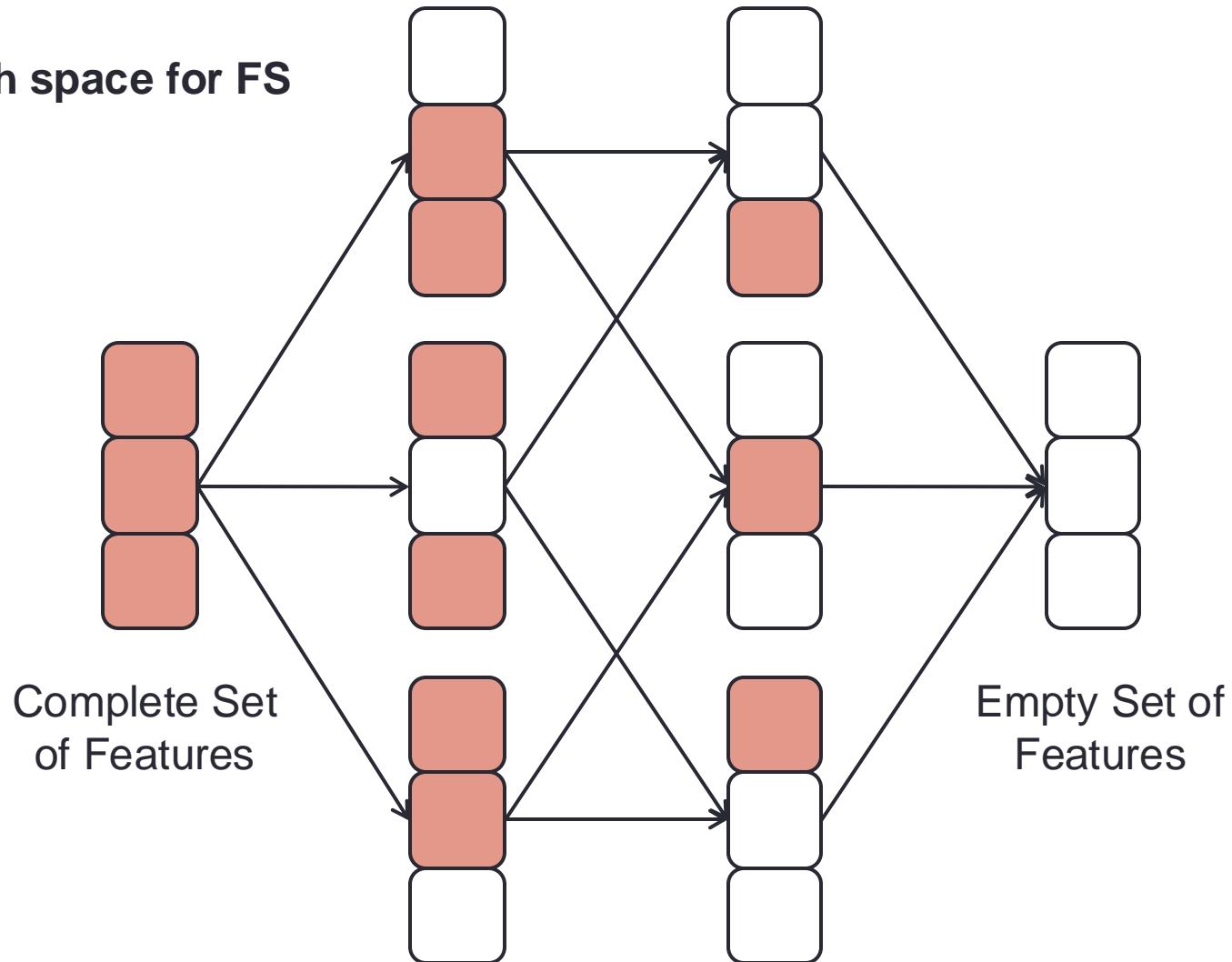
Selección de Características

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.

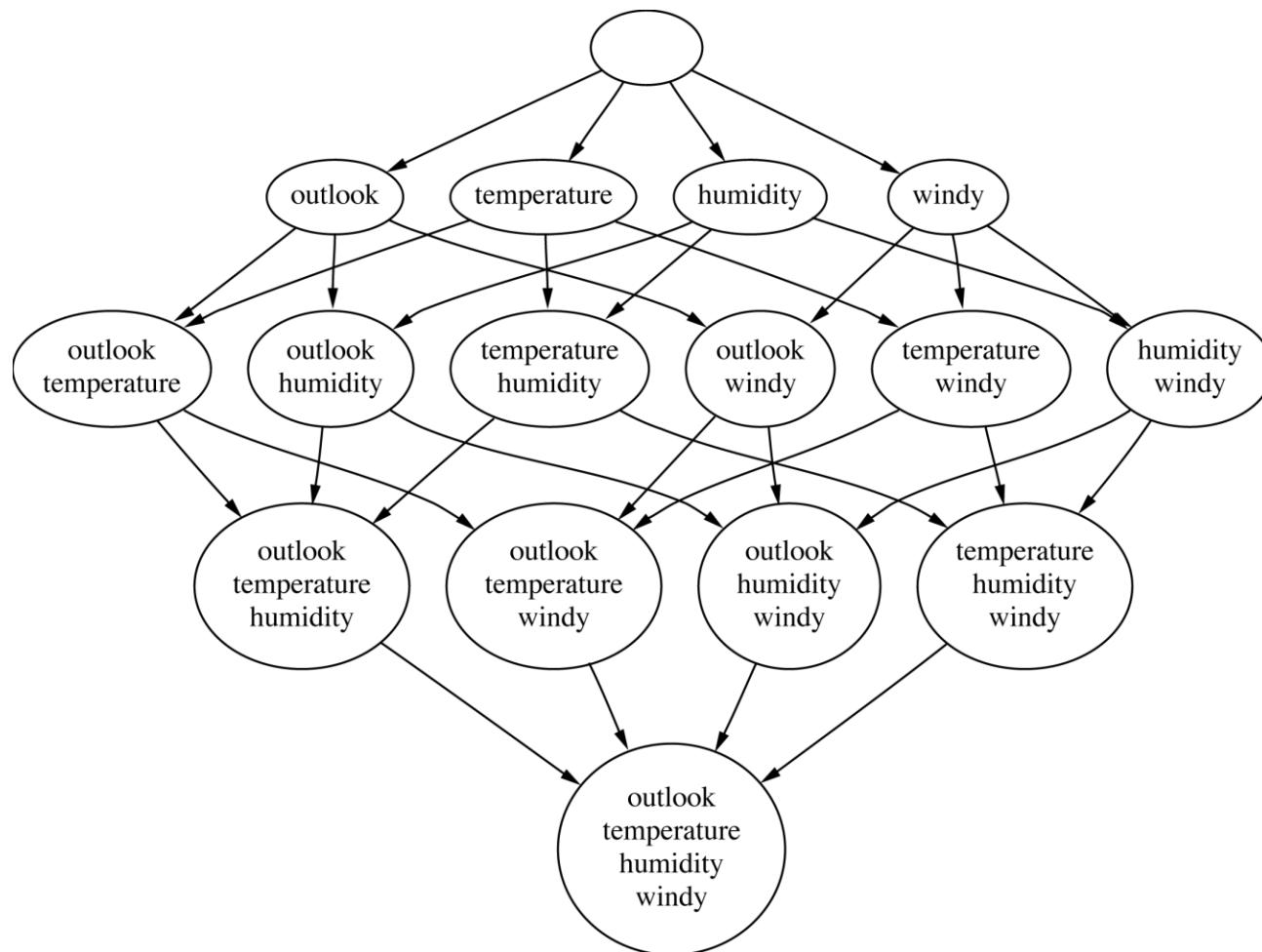
```
function SFG( $F$  - full set,  $U$  - measure)
    initialize:  $S = \emptyset$ 
    repeat
         $f = \text{FINDNEXT}(F)$ 
         $S = S \cup \{f\}$ 
         $F = F - \{f\}$ 
    until  $S$  satisfies  $U$  or  $F = \emptyset$ 
    return  $S$ 
end function
```

Algorithm 2 Sequential backward feature set generation - SBG.

```
function SBG( $F$  - full set,  $U$  - measure)
    initialize:  $S = \emptyset$ 
    repeat
         $f = \text{GETNEXT}(F)$ 
         $F = F - \{f\}$ 
         $S = S \cup \{f\}$ 
    until  $S$  does not satisfy  $U$  or  $F = \emptyset$ 
    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 $I(X_{1,\dots,M}; C)$

$$\begin{aligned} f(X_{1,\dots,M}) &= I(X_{1,\dots,M}; C) = H(C) - H(C|X_{1,\dots,M}) = \\ &\sum_{c=1}^{|C|} \int_{X_{1,\dots,M}} P(X_{1\dots M}, \omega_c) \log \frac{P(X_{1\dots M}, \omega_c)}{P(X_{1\dots M})P(\omega_c)} dx \end{aligned}$$

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 $X_{1\dots M}$
- Por la complejidad de cálculo de I se suelen utilizar reglas heurísticas

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

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

- **Envolventes:**
 - 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

■ Envolventes:

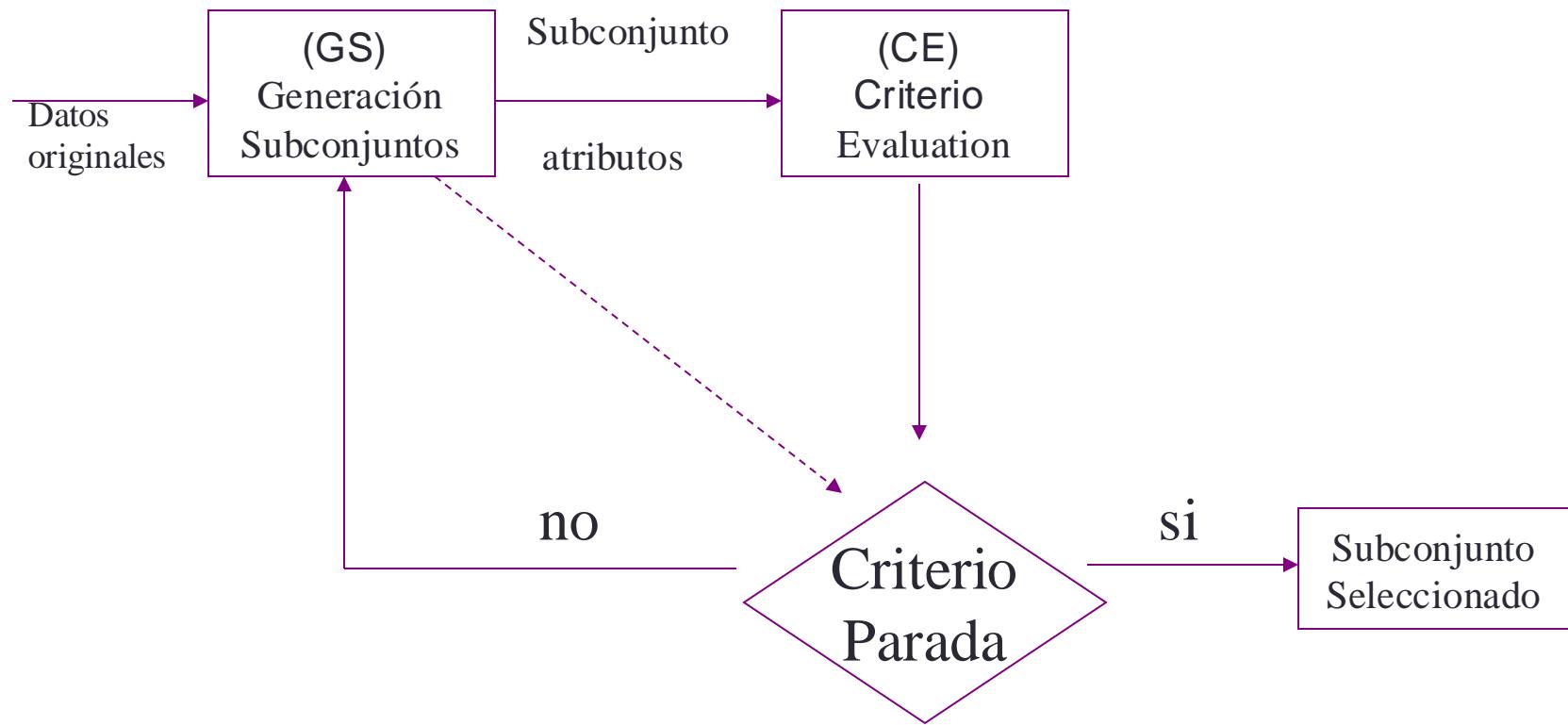
- Muy costosos: para cada evaluación hay que aprender un modelo y validararlo. No es factible para clasificadores costosos
- Pérdida de generalidad: La solución está sesgada hacia el clasificador utilizado

■ 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

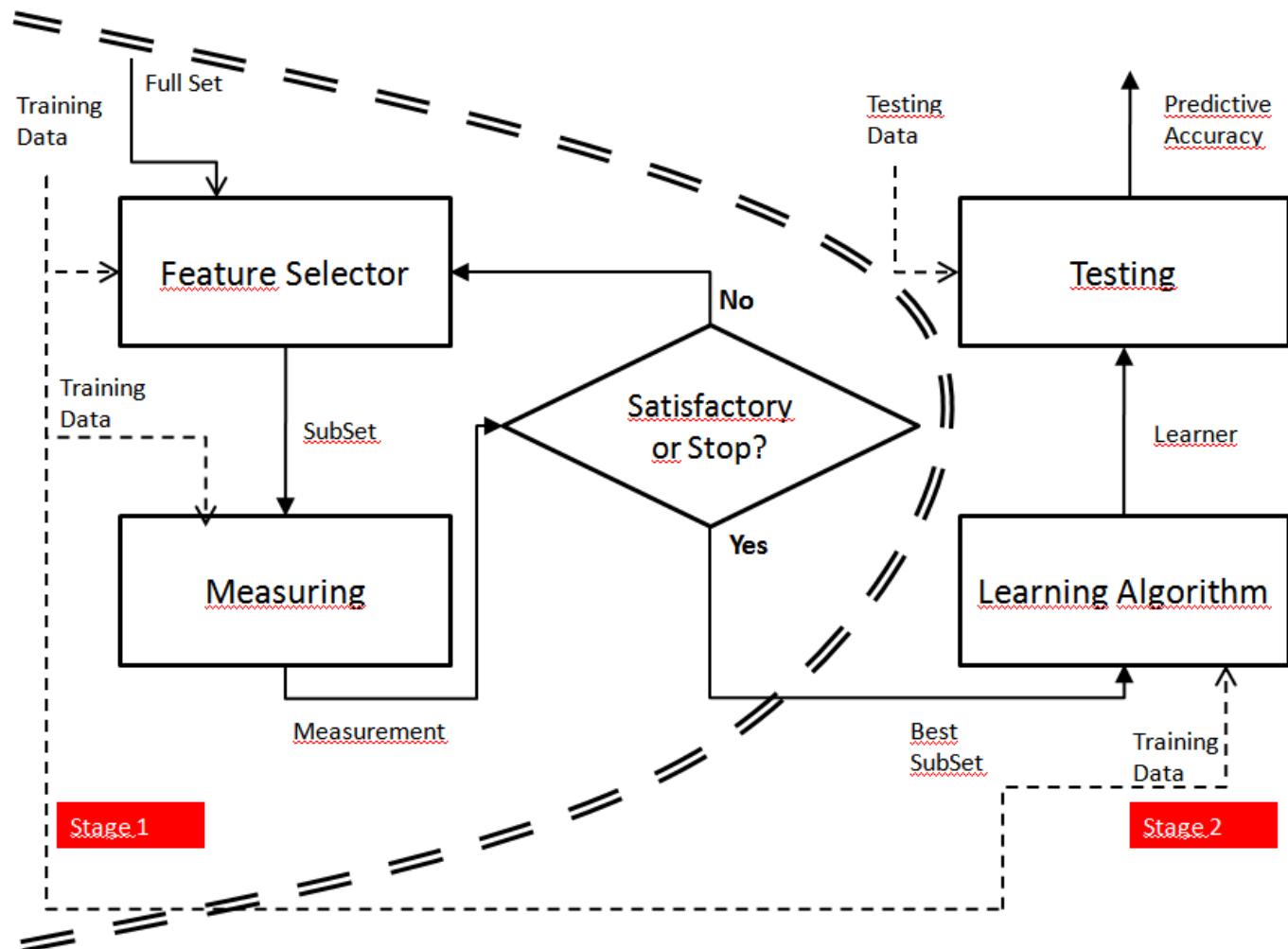


Fig. 7.2 A filter model for FS

Selección de Características

Proceso

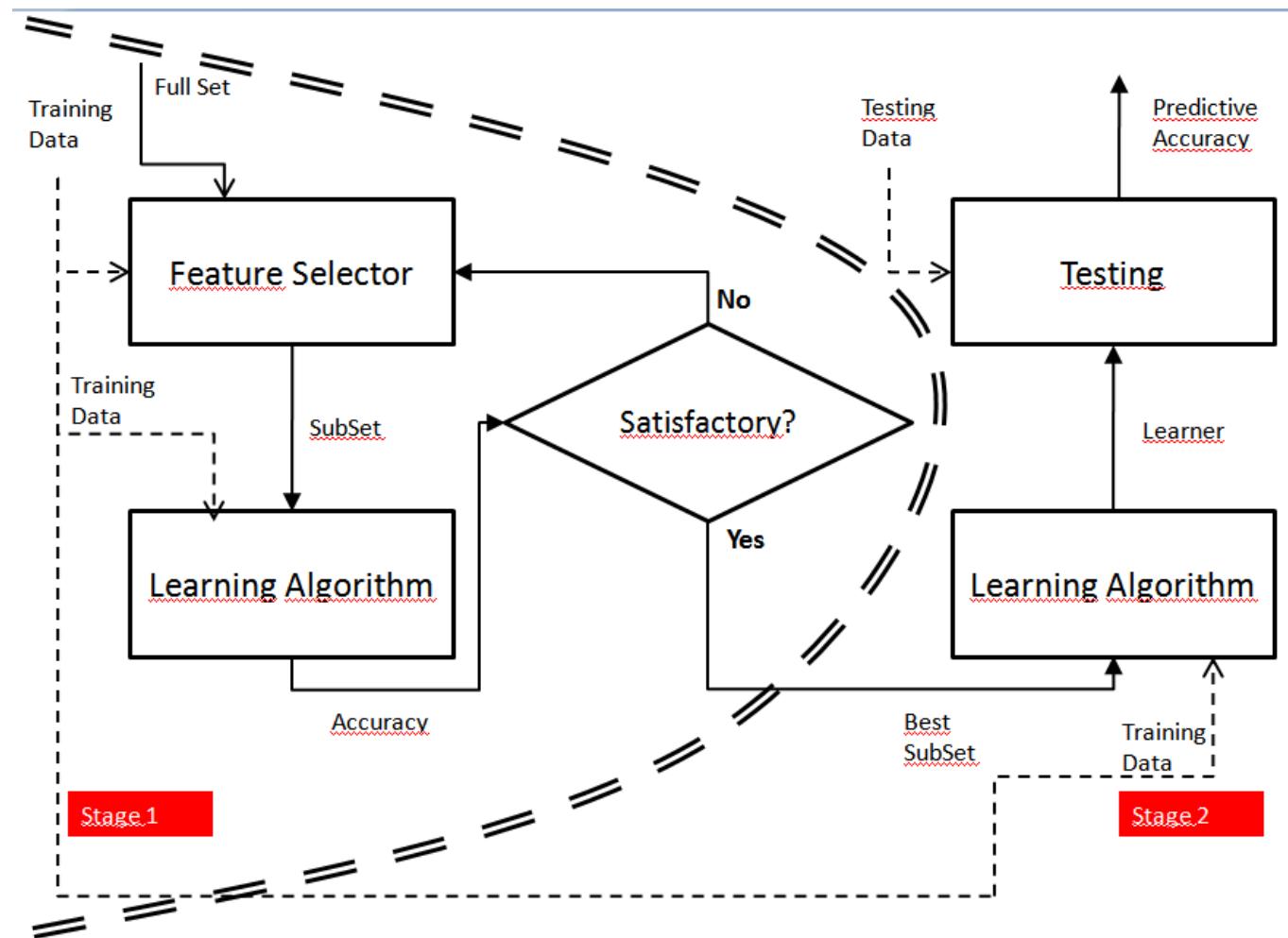


Fig. 7.2 A wrapper model for FS

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, attrs 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, attrs más relevantes al principio

Selección de Características

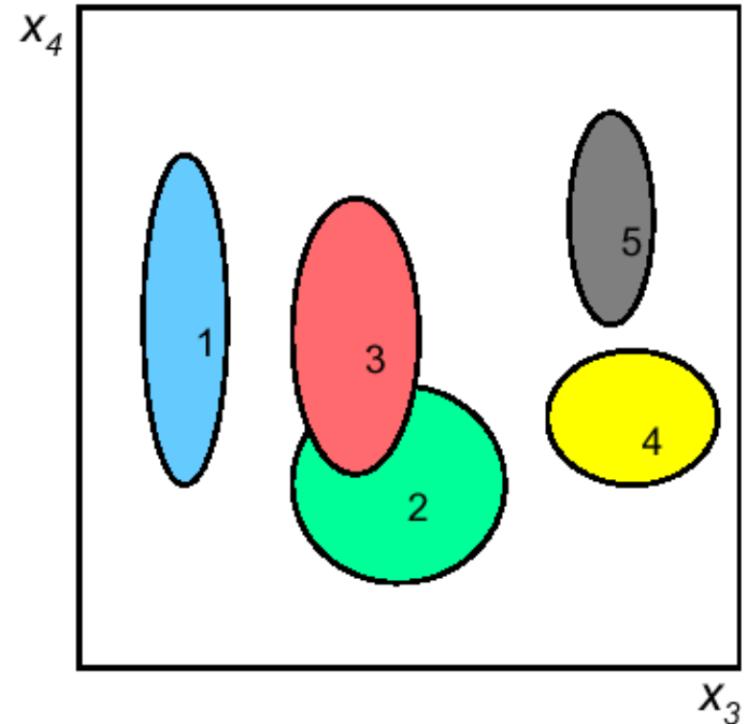
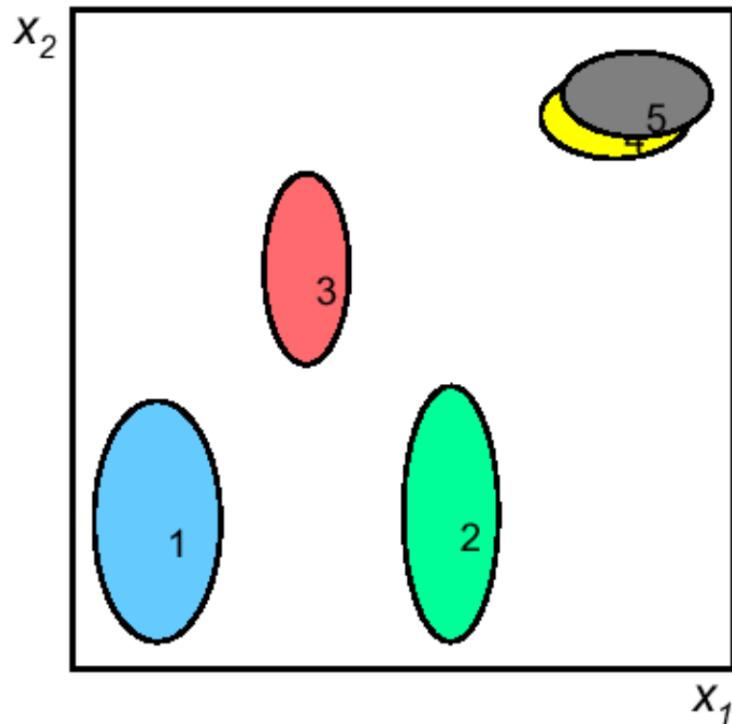
Validación algoritmos ranking

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

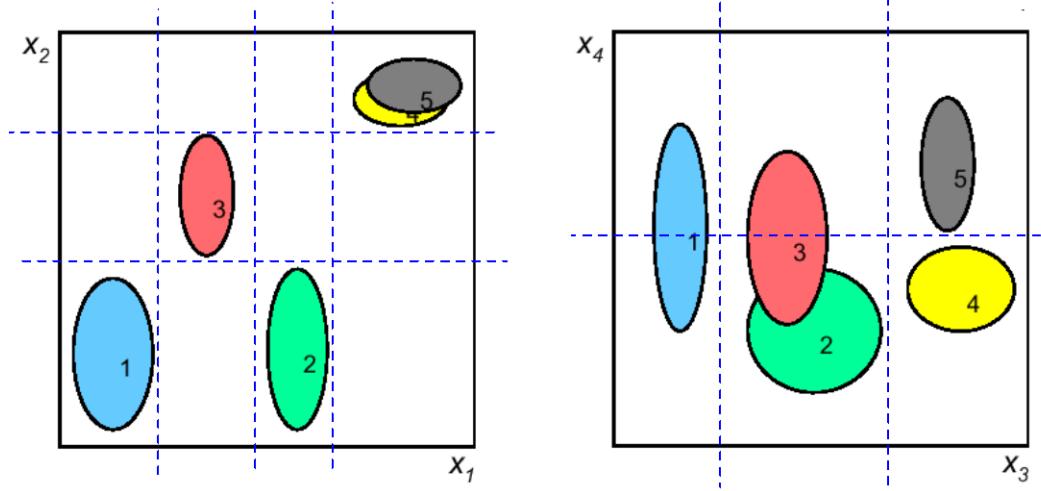
Algunos algoritmos.

¿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.

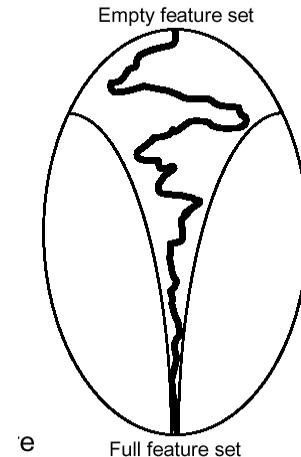
Selección hacia delante

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.

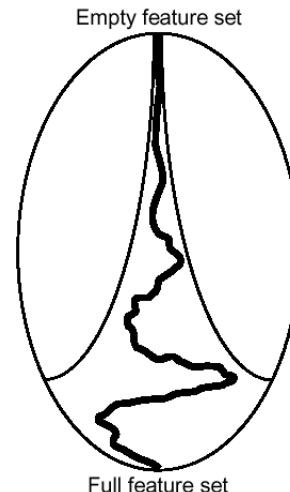
Selección hacia atrás

- 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

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

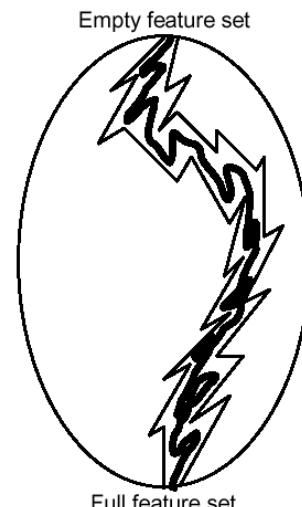
$$S = S \cup \{X^+\}$$

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

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

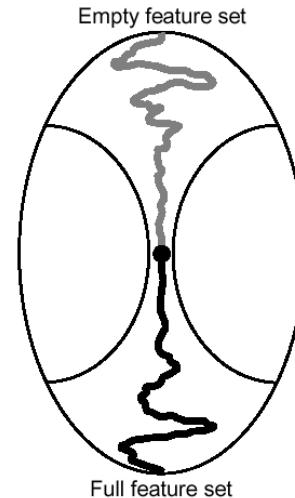
$$S_F = S_F \cup \{X^+\}$$

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

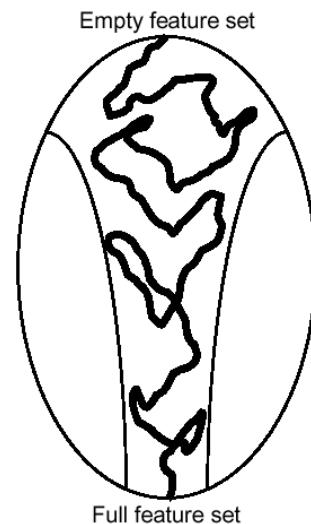
$$X^+ = \arg \max_{X \in U - S} f(S \cup X)$$

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

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

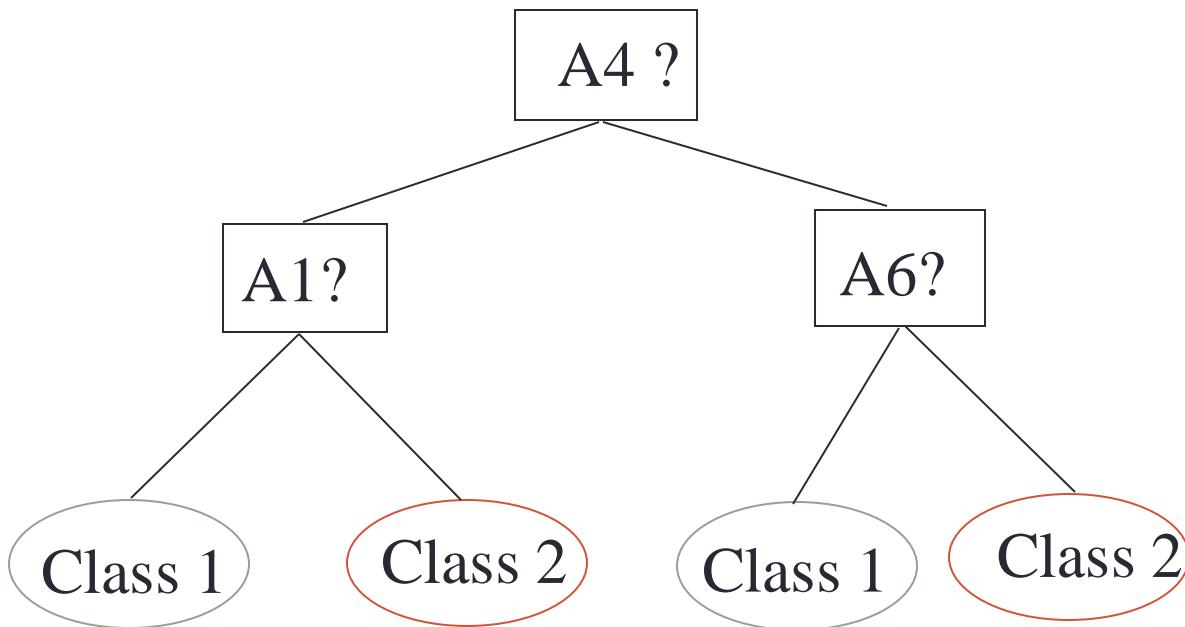


Selección de Características

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

Fselector ([varias formas de selección tanto filter como wrapper](#))

Software en R:



Caret

Boruta

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  $S$ 
```

Selección de Características

Algorithm 9 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 10 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 = {}
Abest = ESTIMATE(D,F,LA)
for maxTries iterations do
    S = RANDOMSET(seed)
    A = ESTIMATE(D,S,LA)                                ▷ # - the cardinality of S
    if A > Abest then
        Sbest = S
        Abest = A
        L = {S}                                         ▷ L is reinitialized
    else if A = Abest then
        L = APPEND(S,L)
    end if
end for
return L                                              ▷ all equivalently good subsets found by LVW
end function
```

Selección de Características

Algorithm 11 Relief algorithm.

```
function RELIEF( $\mathbf{x}$  - features,  $m$  - number of instances sampled,  $\tau$  - relevance threshold)
    initialize:  $\mathbf{w} = 0$ 
    for  $i = 1$  to  $m$  do
        randomly select an instance  $I$ 
        find nearest-hit  $H$  and nearest-miss  $J$ 
        for  $j = 1$  to  $M$  do
             $\mathbf{w}(j) = \mathbf{w}(j) - dist(j, I, H)^2/m + dist(j, I, J)^2/m$      $\triangleright dist$  is a distance function
        end for
    end for
    return  $\mathbf{w}$  greater than  $\tau$ 
end function
```

mRMR

- Combining two constraints:

-- max-relevance:

$$\max RL(S, c),$$

$$RL = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c)$$

-- min-redundancy:

$$\min RD(S),$$

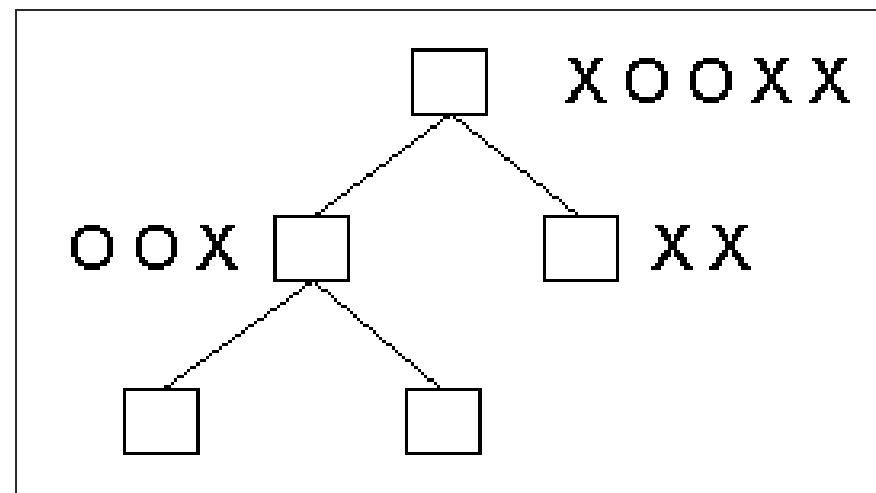
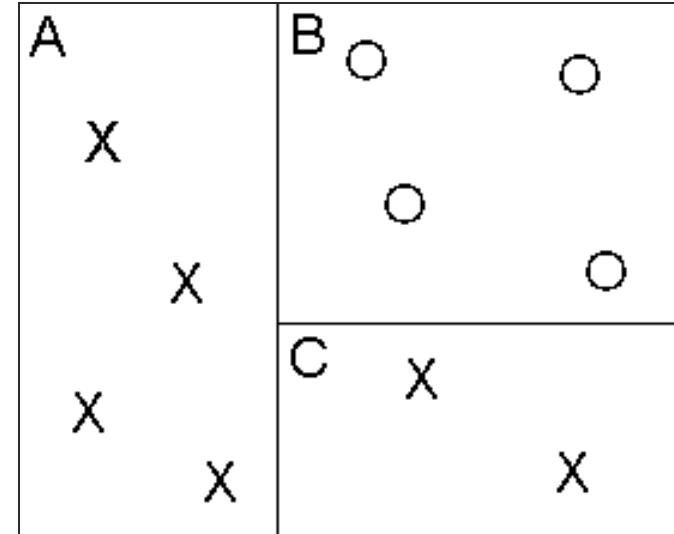
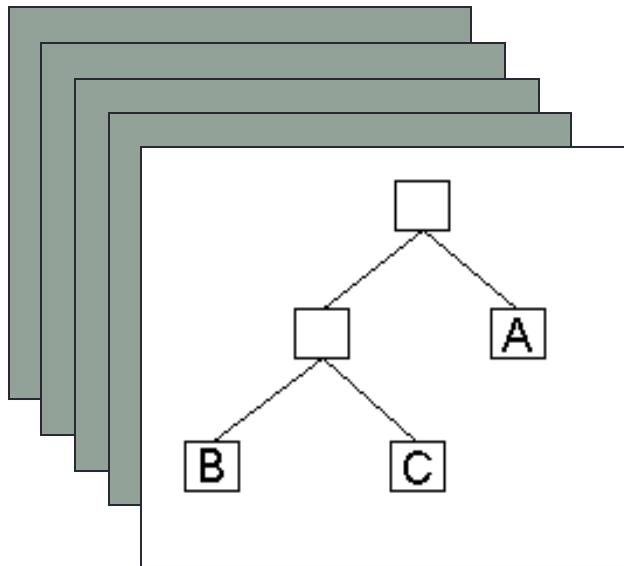
$$RD = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i; x_j)$$

$$\max \Phi(RL, RD), \quad \Phi = RL - RD$$



$$\max_{x_j \in X - S_{m-1}} [I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j; x_i)]$$

Selección de Características mediante Random Forest



Selección de Características

Mediante regularización:

- La regularización consiste en añadir una penalización a los diferentes parámetros del modelo de aprendizaje de la máquina para reducir la libertad del modelo.
- De esta manera, se utiliza para evitar el sobre-ajuste.
- En la regularización de un modelo lineal, la penalización se aplica sobre los coeficientes que multiplican cada uno de los predictores.
- De los diferentes tipos de regularización, Lasso o L1 tiene la propiedad que es capaz de reducir algunos de los coeficientes a cero.
- Por lo tanto, esa característica puede ser eliminada del modelo.

LASSO Least Absolute Shrinkage and Selection Operator

- Outcome variable y_i , for cases $i = 1, 2, \dots, n$, features x_{ij} , $j = 1, 2, \dots, p$
- Minimize

$$\sum_{i=1}^n (y_i - \sum_j x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Equivalent to minimizing sum of squares with constraint $\sum |\beta_j| \leq s$.
- Similar to **ridge regression**, which has constraint $\sum_j \beta_j^2 \leq t$
- Lasso does variable selection and shrinkage; ridge only shrinks.

Selección de Características

Extracción de características

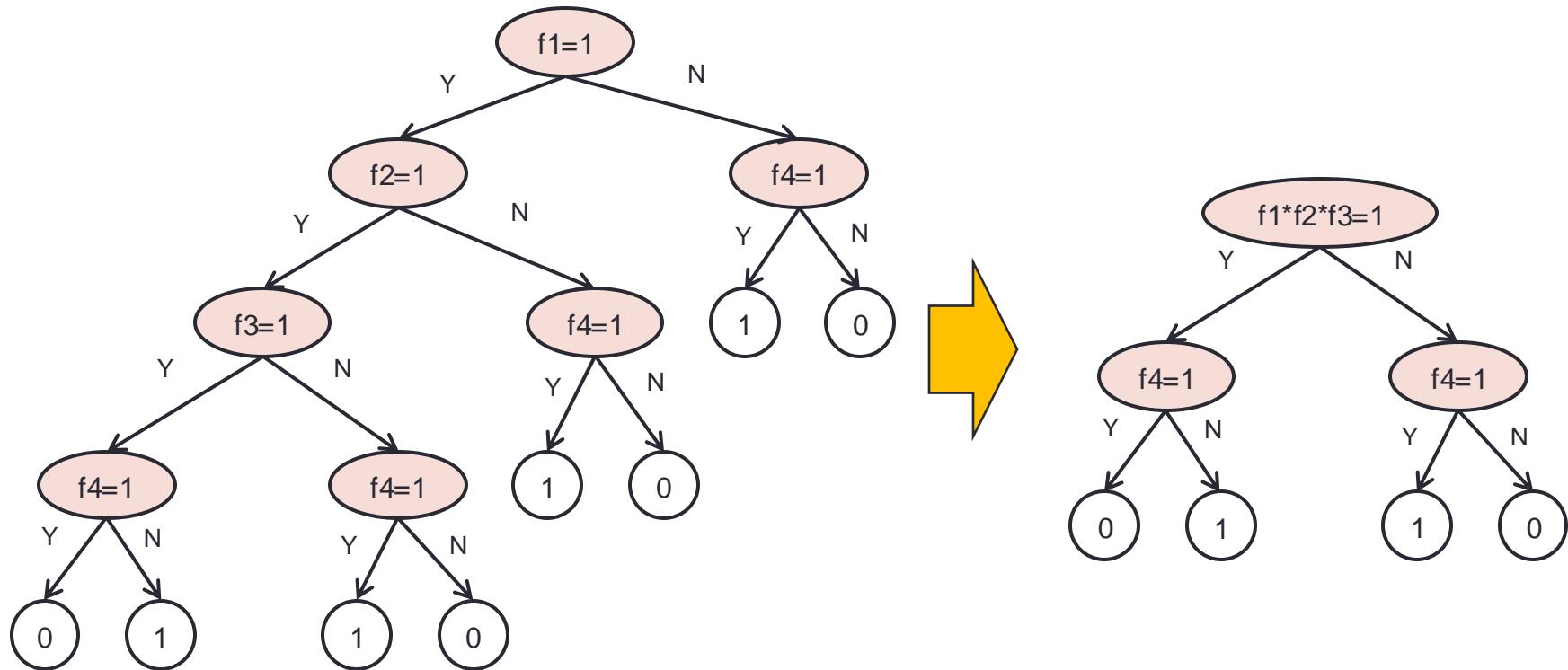


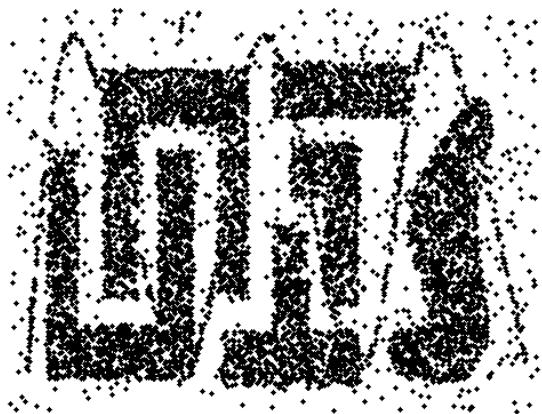
Fig. 7.4 The effect of using the product of features in decision tree modeling

Selección de Instancias

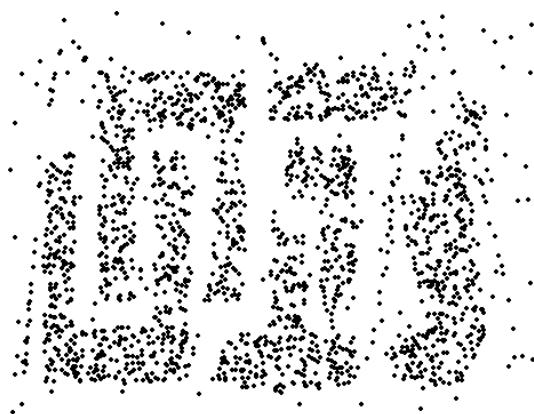
- ✿ La SI pretende elegir los ejemplos que sean relevantes para una aplicación y lograr el máximo rendimiento. El resultado de la SC sería:
 - ❖ Menos datos → los algoritmos pueden aprender más rápidamente
 - ❖ Mayor exactitud → el clasificador generaliza mejor
 - ❖ Resultados más simples → más fácil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

Selección de Instancias

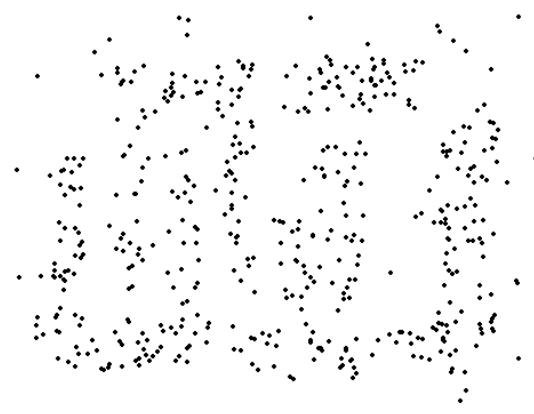
Ejemplos de diferentes tamaños



8000 puntos

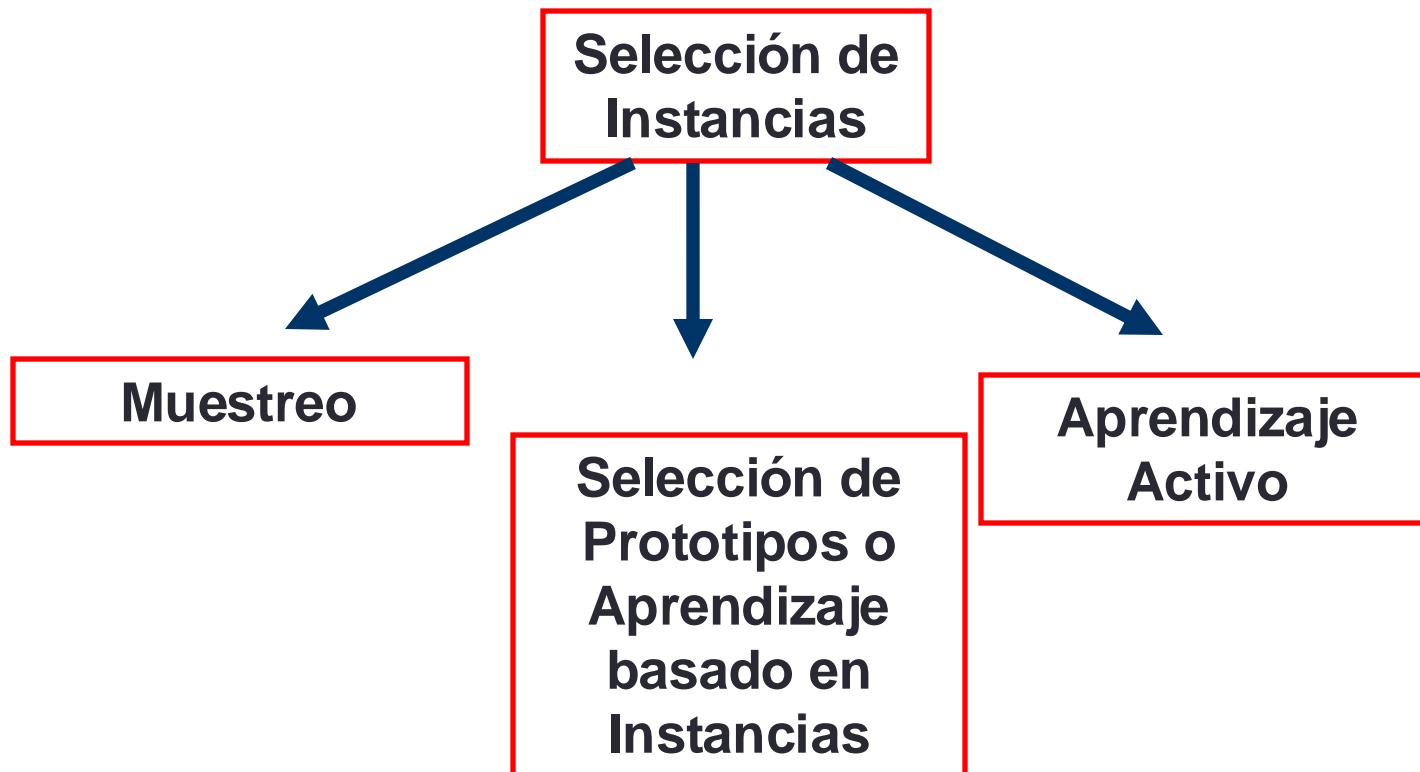


2000 puntos

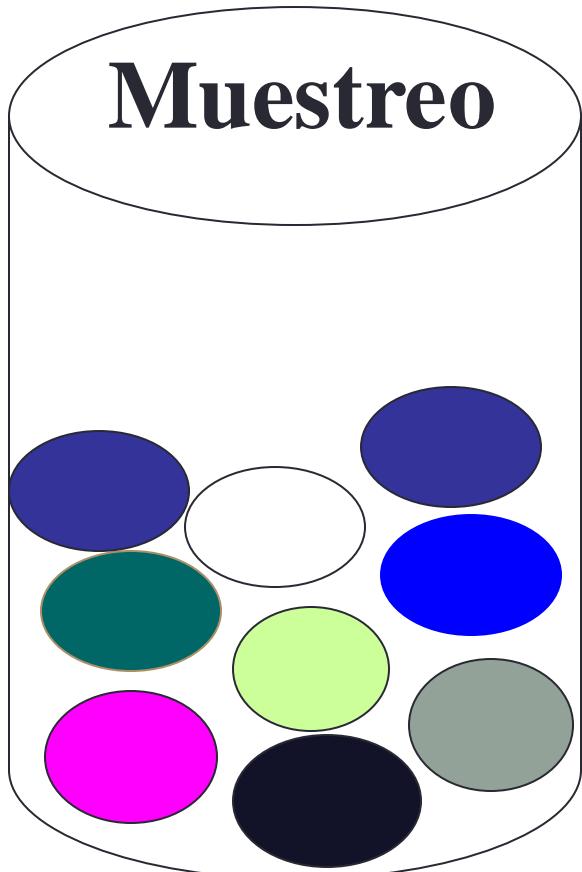


500 puntos

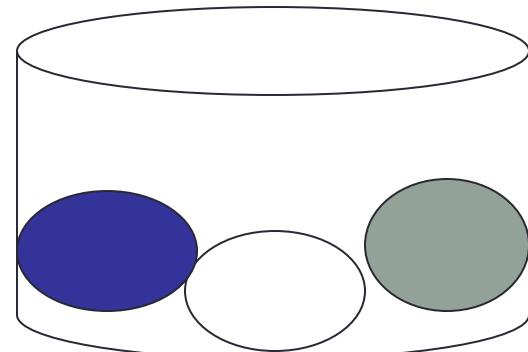
Selección de Instancias



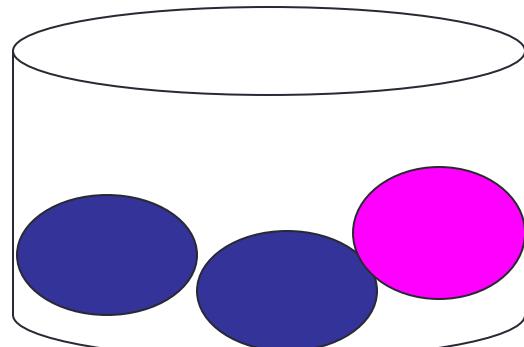
Selección de Instancias



SRSWOR
(muestreo
Simple sin
Reemplazamiento)



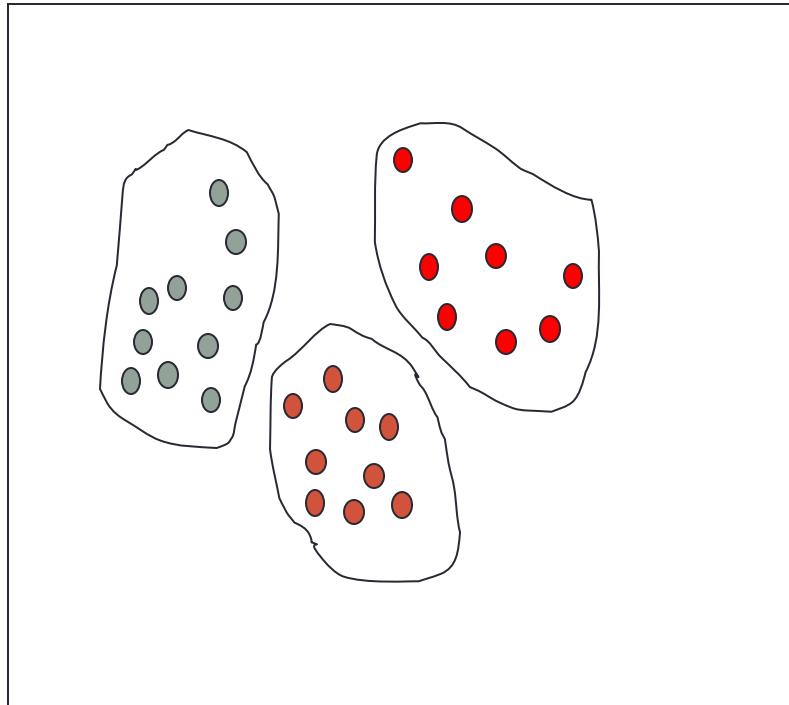
SRSWR



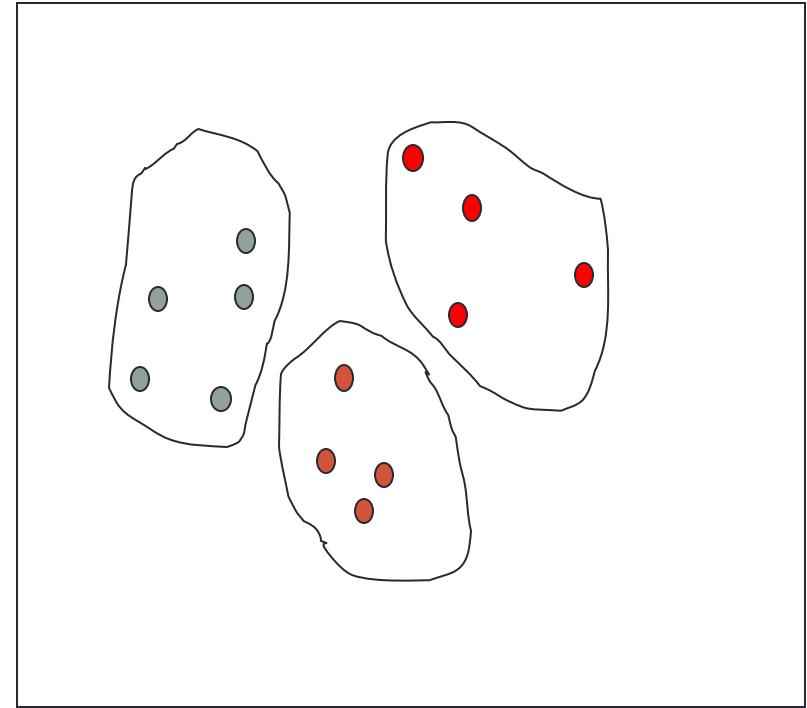
Selección de Instancias

Muestreo

Datos sin refinar



Reducción simple



Selección de Instancias

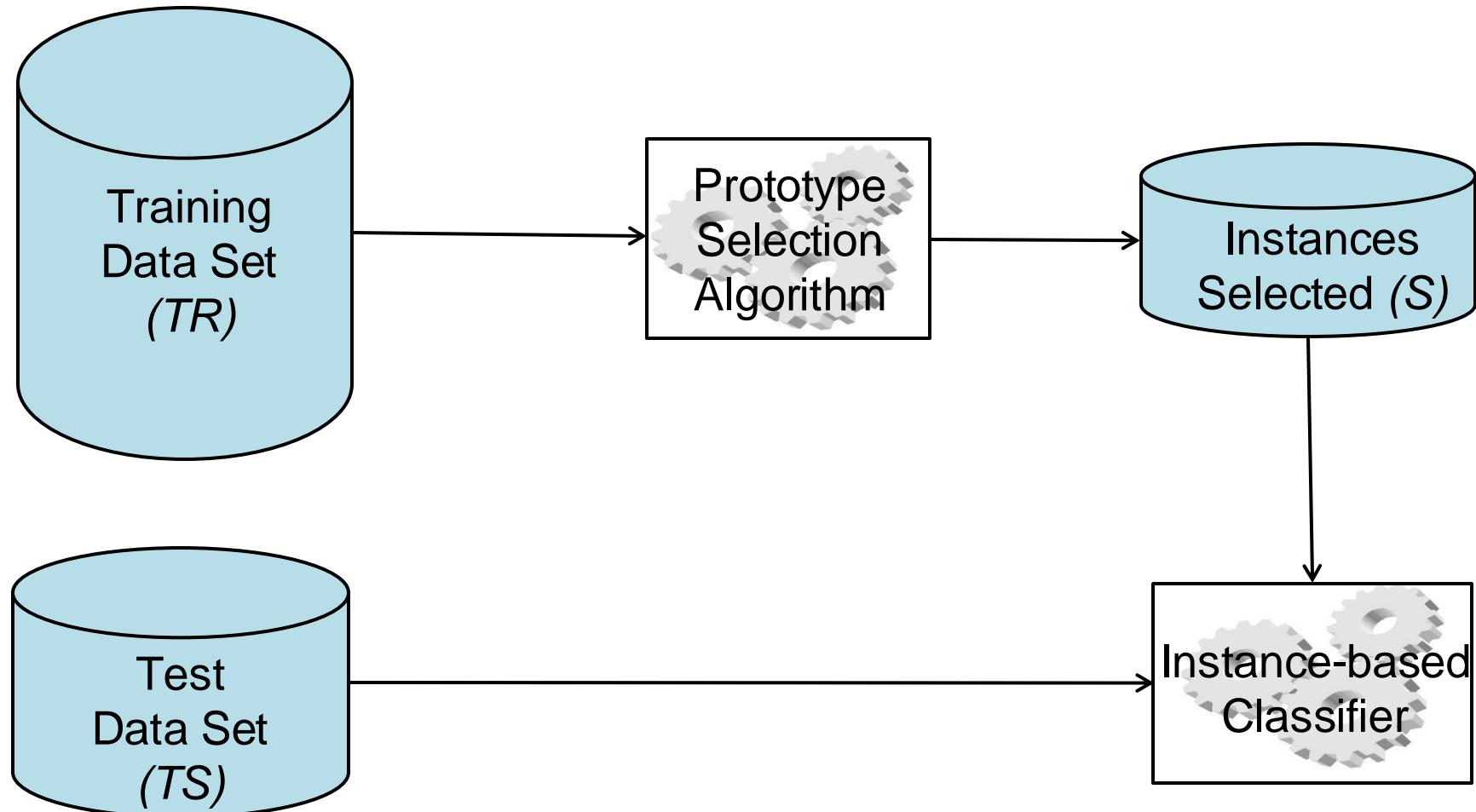
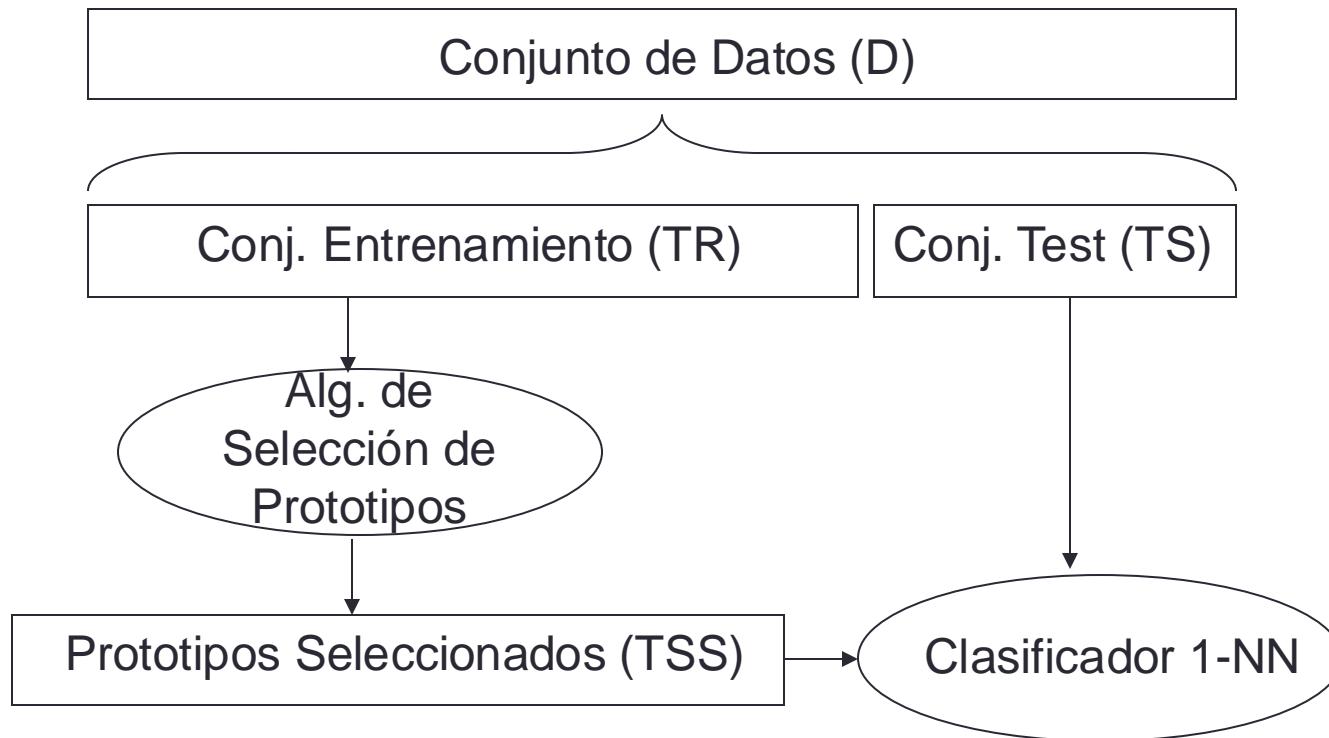


Fig. 8.1 PS process

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

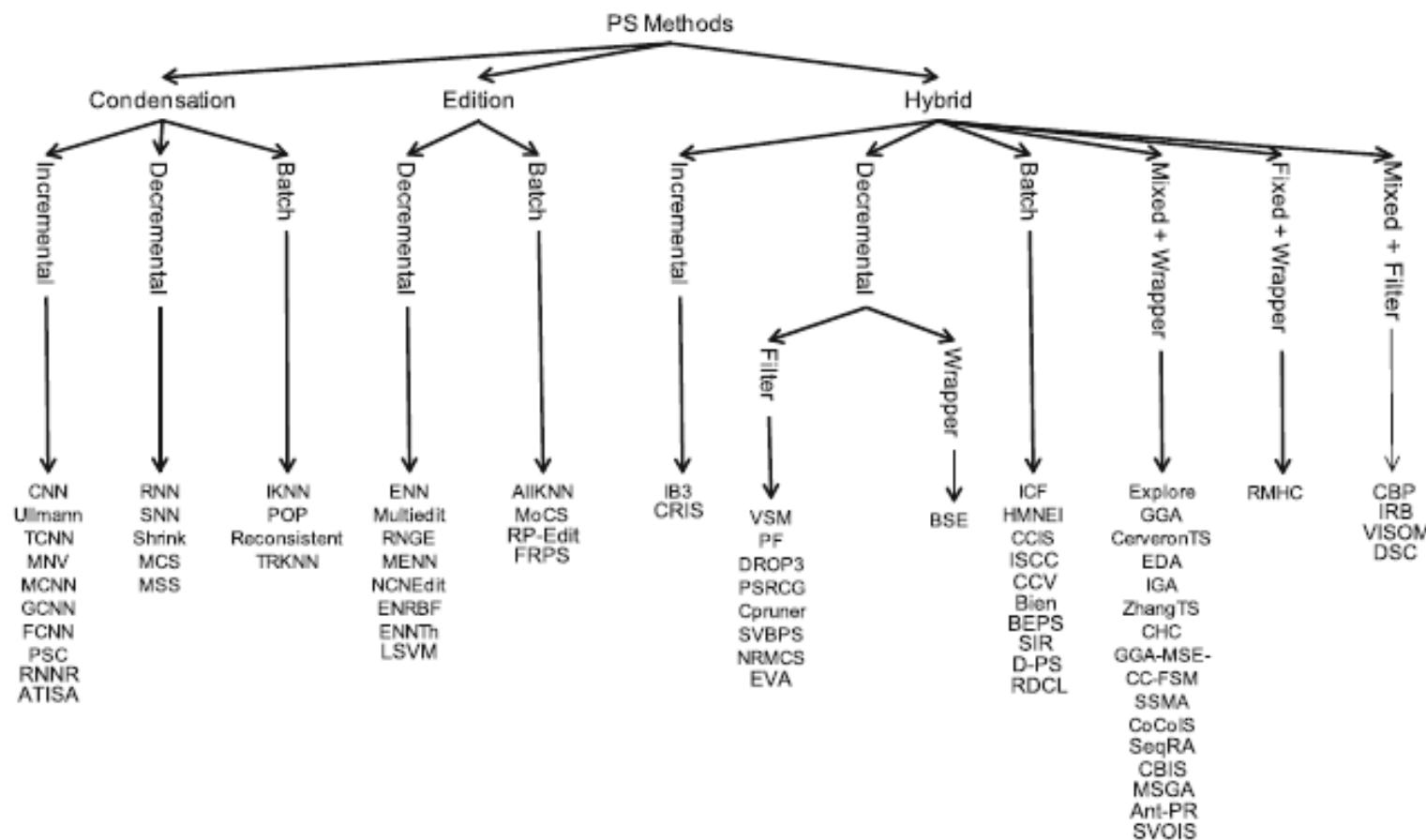


Fig. 8.3 PS taxonomy

Selección de Instancias

Selección de Prototipos o Aprendizaje basado en Instancias

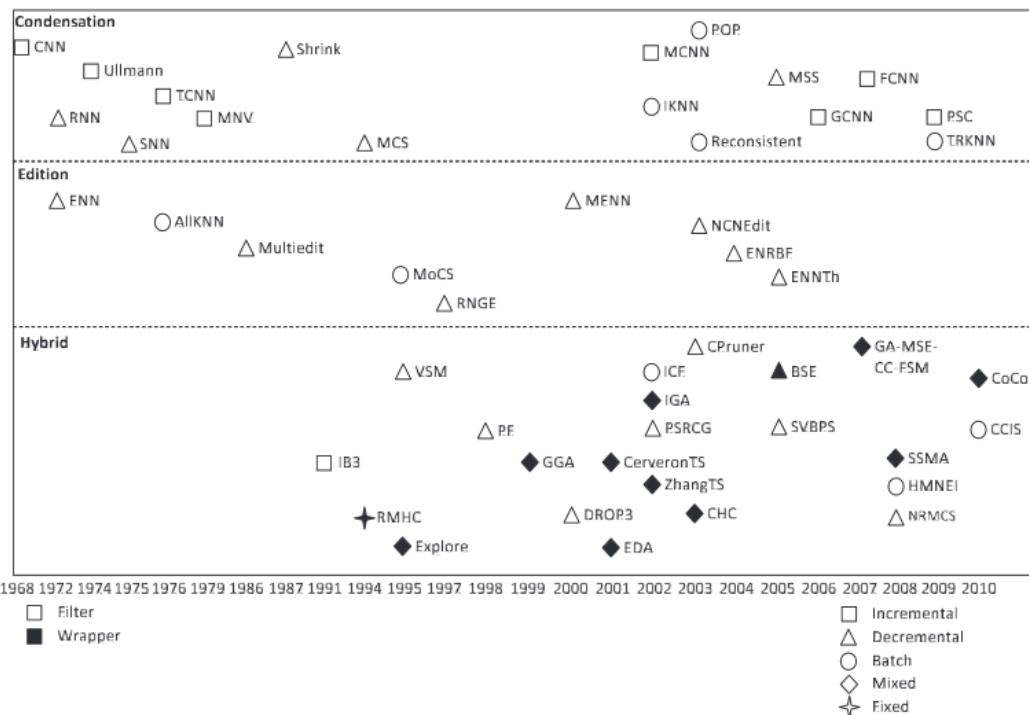


Fig. 2. Prototype selection map.

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

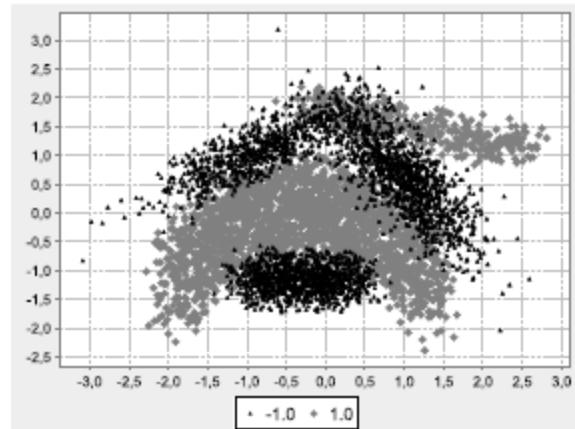
Selección de Instancias

Un par de algoritmos clásicos:

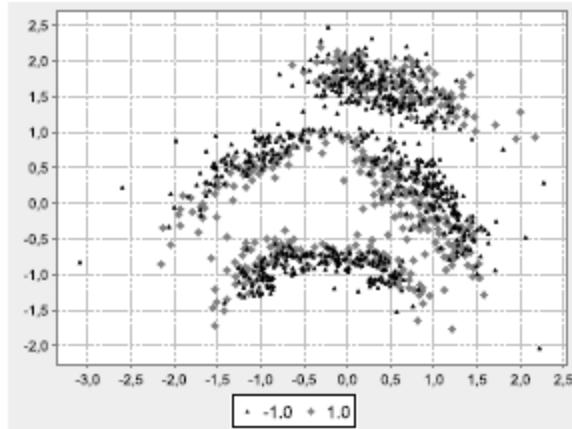
- Algoritmo clásico de Condensación: Condensed Nearest Neighbor (CNN)
 - Incremental
 - Inserta solo las instancias mal clasificadas a partir de una selección aleatoria de una instancia de cada clase.
 - Dependiente del orden de presentación
 - Solo retiene puntos pertenecientes al borde
- Algoritmo clásico de Edición: Edited Nearest Neighbor (ENN)
 - Por lotes
 - Borras aquellas instancias que se clasifican incorrectamente usando sus k vecinos más cercanos ($K = 3, 5 \text{ ó } 9$).
 - “Suaviza” las fronteras, pero retiene el resto de puntos (muchos redundantes)

Selección de Instancias

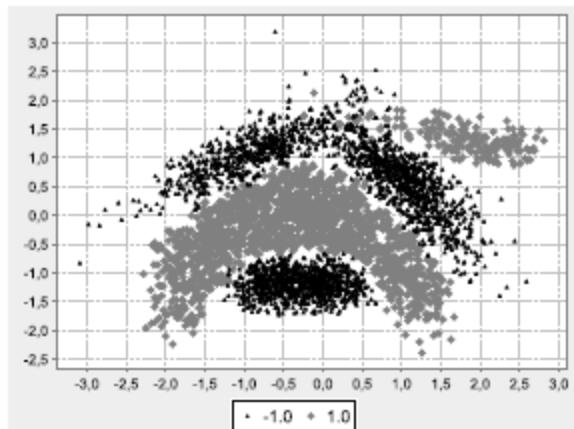
Ejemplos gráficos:



(a) Banana
(0.8751, 0.7476)



(b) CNN (0.7729, 0.8664, 0.7304)

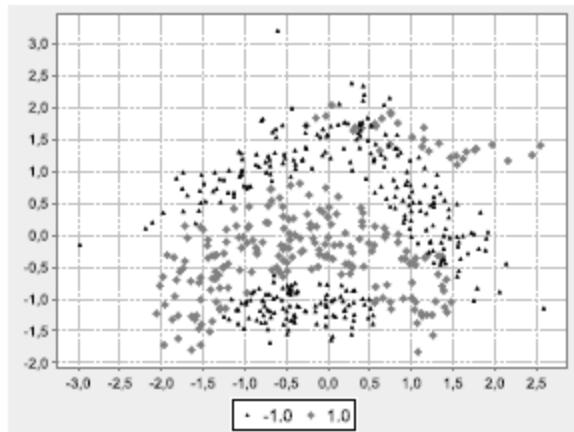


(h) AllKNN (0.1758, 0.8934, 0.7831)

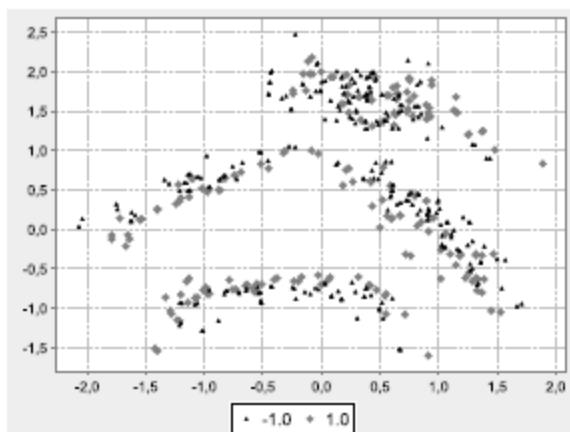
Conjunto banana con 5.300 instancias y dos clases. Conjunto obtenido por CNN y AllKNN (aplicación iterativa de ENN con $k=3, 5$ y 7).

Selección de Instancias

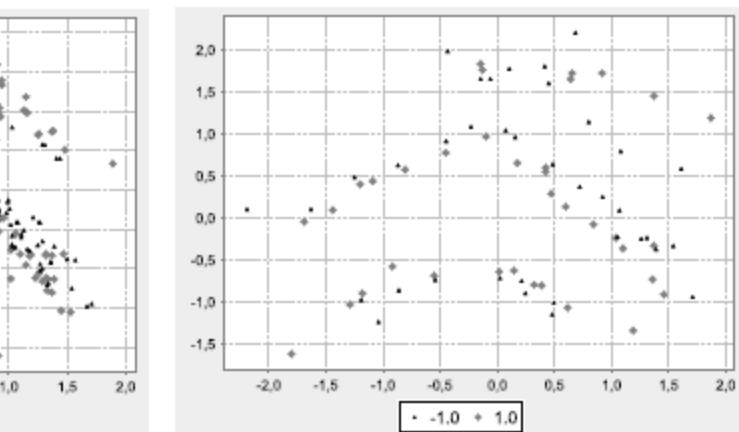
Ejemplos gráficos:



(k) RMHC (0.9000, 0.8972, 0.7915)



(e)
(0.9151, 0.8696, 0.7356)



DROP3 (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

Selección de Instancias. Eficiencia

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

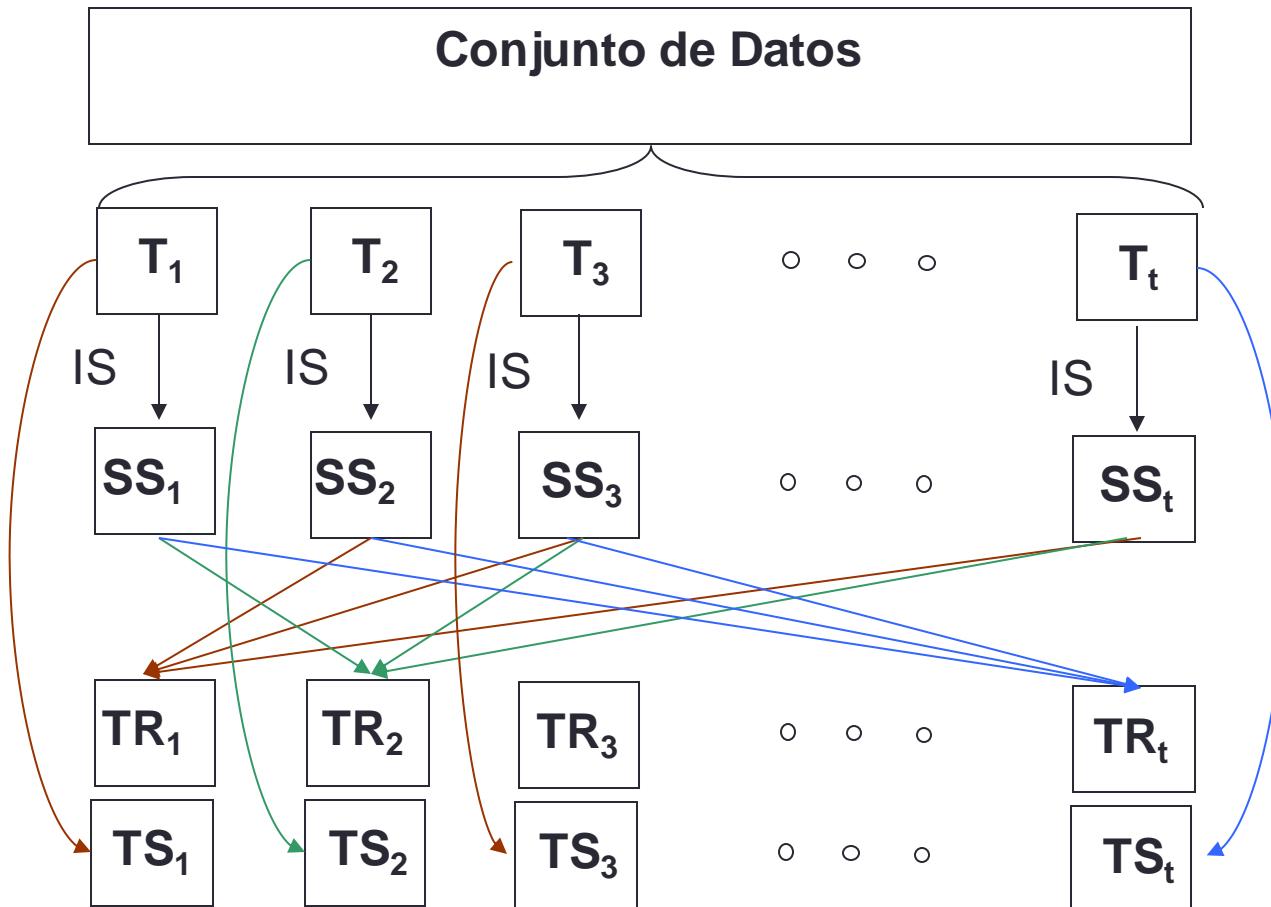
Las principales dificultades que deben afrontar los algoritmos de Selección de Prototipos son: Eficiencia, recursos, generalización, representación.

¿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.



Referencia: J.R. Cano, F. Herrera, M. Lozano. *Stratification for Scaling Up Evolutionary Prototype Selection*. Pattern Recognition Letters 26:7 (2005) 953-963.

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

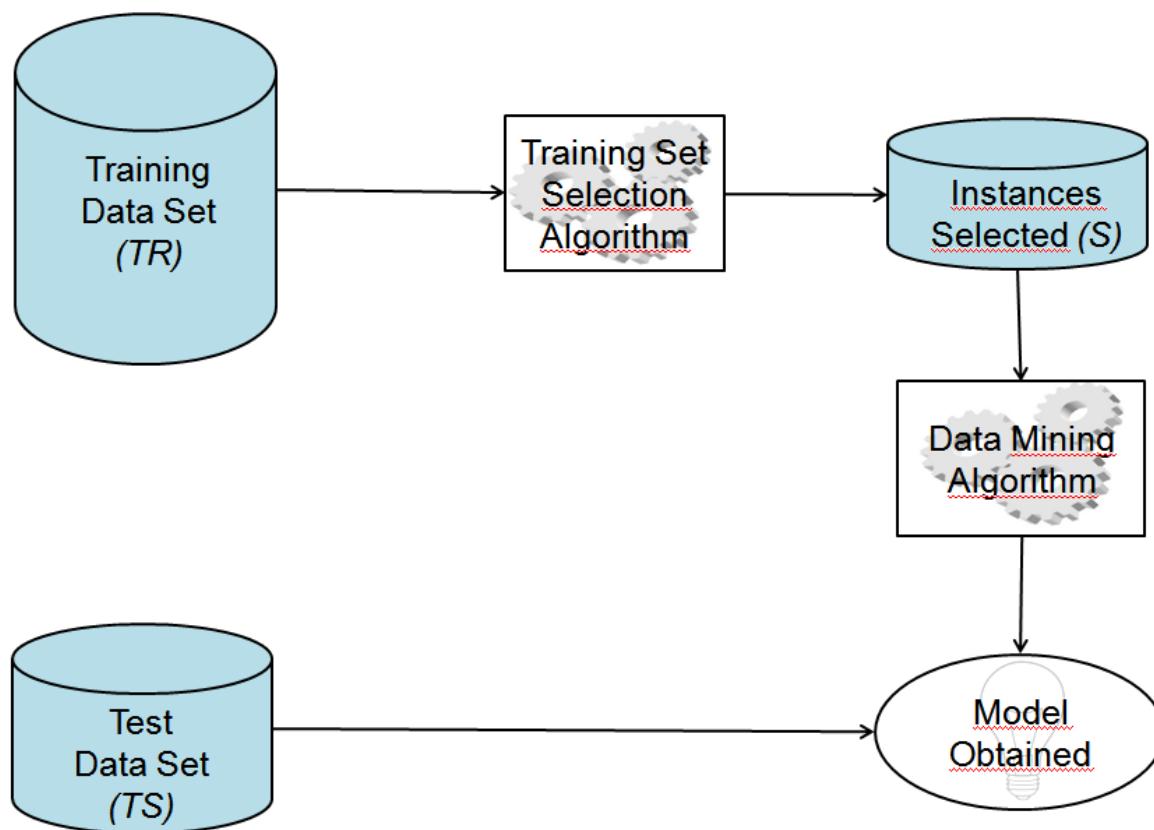
Selección de Instancias. Ejemplo – Kdd Cup'99

	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.

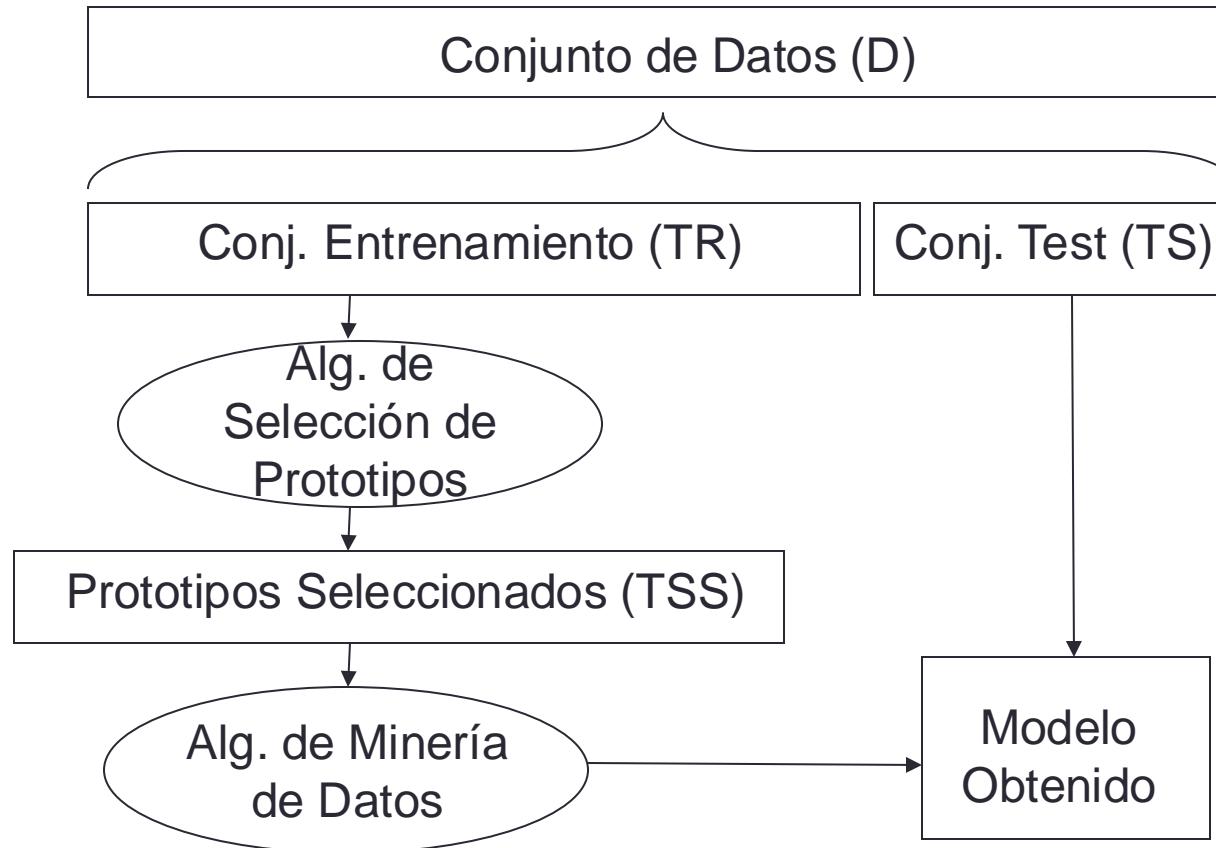
Selección de Instancias

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



Selección de Instancias

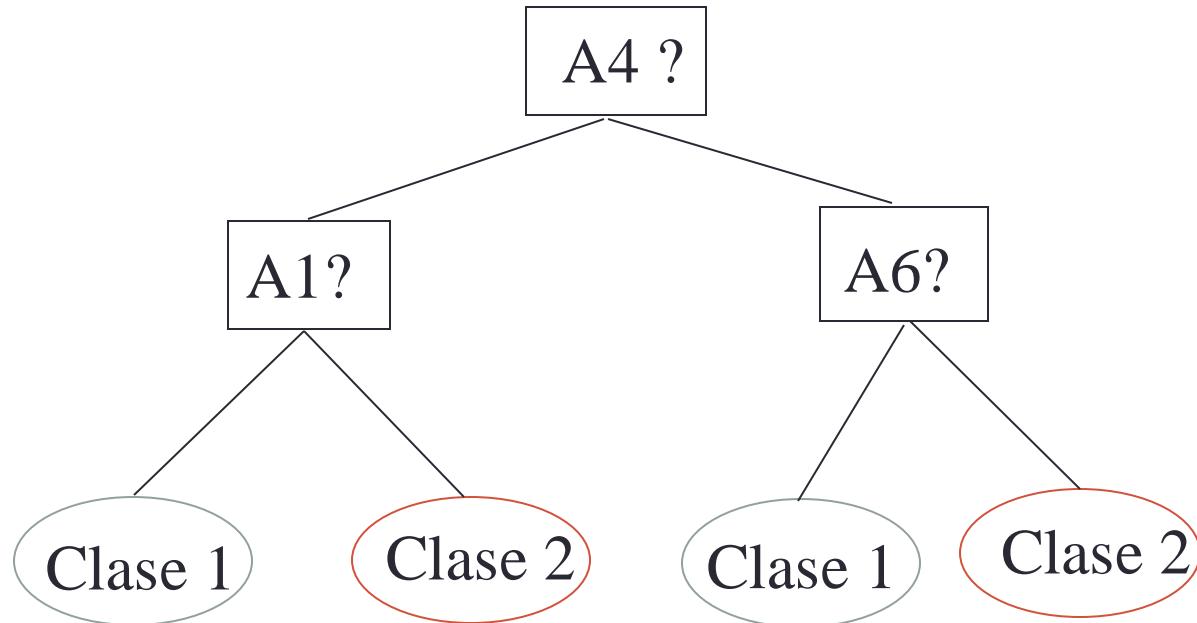
Selección de Prototipos la **Selección de Conjuntos de Entrenamiento**



Ej. Selección de Instancias y Extracción de Árboles de Decisión

Selección de Instancias

Conjunto inicial de atributos:
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$



-----> Conjunto reducido de atributos: $\{A_1, A_4, A_6\}$

Los árboles de decisión seleccionan características

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
C4.5	252		99.97%	99.94%
Cnn Strat	83	81.61%	98.48%	96.43%
Drop1 Strat	3	99.97%	38.63%	34.97%
Drop2 Strat	82	76.66%	81.40%	76.58%
Drop3 Strat	49	56.74%	77.02%	75.38%
Ib2 Strat	48	82.01%	95.81%	95.05%
Ib3 Strat	74	78.92%	99.13%	96.77%
Icf Strat	68	23.62%	99.98%	99.53%
CHC Strat	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.](#)

Selección de Instancias

Conjuntos de datos no balanceados

- Algunos problemas tienen una presencia de las clases desigual
 - Diagnosis médica: 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 97% de clasificación correcta, pero no son útiles

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]
FS and feature weighting	[128]
FS 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 multiclasicadores

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]
Boosting 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

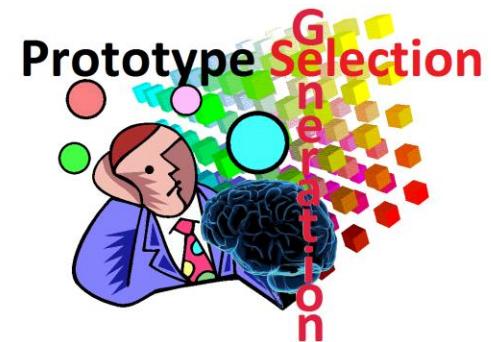
Table 8.5 Scaling-up and distributed approaches

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, 15, 2015



Códigos (Java):

KEEL

Selección de Instancias (website)

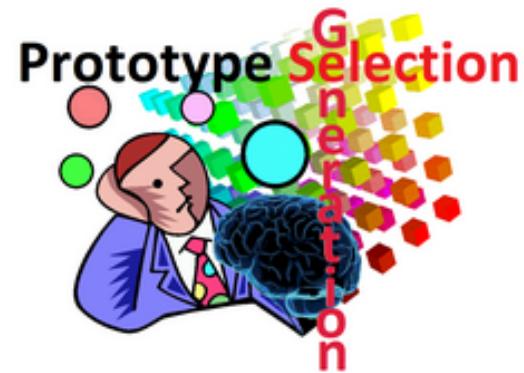
<http://sci2s.ugr.es/pr/>

[Home](#) » [Thematic Sites](#) » Prototype Reduction in Nearest Neighbor Classification: Prototype Selection and Prototype Generation

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](https://doi.org/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](https://doi.org/10.1109/TSMCC.2010.2103939)  COMPLEMENTARY MATERIAL
to the paper [here](#): datasets, experimental results and source codes.



The web is organized according to the following Summary:

The web is organized according to the following Summary:

1. Introduction to Prototype Reduction
2. Prototype Selection
 - A. Background
 - B. Taxonomy
 - C. Prototype Selection Methods
 - D. Experimental Analyses
 - E. SCI²S Approaches on Prototype Selection
3. Prototype Generation
 - A. Background
 - B. Taxonomy
 - C. Prototype Generation Methods
 - D. Experimental Analyses
 - E. SCI²S Approaches on Prototype Generation
4. Prototype Reduction Outlook
 - A. Key Milestones & Surveys
 - B. Hybrid approaches for Prototype and Feature Reduction
 - C. Evolutionary Proposals
 - D. SCI²S Related Approaches
 - E. Prototype Reduction Visibility at the Web of Science
 - F. Software and Algorithm Implementations
 - G. Recent and forthcoming approaches

Selección de Instancias (website)

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

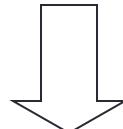
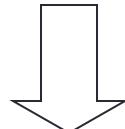
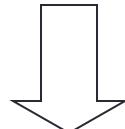
Discretización

- 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.

EDAD	5	6	6	9	...	15	16	16	17	20	...	24	25	41	50	65	...	67
COCHE EN PROPIEDAD	0	0	0	0	...	0	1	0	1	1	...	0	1	1	1	1	...	1

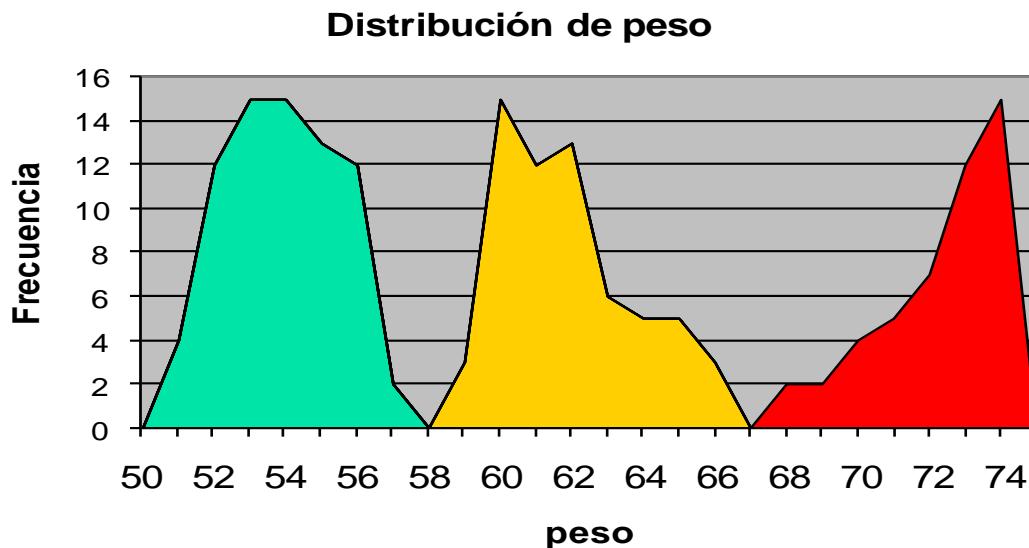


EDAD [5,15]

EDAD [16,24]

EDAD [25,67]

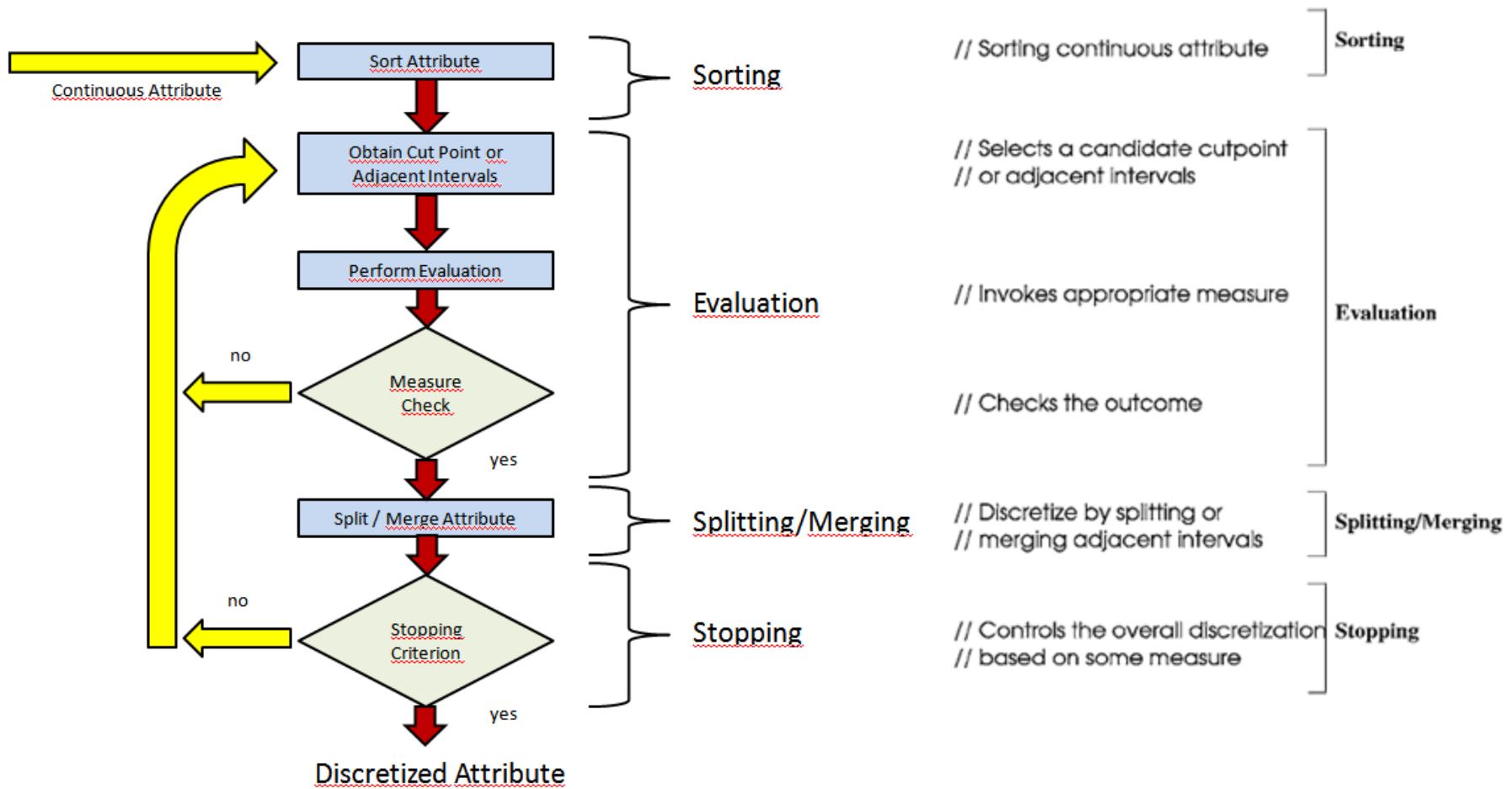
Discretización



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.

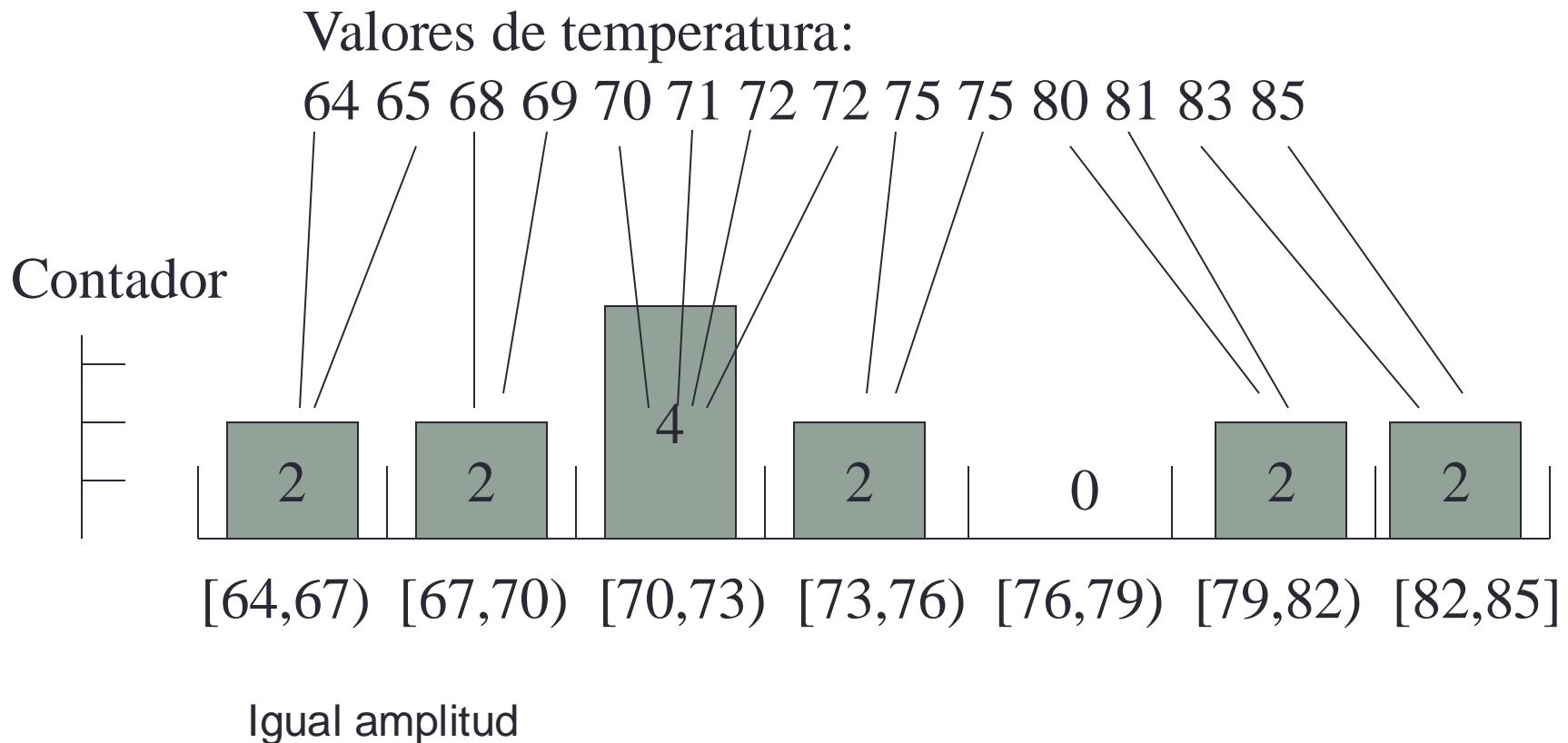
Discretización

- **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



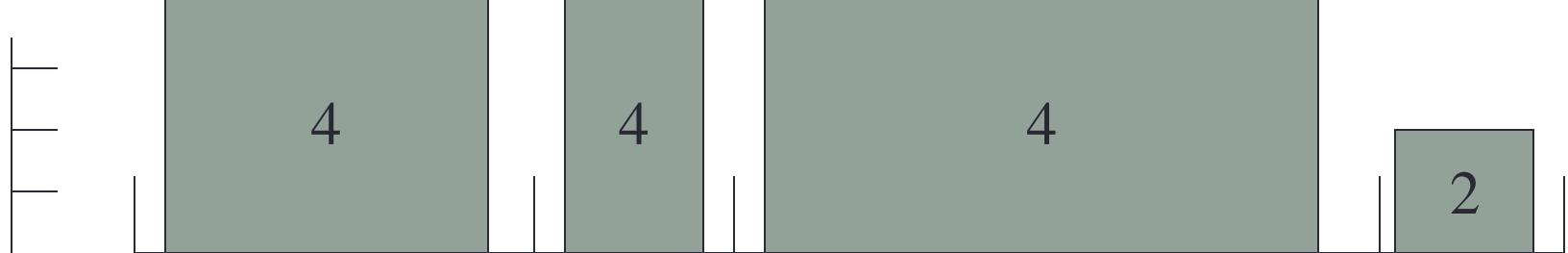
Discretización

Ejemplo Discretización: Igual frecuencia

Valores de la temperatura

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

Contador



[64 69] [70 .. 72] [73 81] [83 .. 85]

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)

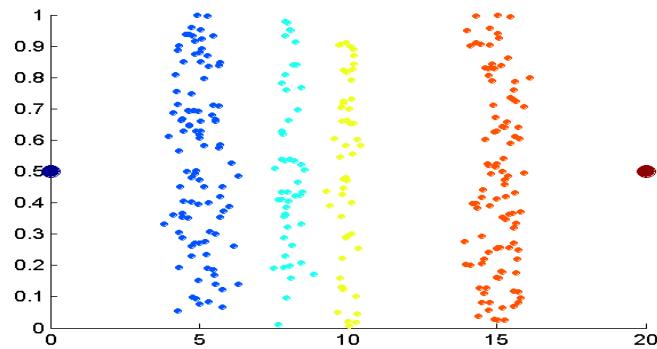
- Comienzan con los puntos de corte dados entre ejemplos de **diferentes** clases:

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

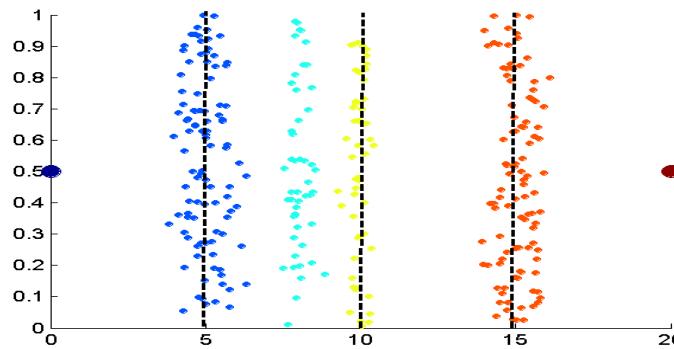
- Minimum Description Length Principle (MDLP), basado en entropía, se utiliza para escoger los puntos de corte útiles entre los anteriores.
- El criterio de parada se basa también en MDLP.
- MDLP se formula como el problema de encontrar el coste de comunicación entre un emisor y un receptor. Se asume 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

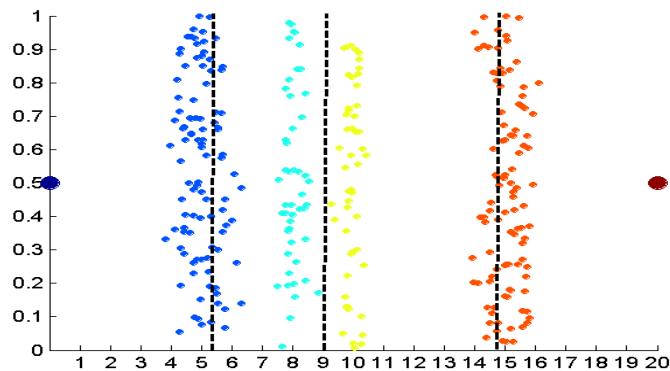
Discretización sin utilizar las clases



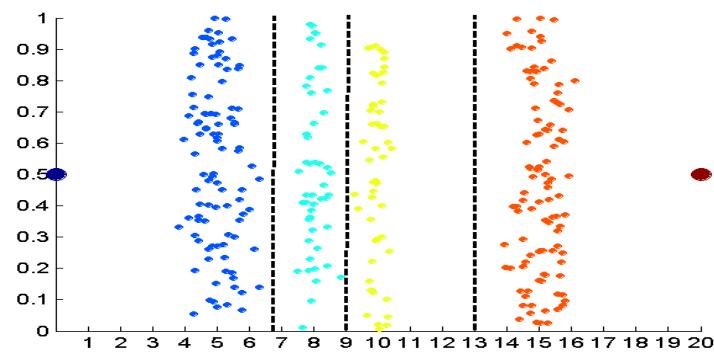
Datos



Igual anchura de intervalo



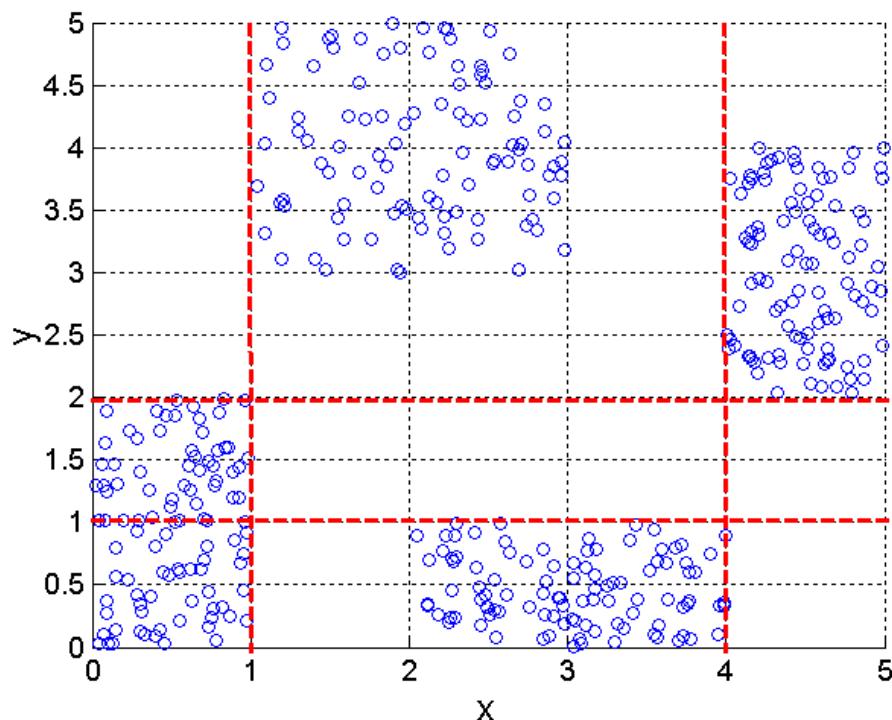
Igual frecuencia



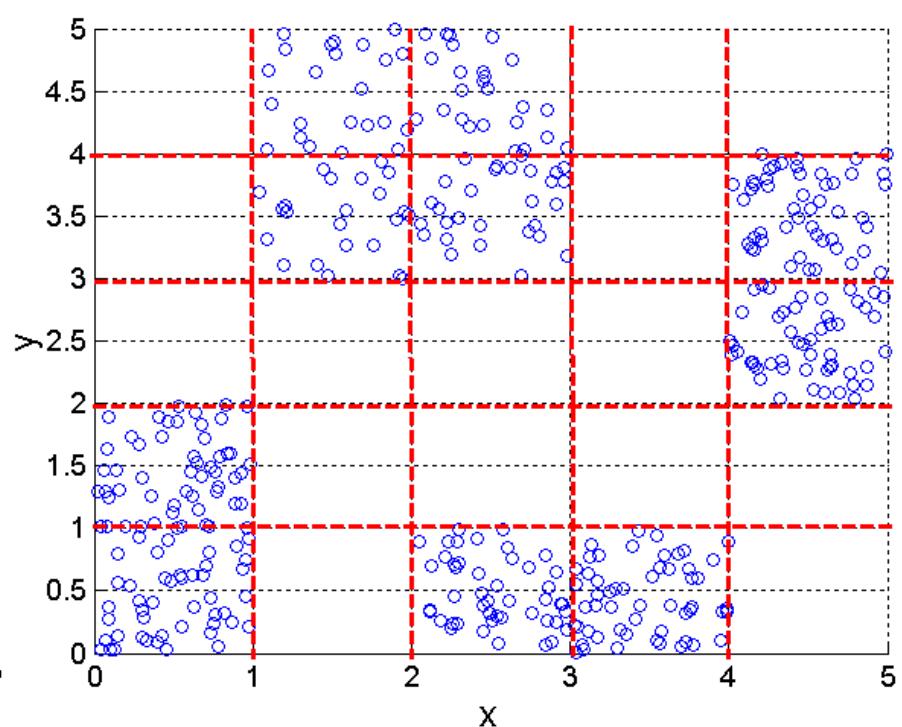
K-medias

Discretización

Discretización utilizando clases (basado en entropía)



3 categorías para ambas variables x e y



5 Categorías para ambas variables

CAIM Algorithm

CAIM discretization criterion

$$CAIM(C, D | F) = \frac{\sum_{r=1}^n \frac{\max_r^2}{M_{+r}}}{n}$$

where:

n is the number of intervals

r iterates through all intervals, i.e. $r = 1, 2, \dots, n$

\max_r is the maximum value among all q_{ir} values (maximum in the r th column of the quanta matrix), $i = 1, 2, \dots, S$,

M_{+r} is the total number of continuous values of attribute F that are within the interval $(d_{r-1}, d_r]$

Quanta matrix:

Class	Interval					Class Total
	$[d_0, d_1]$	\dots	$(d_{r-1}, d_r]$	\dots	$(d_{n-1}, d_n]$	
C_1	q_{11}	\dots	q_{1r}	\dots	q_{1n}	M_{1+}
\vdots	\vdots	\dots	\vdots	\dots	\vdots	\vdots
C_i	q_{i1}	\dots	q_{ir}	\dots	q_{in}	M_{i+}
\vdots	\vdots	\dots	\vdots	\dots	\vdots	\vdots
C_S	q_{S1}	\dots	q_{Sr}	\dots	q_{Sn}	M_{S+}
Interval Total	M_{+1}	\dots	M_{+r}	\dots	M_{+n}	M

CAIM Algorithm

CAIM discretization criterion

$$CAIM(C, D | F) = \frac{\sum_{r=1}^n \frac{\max_r^2}{M_{+r}}}{n}$$

- The larger the value of the CAIM ([0, M], where M is # of values of attribute F, the higher the interdependence between the class labels and the intervals)
- The algorithm favors discretization schemes where each interval contains majority of its values grouped within a single class label (the \max_r values)
- The squared \max_r value is scaled by the M_{+r} to eliminate negative influence of the values belonging to other classes, on the class with the maximum number of values, on the entire discretization scheme
- The sum is divided by the number of intervals (n) to favor discretization schemes with small number of intervals

CAIM Algorithm

Given: M examples described by continuous attributes F_i , S classes

For every F_i do:

Step1

- 1.1 find maximum (d_n) and minimum (d_o) values
- 1.2 sort all distinct values of F_i in ascending order and initialize all possible interval boundaries, B, with the minimum, maximum, and the midpoints, for all adjacent pairs
- 1.3 set the initial discretization scheme to $D: \{[d_o, d_n]\}$, set variable GlobalCAIM=0

Step2

- 2.1 initialize $k=1$
- 2.2 tentatively add an inner boundary, which is not already in D, from set B, and calculate the corresponding CAIM value
- 2.3 after all tentative additions have been tried, accept the one with the highest corresponding value of CAIM
- 2.4 if (CAIM > GlobalCAIM or $k < S$) then update D with the accepted, in step 2.3, boundary and set the GlobalCAIM=CAIM, otherwise terminate
- 2.5 set $k=k+1$ and go to 2.2

Result: Discretization scheme D

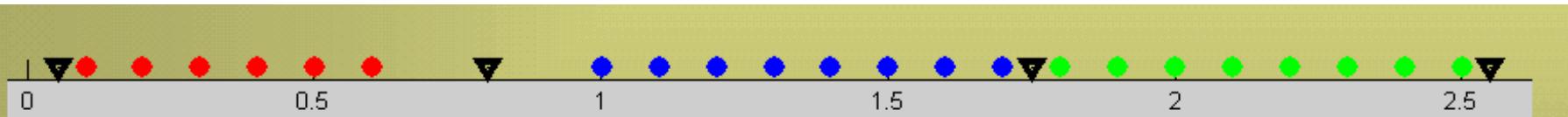
CAIM Algorithm

- Uses **greedy top-down approach** that finds local maximum values of CAIM. Although the algorithm does not guarantee finding the global maximum **of the CAIM criterion** it is effective and computationally efficient: **$O(M \log(M))$**
- Starts with a single interval and divides it iteratively using for the division the boundaries that correspond to the **highest values of CAIM criterion**
- Assumes that every discretized attribute needs at least the number of intervals that is equal to the number of classes (almost always the case)

CAIM Algorithm Example

iteration	max CAIM	# intervals
1	16.7	1
2	37.5	2
3	46.1	3
4	34.7	4

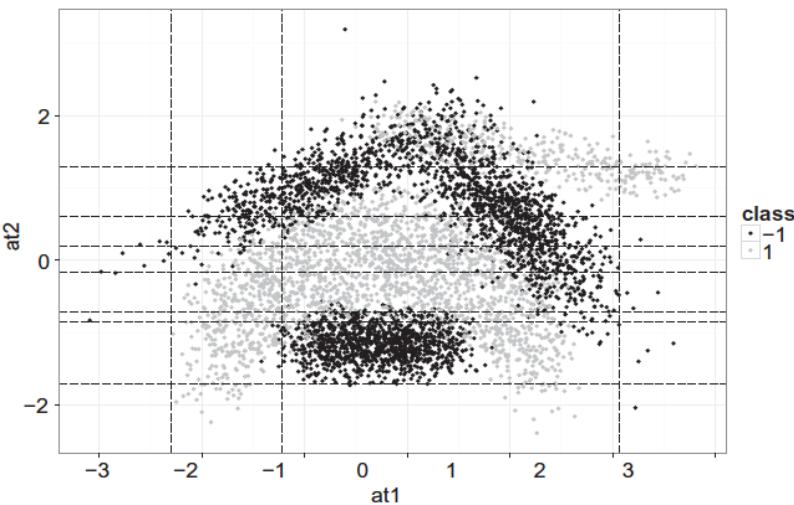
Discretization scheme generated by the **CAIM** algorithm



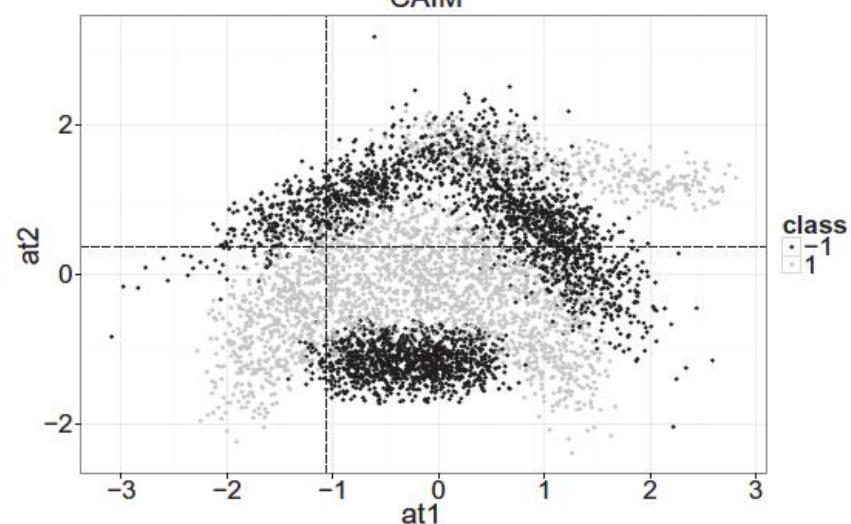
raw data (red = Iris-setosa, blue = Iris-versicolor, black = Iris-virginica)

Discretización

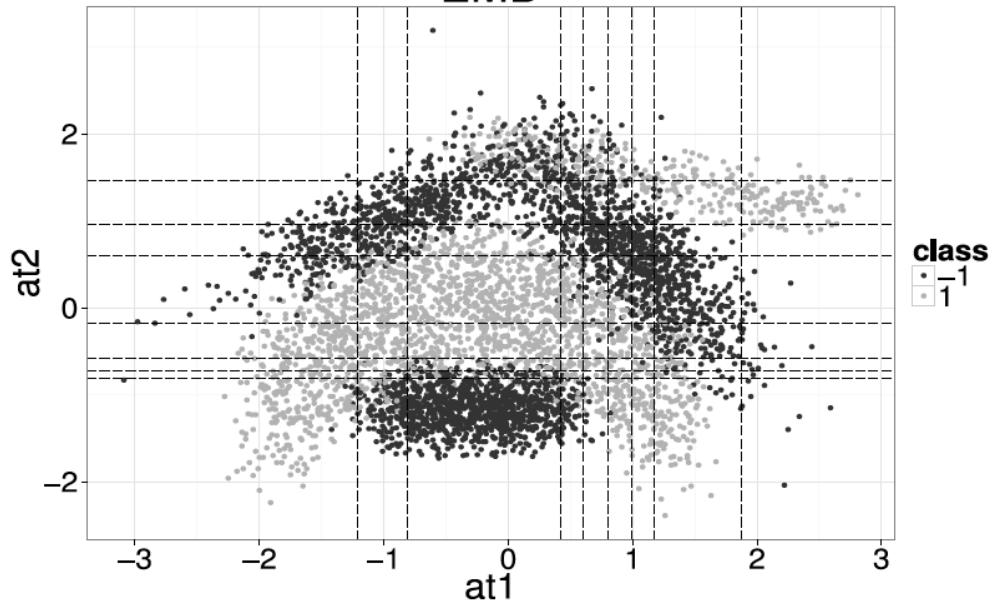
MDLP



CAIM



EMD



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

Software en R:  **discretization**

Preprocesamiento de Datos

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

Comentarios Finales

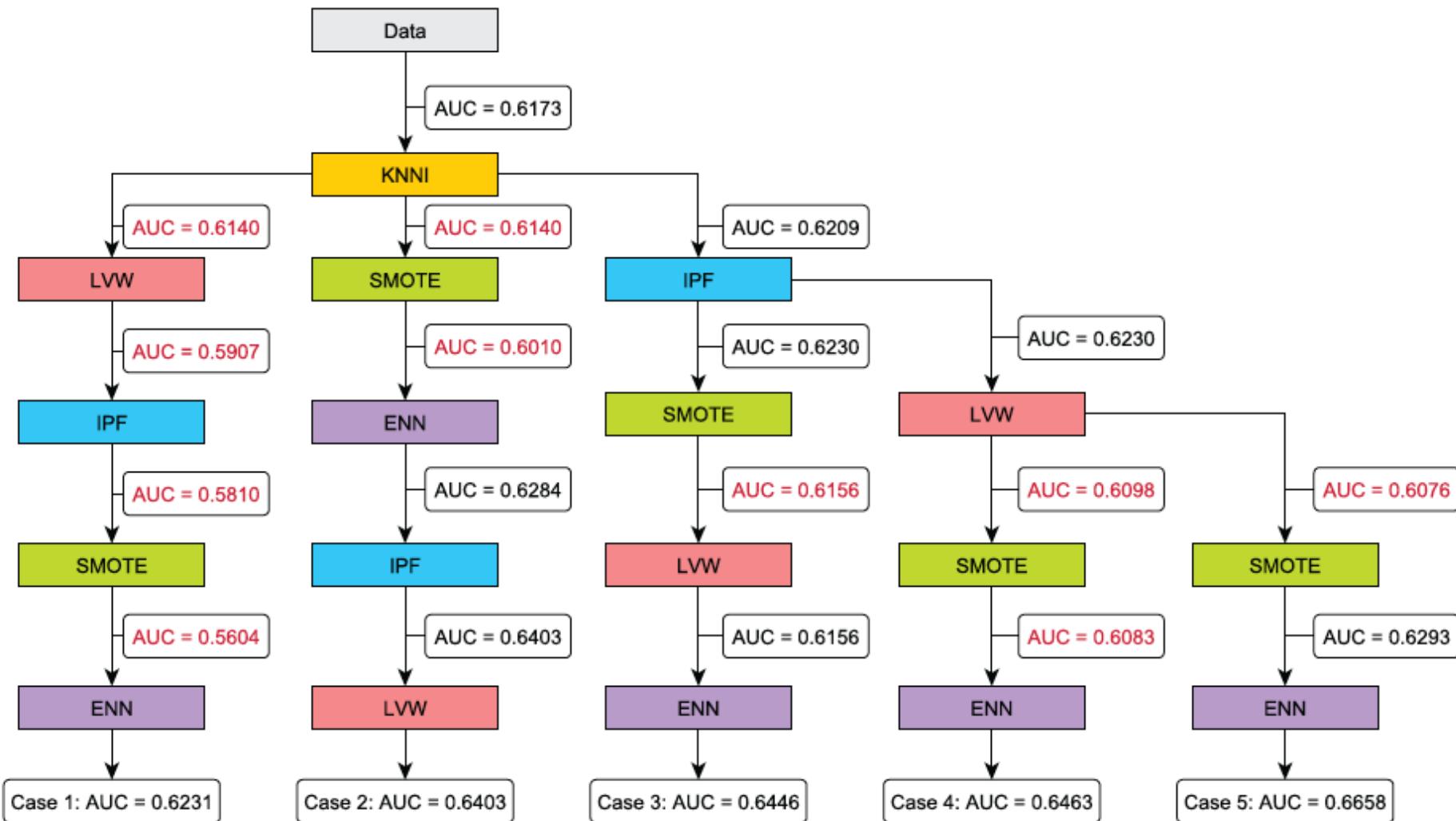
El preprocessamiento de datos es una necesidad cuando se trabaja con una aplicación real, con datos obtenidos directamente del problema.

Comentarios Finales

Una ventaja: El preprocessamiento 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 preprocessamiento 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 preprocessamiento.

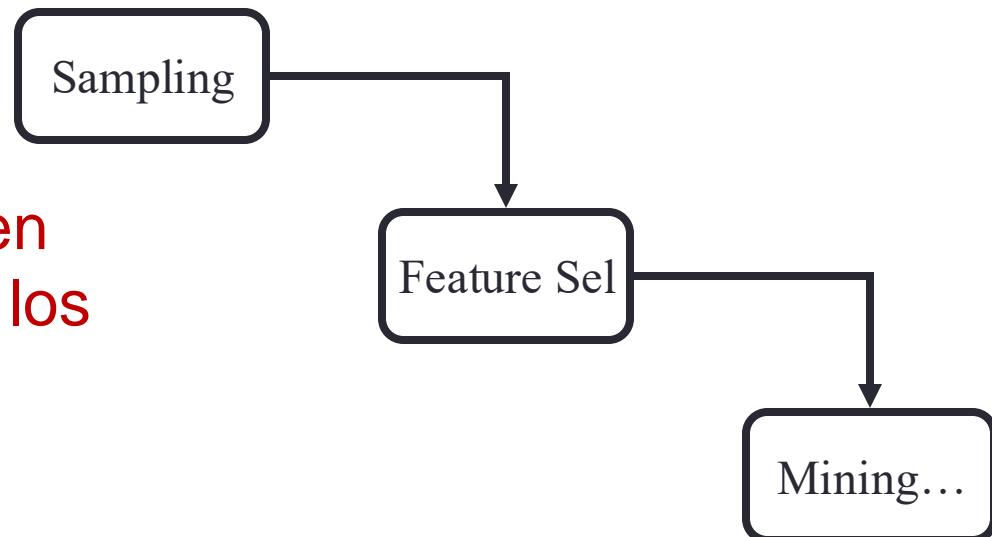
Comentarios Finales



Comentarios Finales

Un inconveniente: El preprocessamiento 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.

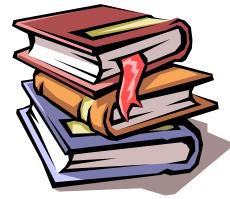
International Journal of Information Technology & Decision Making 5:4 (2006) 597-604

Comentarios Finales

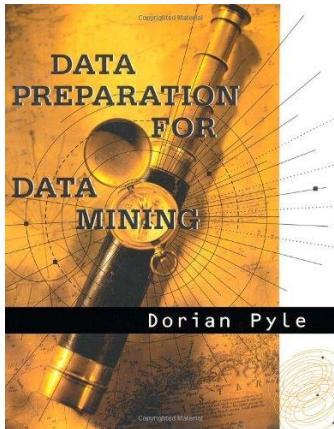
El software de minería de datos KEEL (knowledge extraction based on evolutionary learning) incluye un módulo de preparación de datos de datos (selección de características, imputación de valores perdidos, selección de instancias, discretización, ...)

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

Bibliografía



Bibliografía –Preprocesamiento



**Dorian Pyle
Morgan Kaufmann, Mar 15, 1999**

**S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 15, 2015**

