Minería de Datos

Preprocesamiento de Datos

Dr. Edgar Acuña
Departamento de Ciencias Matemáticas
Universidad de Puerto Rico-Mayaguez

E-mail: edgar@math.uprm.edu, eacunaf@gmail.com

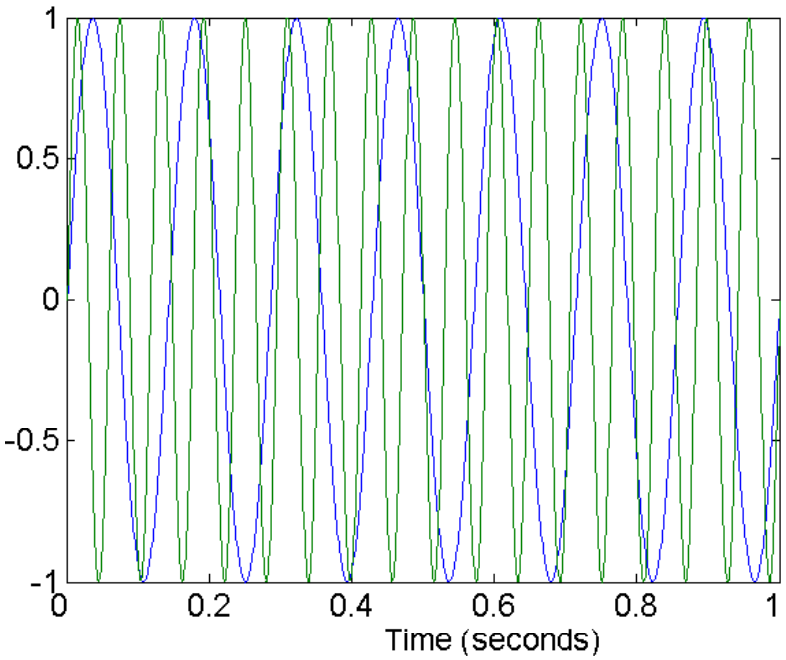
Website: math.uprm.edu/~edgar

¿Por qué preprocesar los datos?

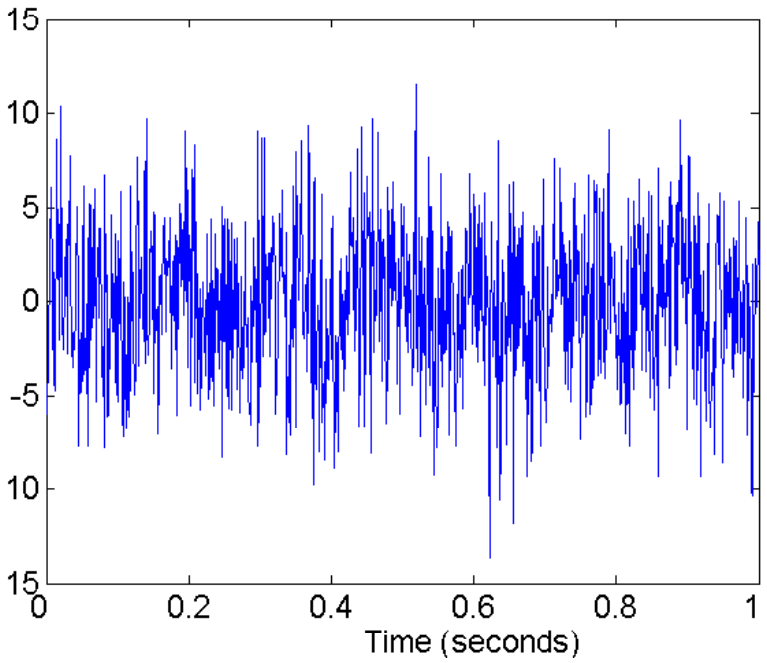
- Los datos en el mundo real son “sucios”:
  - incompletos: falta valores en los atributos, carecen de algunos atributos de interés, o contienen sólo totalizaciones.
  - ruidosos: contienen errores o valores anómalos.
  - inconsistentes: contienen discrepancias en códigos o nombres (Notas: A,AB,B,C,D,F,W).
  - datos duplicados.
- Si no hay calidad en los datos, no hay calidad en los resultados!
  - Las decisiones de calidad deben estar basadas en datos de calidad.
  - Para hacer Data Warehouse se necesita integrar consistentemente datos de calidad.

Ruido

- El ruido se refiere a la modificación de valores originales.
- Ejemplos: distorsión de la voz de una persona cuando habla por teléfono, “nieve” en la pantalla de televisión.



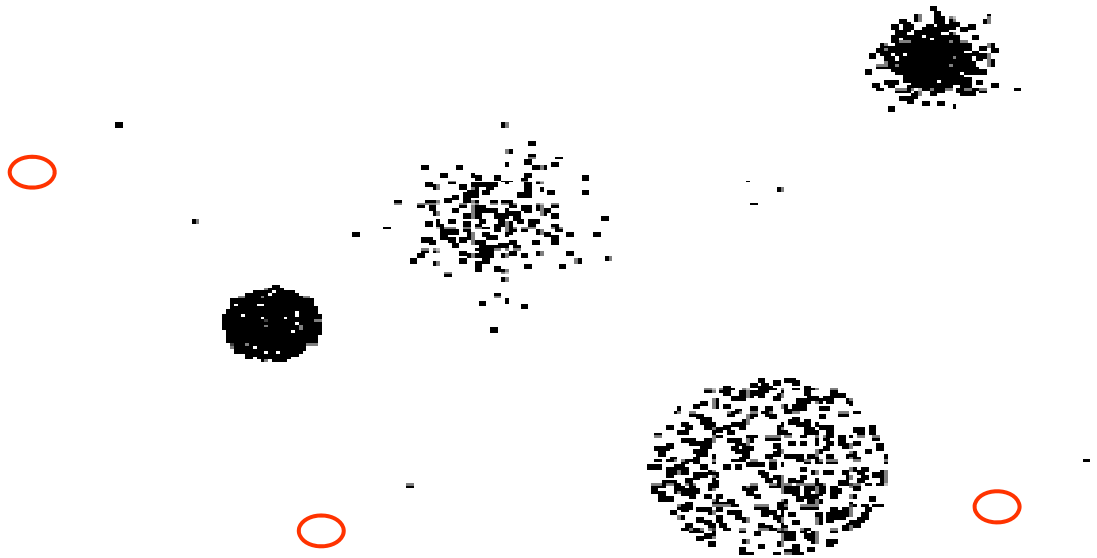
Two Sine Waves



Two Sine Waves + Noise

“Outliers”

- Los “outliers” son datos con características que difieren considerablemente de la mayoría de los datos en el conjunto de datos.



Principales Tareas de Preprocesamiento de Datos

- Limpieza de datos
  - Completar valores faltantes, suavizar datos ruidosos, identificar o eliminar “outliers”, y resolver inconsistencias.
- Integración de datos
  - Integración de múltiples bases de datos.
- Transformación de datos
  - Normalización y totalización.
- Reducción de datos
  - Se obtiene una representación más reducida en volumen pero que produce los mismos o similares resultados analíticos.
- Discretización de datos
  - Parte de la reducción de datos pero con particular importancia, especialmente para datos numéricos.

Limpieza de Datos

- Tareas de la limpieza de datos:
  - Completar valores faltantes.
  - Identificar “outliers” y suavizar datos ruidosos.
  - Corregir datos inconsistentes.

Preprocesamiento - Datos Faltantes

- Los datos no siempre están disponibles.
  - E.g., muchas filas no tienen registrados valores para muchos atributos, tales como los ingresos del cliente en datos de ventas.
- La falta de valores se puede deber a:
  - mal funcionamiento de equipos.
  - inconsistencia con otros datos registrados y por lo tanto eliminados.
  - datos no ingresados debido a equivocaciones o malos entendidos.
  - algunos datos pudieron no considerarse importantes al momento de ingresar datos.
  - No se registró historial o cambios en los datos.
- Puede ser necesario estimar los valores faltantes.

valores faltantes (cont.)

- Se ha propuesto muchos métodos para el tratamiento de valores faltantes. Muchos de estos métodos fueron desarrollados para el tratamiento de valores faltantes en encuestas por muestreo.
- Bello (1995), tratamiento de valores faltantes in regresion
- Troyanskaya et al (2001), tratamiento de datos faltantes en clasificacion no supervisada.
- Estudios relacionados con clasificación supervisada:
  - Chan and Dunn (1972) – Imputation en LDA para problemas con dos clases.
  - Dixon (1975) – Imputacion k-nn para lidiar con valores faltantes en clasificacion supervisada.
  - Tresp (1995)- el problema de valores faltantes en aprendizaje supervisado usando redes neurales.

Valores faltantes (cont.)

- Los valores faltantes son un problema común en análisis estadístico.
- Impacto de valores faltantes:
  - 1% datos faltantes – trivial
  - 1-5% - manejable
  - 5-15% - requiere métodos sofisticados
  - 15% - interpretación perjudicial

Mecanismos de valores faltantes

- i) Valores faltantes completamente al azar *(MCAR):* La probabilidad que una instancia tenga un valor faltante para un atributo es la misma para todas las instancias. Es decir, esta probabilidad no depende ni de los valores observados ni de los valores faltantes. La mayoria de los valores faltantes no son MCAR.
  - Por ejemplo, supongamos que peso y edad son variables de interes en un estudio. Si la probabilidad que una persona dee informacion acerca de su peso es la misma para todas las personas sin tomar en cuenta suy peso y edad, entonces los valores faltantes en el atributo peso son considerados como MCAR.

Este mecanismo es mas adecuado para datos a ser usados en clasificacion nopuervisada.

Mecanismos de valores faltantes

- ii) *Valores faltantes al azar (MAR)*: La probabilidad que una instancia tenga un valor faltante en un atributo depende de los valores observado, como por ejemplo la clase a la cual pertenece la instancia, pero no depende de los valores faltantes. Este mecanismo es mas adecuado para datos usados en clasificacion supervisada.
- *iii) Valores faltantes no al azar o no ignorables(NMAR):* La probabilidad de que una instancia tenga un valor faltante en un atributo depende de los valores faltantes en el conjunto de datos. Ocurre cuando las personas entrevistadas no quieren revelar algo muy personal acerca de ellas. El patron de valores faltantes no es predecible de las otras variables en el conjunto de datos. Este tipo de valores faltantes es el mas dificil de tratar y es el que ocurre mas frecuentemente.

Mecanismos de valores faltantes

- Para conjuntos de datos con un bajo porcentaje de valores faltantes el mecanismo se puede considerar MCAR.
- Para conjuntos de datos con un alto porcentaje de valores faltantes el mecanismo se puede considerar NMAR.
- Para conjuntos de datos con valores faltantes simulados el mecanismo se puede considerar MAR

Ejemplo: Conjunto de datos Census

48842 instancias, contiene variables continuas y discretas (entrenamiento=32561, prueba=16281).

Cuando se eliminan las instancias con valores faltantes quedan 45222 (entrenamiento=30162, prueba=15060).

Disponible en: http://ftp.ics.uci.edu/pub/machine-learning-databases

Donantes: Ronny Kohavi y Barry Becker (1996).

Variables en Census

- 1- age: continua.
- 2- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- 3- fnlwgt (final weight) : continuous.
- 4- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- 5- education-num: continua.
- 6- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- 7- occupation:

Variables en Census

- 8-relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- 9-race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- 10-sex: Female[0], Male[1].
- 11-capital-gain: continua.
- 12-capital-loss: continua.
- 13-hours-per-week: continua.
- 14-native-country: nominal
- 15 Salary: >50K [2], <=50K [1].

Ejemplo: Conjunto de datos census

age	employme	education	edun	marital	...	job	relation	race	gender	hour	country	wealth
					...							
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_fam	White	Male	40	United_Sta	poor
51	Self_emp_	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_fam	White	Male	40	United_Sta	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_fam	White	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_Sta	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_Sta	poor
33	Private	Assoc_acd	12	Never_mar	...	Sales	Not_in_fam	Black	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	...	Craft_repai	Husband	Asian	Male	40	*MissingVa	rich
34	Private	7th_8th	4	Married	...	Transport_	Husband	Amer_India	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	...	Farming_fi	Own_child	White	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	...	Machine_c	Unmarried	White	Male	40	United_Sta	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_Sta	poor
44	Self_emp_	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_Sta	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

Leyendo files de Excel en R

```
#metodo1: Pasando a formato csv
 a=read.csv("c://datos1.csv")
a
# Metodo 2:  Usando  la interface ODBC para bases de datos
#incluye bases de datos en ACCESS, Excel y Dbase
 require(RODBC)
  z = odbcConnectExcel("c://datoscluster.xls")
 dd = sqlFetch(z,"datoscluster")
close(z)
 dd
z = odbcConnectAccess2007("c://census.accdb")
 dd = sqlFetch(z,”Datosprueba")
close(z)
 dd
```

Codificando los datos

Para codificar la j-esiam columa de census en una columna numerica se usa el comando
Censusn[,j]=as.numeric(factor(census[,j]))

Explorando el conjunto de datos

usando ***imagmiss()***

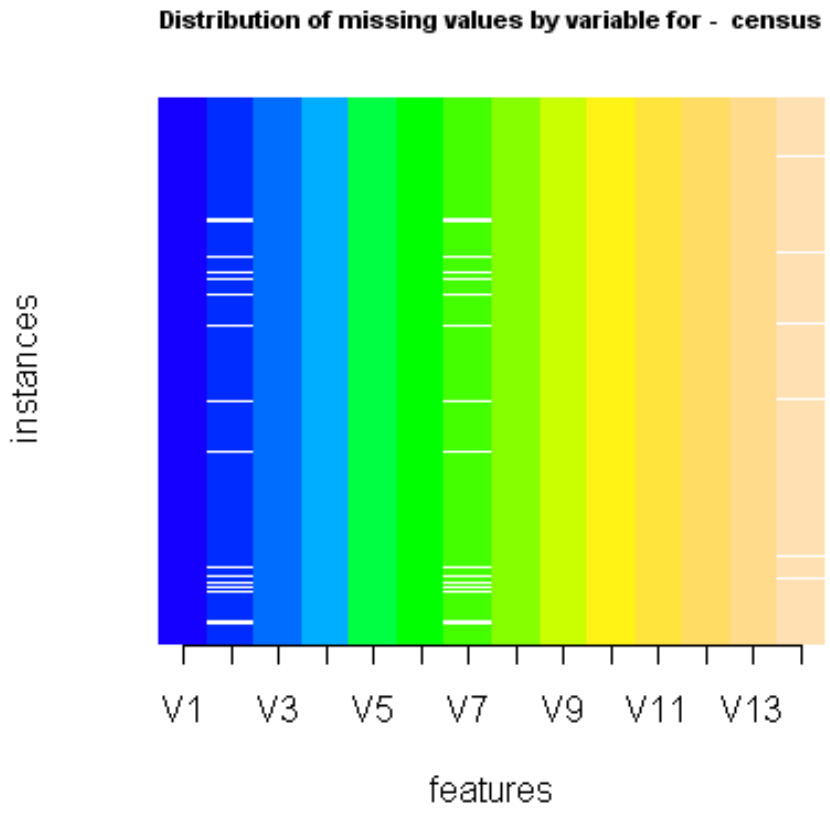
> imagmiss(censusn, name="census")

Reporte de valores faltantes para Census :

Cantidad de valores faltantes sobre el total:
4262
Porcentaje de valores faltantes sobre el total:
0.9349485
Variable con valores faltantes (porcentaje):

V2	V6	V13
5.638647	5.660146	1.790486

Porcentaje de variables con valores faltantes :
21.42857
Cantidad de instancias con valores faltantes :
2399
Porcentaje de instancias con valores faltantes:
7.36771



Funciones en R para Valores Faltantes

- Para detectar las columnas con missing values

  which(colSums(is.na(censusn))!=0)
- Para detectar las filas con missing values

  rmiss=which(rowSums(is.na(censusn))!=0,arr.ind=T)
- Para hallar el porcentaje de filas con missing values

  length(rmiss)*100/dim(censusn)[1]
- Para hallar el porcentaje de missing values por columna

  colmiss=c(2,6,13)

  per.miss.col=100*colSums(is.na(censusn[,colmiss]))/dim(censusn)[1]
- Para eliminar los missing values

  census.omit=na.omit(censusn)

  dim(census.omit)

  [1] 30162 15

La función clean

- Esta función elimina columnas y filas que tienen un gran número de valores faltantes.

```
census.cl=clean(censusn,tol.col=.5,
tol.row=.3,name="cl.census")
```

	Variables	Percent.of.missing
1	V2	5.6386474616873
2	V6	5.66014557292466
3	V13	1.79048555019809

Maximum number of values to be imputed: 4262

Tratamiento de valores faltantes

- Eliminación de casos. Ignorar la fila que contiene datos faltantes. Usualmente es aplicado cuando el valor que falta es el de la clase (asumiendo que se esta haciendo clasificación). No es efectiva cuando el porcentaje de valores faltantes por atributo varía considerablemente.
- Estimación de parámetros, donde los procedimientos de Máxima Verosimilitud que usan variantes del algoritmo EM (Expectation-Maximization) pueden manejar la estimación de parámetros en presencia de valores faltantes.
- Técnicas de Imputación, donde los valores faltantes son reemplazados con valores estimados basados en la información disponible en el conjunto de datos.

Tratamiento de valores faltantes

- En el contexto de clasificación supervisada, se usan cuatro métodos para el tratamiento de valores faltantes:
- Case Deletion (CD) – Este método consiste en descartar todas las instancias (casos) con valores perdidos en por lo menos un atributo. Una variante de este método consiste en determinar el grado de valores faltantes en cada instancia y atributo, y eliminar las instancias y/o atributos con altos niveles de valores faltantes. Antes de eliminar cualquier atributo es necesario evaluar su relevancia en el análisis.

Tratamiento de valores faltantes

- Mean Imputation (MI) – Reemplazar los valores faltantes de un atributo dado por la media de todos los valores conocidos de ese atributo en la clase a la que la instancia con el valor faltante pertenece.
- Median Imputation (MDI). Como la media se ve afectada por la presencia de outliers, parece natural usar la mediana en su lugar para asegurar robustez. En este caso los valores faltantes para un atributo dado es reemplazado por la mediana de todos los valores conocidos de ese atributo en la clase a la que la instancia con el valor faltante pertenece.
- census.mimp=ce.mimp(censusn,"mean",1:13)
- census.mdimp=ce.mimp(censusn,"median",atr=1:13,nomatr=c(2,4:10,13))

Imputación con k-Nearest Neighbor (KNNI)

- Dividir el conjunto de datos D en dos partes. Sea Dm el conjunto que contiene las instancias en las cuales falta por lo menos uno de los valores. Las demás instancias con información completa forman un conjunto llamado Dc.
  - Para cada vector x en Dm:

    A) Dividir el vector en dos partes: una la de información observada y otra la de información faltante, x = [xo; xm].

    B) Calcular la distancia entre xo y todos los vectores del conjunto Dc. Usar solo aquellos atributos en los vectores de Dc que están observados en el vector x.

    C) Usar los K vectores más cercanos (K-nearest neighbors) y realizar un estimado “majority voting” de los valores faltates para los atributos categóricos. Para atributos continuos, reemplazar el valor faltante por la media del atributo en la vecindad k mas cercana (k-nearest neighborhood).

Imputación con k-Nearest Neighbor (KNNI)

- Usualmente, se toma k igual a 10

  census.knn=ec.knnimp(censusn,k=10)

  census.knn=ec.knnimp(censusn,nomatr=c(2,4:10,14),k=10)

Insertando aleatoriamente valores faltantes

```
> dim(mat1)
[1] 6 2
> mat2=as.vector(mat1)
> mat2
 [1] "a"  "b"  "ba" "d"  "c"  "ab" "ac" "ad" "bf" "ba" "ac" "ba"
> mat2[sample(1:12,6)]="NA"
> mat2
 [1] NA   "b"  "ba" NA   NA   "ab" "ac" NA   "bf" NA   NA   "ba"
```

- mat3=matrix(mat2,6,2)

```
> mat3=as.numeric(factor(mat2))
> mat3=matrix(mat3,6,2)
```

Imputando los datos faltantes

```
> mat3
     [,1] [,2]
[1,]   NA    2
[2,]    3   NA
[3,]    4    5
[4,]   NA   NA
[5,]   NA   NA
[6,]    1    4
>#anadiendole una columanb ficticia de clases
>mat3=cbind(mat3,c(1,1,1,1,1,1))
>mat3=cbind(mat3,rep(1,6))
```

Imputando los datos faltantes

```
> ce.mimp(mat3,"mean",1:2)
Summary of imputations using substitution of  mean (mode for nominal features):
     Row Column Class Imput.value
[1,]   1      1     1    2.666667
[2,]   2      2     1    3.666667
[3,]   4      1     1    2.666667
[4,]   4      2     1    3.666667
[5,]   5      1     1    2.666667
[6,]   5      2     1    3.666667
Total number of imputations per class:
Class  1 :  6
Total number of imputations:  6
```

Otros métodos de imputación.

- Hot deck and Cold deck. [nces.ed.gov/statprog]. En Cold deck se usan valores de estudios similares para reemplazar valores perdidos en el estudio actual. En hot deck. En Hot deck se usa valores de atributos correlacionados con el atributo que contiene el valor faltante para sustituirlos.
- Modelo predictivo: Regresión Linear (atributos continuos), Regresión Logística (atributos binarios), logística Polychotomous (atributos nominales). El atributo con valor faltante es usado como la variable de respuesta y los demás atributos son considerados predictoras. Desventajas: Puede crear sesgo, requiere correlación alta entre predictoras. Cómputo lento.

Otros métodos de imputación.

- Imputación Múltiple. Se imputan varias veces los valores faltantes con valores simulados de una distribucion que se asume para cada variable.
- Algoritmo EM.
- Los árboles de decisión tienen su propio enfoque para tratar valores faltantes.

- #imputation using linear regression
- m2

```
      [,1] [,2] [,3] [,4]
 [1,]   26   26   24    3
 [2,]   25   20   25    3
 [3,]   18   13   81    3
 [4,]   22   27   10    4
 [5,]   18   23   13    4
 [6,]   25   19   14    4
 [7,]   27   NA   17    4
 [8,]   37   23   19    4
 [9,]   33   NA   18    4
[10,]   17    5   79    5
[11,]   29   20   23    4
[12,]   22   55    9    4
[13,]   28   23   21    4
[14,]   26   23   17    5
[15,]   24   NA   17    4
[16,]   34   49  169    4
[17,]   11   15    8    4
[18,]   19   NA   21    4
[19,]   17   21   14    4
[20,]   15   17   17    4
[21,]   15   16   16    4
```

```
> l1=lm(V2~.,data=as.data.frame(m2[-c(7,9,15,18),]))
> l1
Call:
lm(formula = V2 ~ ., data = as.data.frame(m2[-c(7, 9, 15, 18),     ]))
Coefficients:
(Intercept)          V1          V3          V4
   10.89763     0.68296     0.03664    -1.09970
> a=as.data.frame(m2[c(7,9,15,18),-2])
> colnames(a)=c("V1","V3","V4")
> a
  V1 V3 V4
1 27 17  4
2 33 18  4
3 24 17  4
4 19 21  4
> lmimp=predict(l1,a)
> lmimp
       1        2        3        4
25.56156 29.69596 23.51269 20.24445
>
```

Preprocesamiento - Normalización

- La normalización de datos consiste en re-escalar los valores de los datos dentro de un rango especificado, tal como -1 a 1 o 0 a 1.
- También es conocido como “normalizacion del rango”.
- En datos de microarreglos se consume mucho tiempo haciendo normalización de la varianza.

Razones para normalizar

- Normalizar los datos de entrada ayudará a acelerar la fase de aprendizaje.
- Los atributos con rangos grandes de valores tendran mas peso que los atributos con rangos de valores mas pequenos, y entonces dominarán la medida de distancia. Por ejemplo, el clasificador K-nearest usando la medida de distancia euclideana depende de que todas las dimensiones de los valores de entrada estén en la misma escala.
- También puede ser necesario aplicar algún tipo de normalización de datos para evitar problemas numéricos tales como pérdida de precisión y desbordamientos aritméticos (overflows).

Conjunto de datos Bupa

Número de instancias: 345

Número de atributos: 7

Descripción de los atributos:

1. Mcv volumen corpuscular
2. alkphos fosfotasa alcalina
3. sgpt alamine aminotransferase
4. sgot aspartate aminotransferase
5. gammagt gamma-glutamyl transpeptidase
6. drinks numero de bebidas alcohoilicas
7. Class: 1 (higado enfermo) y 2 (higado sano)

> bupa[1:20,]

	V1	V2	V3	V4	V5	V6	V7
1	85	92	45	27	31	0	1
2	85	64	59	32	23	0	2
3	86	54	33	16	54	0	2
4	91	78	34	24	36	0	2
5	87	70	12	28	10	0	2
6	98	55	13	17	17	0	2
7	88	62	20	17	9	0.5	1
8	88	67	21	11	11	0.5	1
9	92	54	22	20	7	0.5	1
10	90	60	25	19	5	0.5	1
11	89	52	13	24	15	0.5	1
12	82	62	17	17	15	0.5	1
13	90	64	61	32	13	0.5	1
14	86	77	25	19	18	0.5	1
15	96	67	29	20	11	0.5	1
16	91	78	20	31	18	0.5	1
17	89	67	23	16	10	0.5	1
18	89	79	17	17	16	0.5	1
19	91	107	20	20	56	0.5	1
20	94	116	11	33	11	0.5	1

Normalización Z-score

Los valores V son normalizados en base a la media y desviación estándar.

V' = (V - mean) / std

Este método trabaja bien en los casos en que no se conoce el máximo y mínimo de los datos de entrada o cuando existen outliers que tienen un gran efecto en el rango de los datos.

> zbupa=znorm(bupa)

Normalización Min-Max

Este método realiza una transformación lineal de los datos originales V en el intervalo especificado [newmin,newmax]

V' = (V - min) * (newmax - newmin) / (max - min) + newmin

La ventaja de este método es que preserva exactamente todas las relaciones entre los datos. No introduce ningún potencial sesgo en los datos. La desventaja es que se encontrará un error “fuera del límite” ("out of bounds“) si un futuro ingreso de datos cae fuera del rango original.

- > mmbupa=mmnorm(bupa)

Normalización por escalamiento decimal

Este método realiza la normalización moviendo el punto decimal de los valores. El número de puntos decimales movidos depende del máximo valor absoluto.

$V' = V / 10^j$

donde j es el entero mas pequeño tal que $Max(|V|) < 1$.

Sólo es útil cuando los valores de los atributos son mayores que 1 en valor absoluto.

- > dsbupa=decscale(bupa)

Normalización Sigmoidal

Este método realiza una transformación no linear de los datos de entrada en el rango -1 a 1, usando una función sigmoidal.

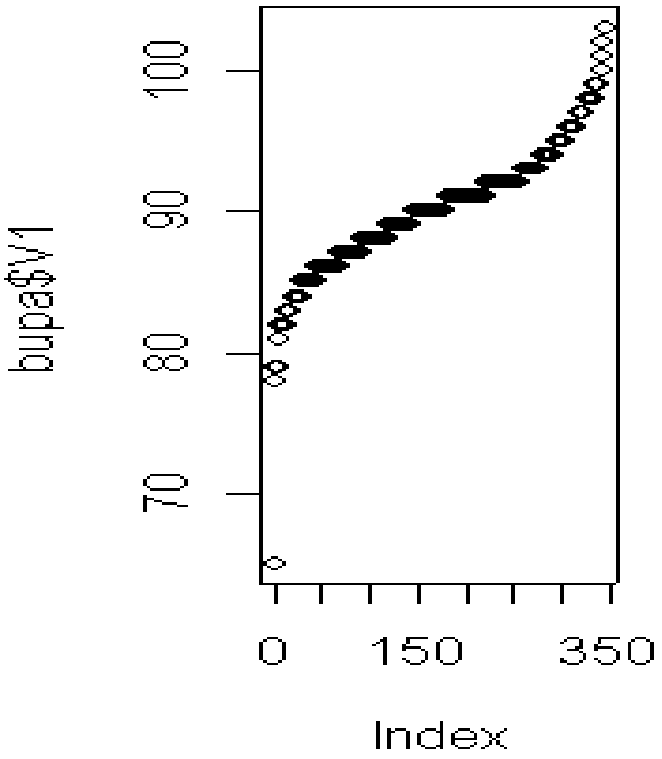
V' = (1 - e^(-a)) / (1 + e^(-a)) donde a = (V - mean) / std

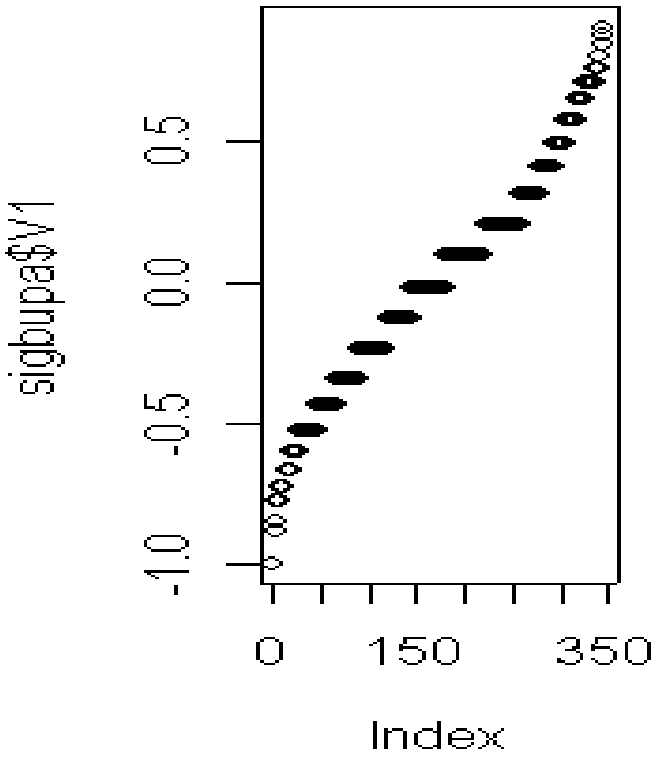
Los datos dentro de una desviación estándar de la media son mapeados a la región casi linear del sigmoide. Los puntos anómalos son comprimidos a lo largo de las colas de la función sigmoidal.

La normalización sigmoidal es especialmente apropiada cuando se tienen datos anómalos que se desean incluir en el conjunto de datos. Este previene que los valores que ocurren más comúnmente sean comprimidos en los mismos valores, sin perder la habilidad de representar grandes valores anómalos.

```
> sigbupa=signorm(bupa)
> plot(sort(bupa$V1))
> plot(sort(sigbupa$V1))
```

Visualización del efecto de la transformación sigmoidal





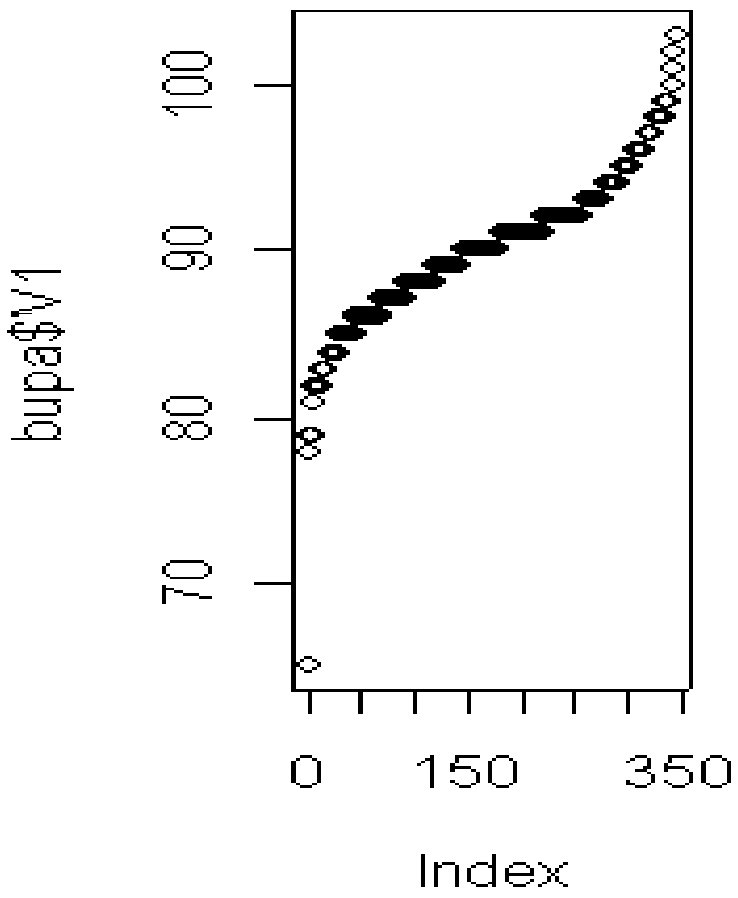
Normalización Softmax

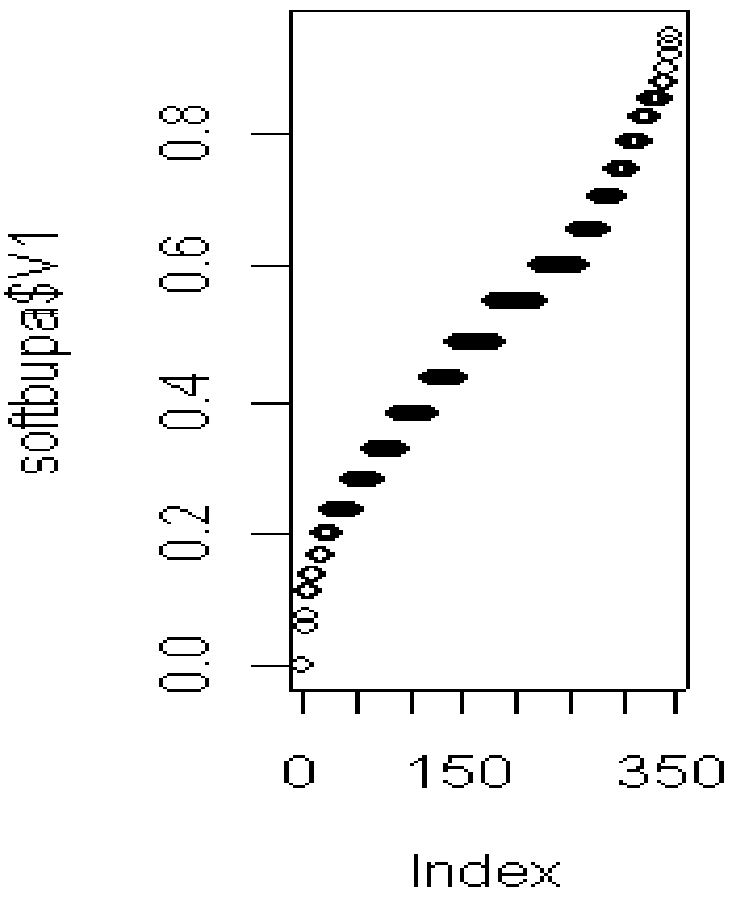
Se llama así porque llega suavemente hacia su valor máximo y mínimo, sin nunca llegar absolutamente. La transformación es mas o menos linear en el rango medio, y tiene una ligera no linearidad a ambos extremos. El rango total cubierto es 0 a 1 y la transformación asegura que no presenta valores que caigan fuera del rango.

V' = 1 / (1 + e^(-a))

donde a= (V - mean) / std

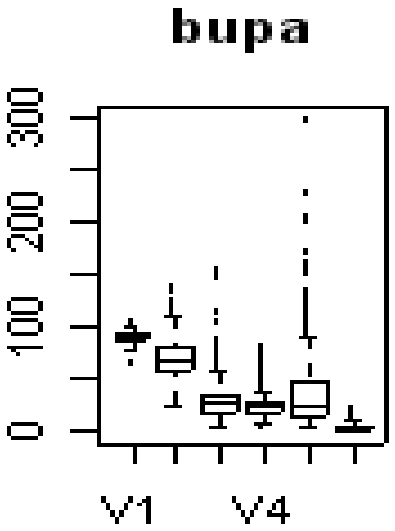
- softbupa=softmaxnorm(bupa)
- zbupa=rangenorm(bupa,method="znorm")

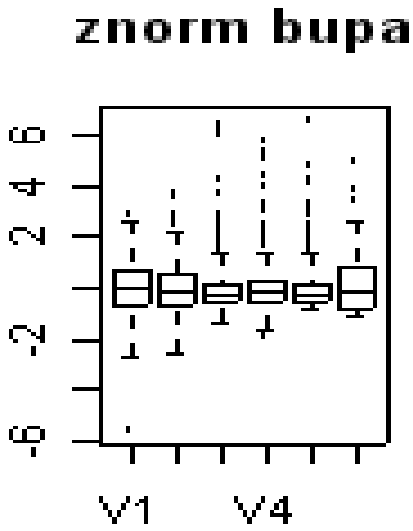

bupa$V1
70
80
90
100
0
150
350
Index

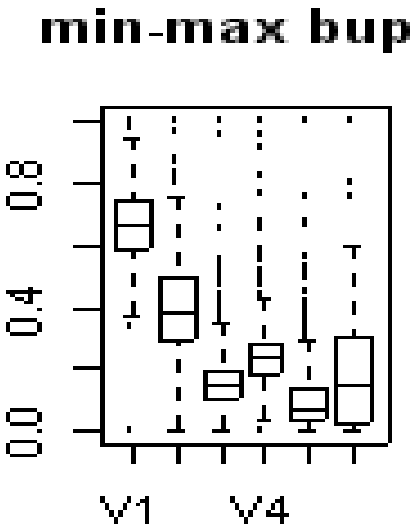

softbupa$V1
0.0
0.2
0.4
0.6
0.8
0
150
350
Index

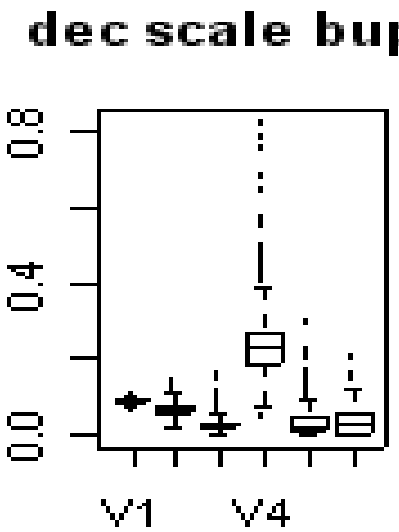
Boxplots to show the effect of the normalization

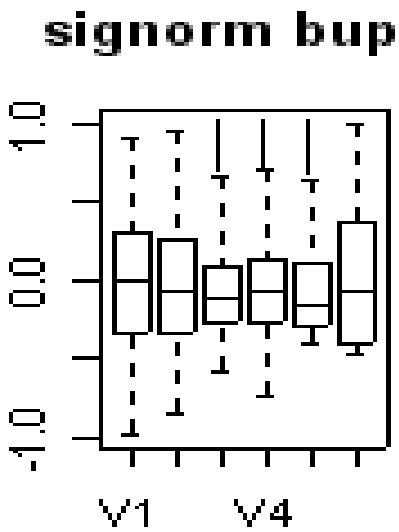
```
> par(mfrow=c(2,3))
> boxplot(bupa[,1:6],main="bupa")
> boxplot(zbupa[,1:6],main="znorm bupa")
> boxplot(mmbupa[,1:6],main="min-max bupa")
> boxplot(dsbupa[,1:6],main="dec scale bupa")
> boxplot(sigbupa[,1:6],main="signorm bupa")
> boxplot(softbupa[,1:6],main="softmax bupa")
```

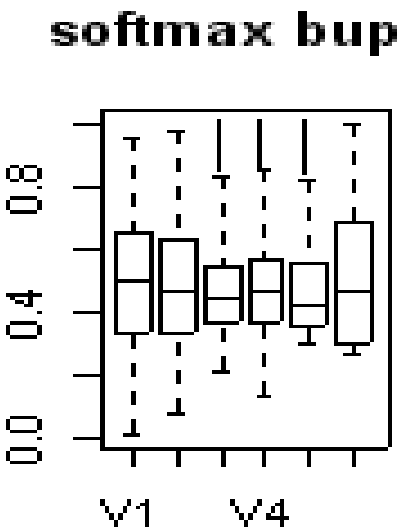

bupa
0
100
200
300
V1
V4


znorm bupa
-6
-2
2
4
6
V1
V4


min-max bupa
0.0
0.4
0.8
V1
V4


dec scale bupa
0.0
0.4
0.8
V1
V4


signorm bupa
-1.0
0.0
1.0
V1
V4


softmax bupa
0.0
0.4
0.8
V1
V4