COMP 6838 Data Mining

LECTURE 3: Data Preprocessing:Missing Values and Normalization

Dr. Edgar Acuna
Department of Mathematics

University of Puerto Rico- Mayaguez

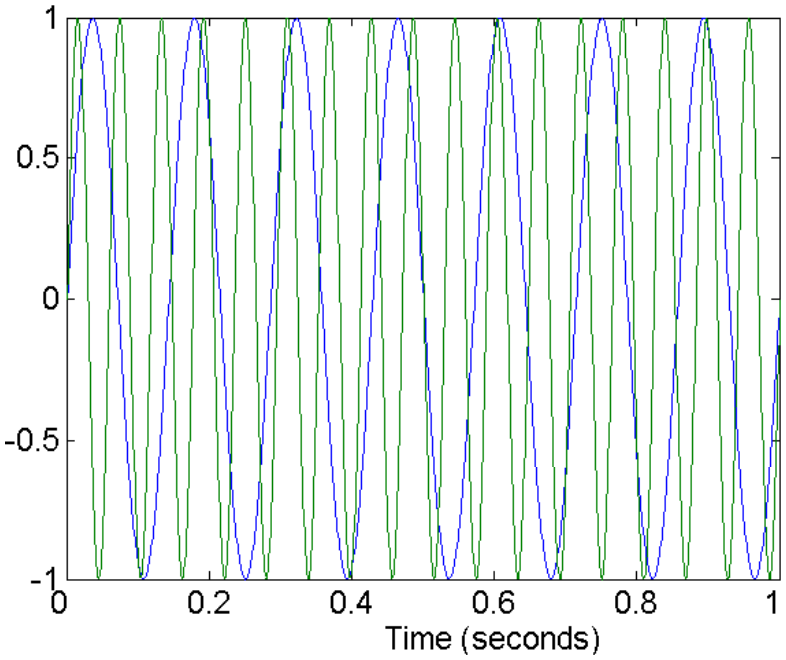
math.uprm.edu/~edgar

Why Data Preprocessing?

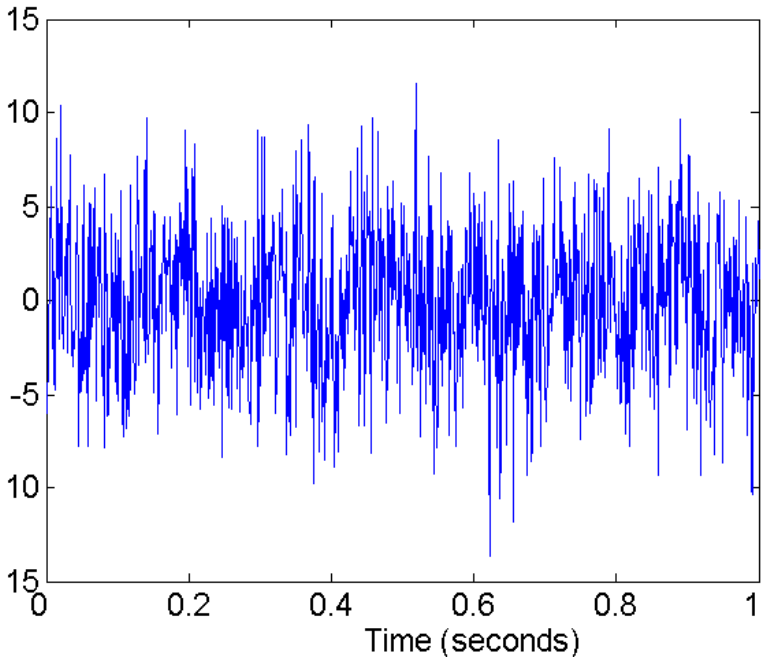
- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names(Grades: A,AB,B,C,D,F,W).
  - Duplicate data.
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data

Noise

- Noise refers to modification of original values
  - Examples: distortion of a person's voice when talking on a poor phone and "snow" on television screen



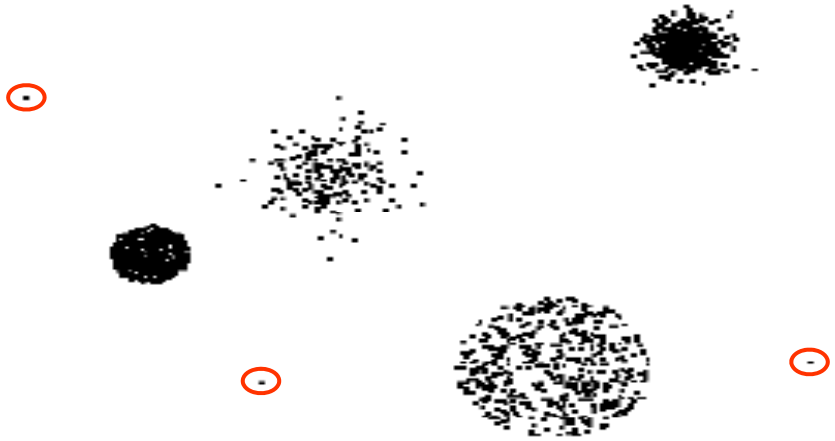
Two Sine Waves



Two Sine Waves + Noise

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases.
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Missing Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.

Acuna and Rodriguez, IFCS 2004

- **Several methods have been proposed in literature to treat missing data. Many of these methods were developed for dealing with missing data in sample surveys.**
- **Bello (1995), MV in regression**
- **Troyanskaya et al (2001), MV in unsupervised classification.**
- **Studies related to supervised classification**
  - **Chan and Dunn (1972) – Imputation on LDA for two class problems.**
  - **Dixon (1975) - k-nn imputation technique for dealing with missing values in supervised classification.**
  - **Tresp (1995)- missing value problem in a supervised learning context for neural networks.**

Background

- **Missing data is a common problem in statistical analysis.**
- **Impact of missing data**
  - **1% missing data – trivial**
  - **1-5% - manageable.**
  - **5-15% - requires sophisticated methods**
  - **15% - detrimental interpretation.**

Missing value mechanisms

- i) *Missing completely at random (MCAR):* The probability of an instance having a missing value for an attribute is the same for all the instances. This means that such probability does not depend on either the known values or the missing data. Most missing data are not MCAR.
  - As an example, suppose weight and age are variables of interest for a particular study. If the likelihood that a person will provide his or her weight information is the same for all individuals regardless of their weight or age, then the missing data in the attribute weight is considered to be MCAR.

This mechanism is more suitable for data to be used in an unsupervised classification task.

Missing value mechanisms

- ii) *Missing at random (MAR)*: The probability of an instance having a missing value for an attribute depends on a known value, such as the class to which the instance belongs, but does not depend on the missing data itself. This mechanism is more suitable for data to be used in a supervised classification task.
- *Non-ignorable or Not Missing at Random (NMAR):* The probability that a value of an attribute is missing depends on the missing data itself. It commonly occurs when interviewed people do not want to reveal something very personal or unpopular about themselves. The pattern of data missingness is non-random and it is not predictable from other variables in the database. Typically this type of missing data is the hardest condition to deal with, but unfortunately, the most likely to occur as well.

Missing value mechanisms

- **missing randomness for the datasets with a low percentage of missing values: MAR.**
- **For datasets where the percentage of missing values is high: NMAR.**
- **missing randomness for the datasets that contained simulated missing values: MAR**

The Census dataset

48842 instances, mix of continuous and discrete features (train=32561, test=16281).

if instances with unknown values are removed only 45222 (train=30162,test=15060) remain.

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

Donantes: Ronny Kohavi y Barry Becker (1996).

Variables in Census

- 1-age: continuous.
- 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: continuous.
- 6-marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- 7-occupation:

Variables in 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: continuous.
- 12-capital-loss: continuous.
- 13-hours-per-week: continuous.
- 14-native-country: nominal
- 15 Salary: >50K (rich)[2], <=50K (poor)[1].

Example: The census data set

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
:	:	:	:	:	:	:	:	:	:	:	:	:

Reading Excel files into R

```
#metho1:  Uisng the csv format
 a=read.csv("c://datos1.csv")
a
# Method 2:  Using   the ODBC interface
#include  databases in ACCESS, Excel and 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
```

Codifying the data

To codify the j-th column of census in a numeric column use the comand

Censusn[,j]=as.numeric(factor(census[,j]))

Exploring the dataset using *imagmiss()*

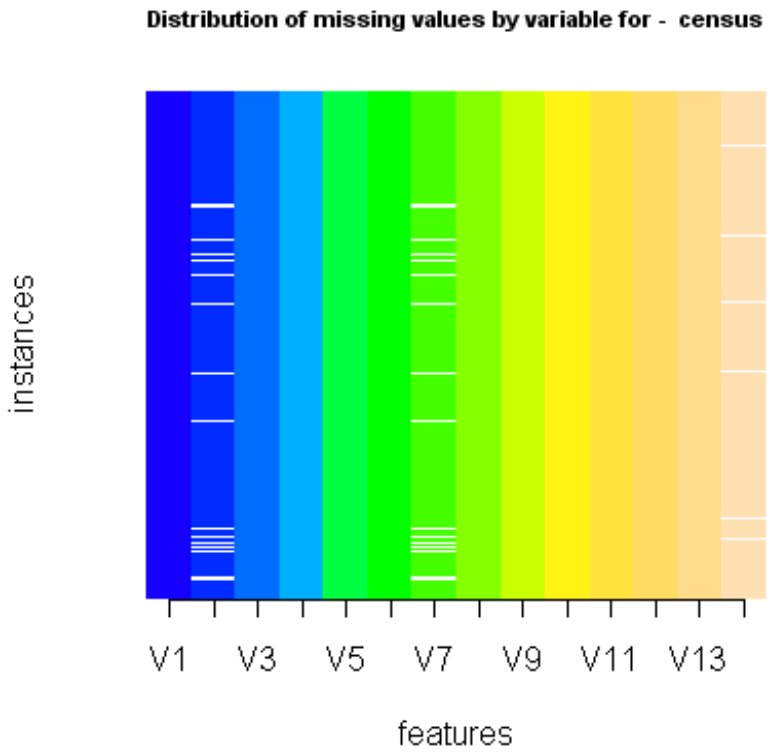
> **imagmiss(data, name="dataname")**

Report on missing values for Census :

Number of missing values overall: 4262
Percent of missing values overall: 0.9349485
Features with missing values (percent):

V2	V7	V14
5.638647	5.660146	1.790486

Percent of features with missing values: 21.42857
Number of instances with missing values: 2399
Percent of instances with missing values: 7.36771



R functions for MV

- 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 missings
  length(rmiss)*100/dim(censusn)[1]
- Para hallar el porcentaje de missings 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

The clean function

- This function deletes columns and rows with a large number of missing values
- census.cl=clean(censusn,tol.col=.5,tol.row=.3,name="cl.census")
  - Variables Percent.of.missing
  - 1 V2 5.6386474616873
  - 2 V7 5.66014557292466
  - 3 V14 1.79048555019809
  - Maximum number of values to be imputed: 4262

Treating Missing values

- Case/Pairwise Deletion. Ignore the tuple: usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Parameter estimation, where Maximum likelihood procedures that use variants of the Expectation-Maximization algorithm can handle parameter estimation in the presence of missing data.
- Imputation techniques, where missing values are replaced with estimated ones based on information available in the data set.

Treating Missing Values

Four methods are used in this study to treat missing values in the supervised classification context:

- **Case Deletion (CD) - This method consists of discarding all instances (cases) with missing values for at least one feature. A variation of this method consists of determining the extent of missing data on each instance and attribute, and delete the instances and/or attributes with high levels of missing data. Before deleting any attribute it is necessary to evaluate its relevance to the analysis.**

Treating Missing Values

- **Mean Imputation (MI) - Replace the missing data for a given feature (attribute) by the mean of all known values of that attribute in the class to which the instance with missing attribute belongs.**
- **Median Imputation (MDI). Since the mean is affected by the presence of outliers it seems natural to use the median instead just to assure robustness. In this case the missing data for a given feature is replaced by the median of all known values of that attribute in the class to which the instance with missing feature belongs.**
  - **census.mimp=ce.mimp(censusn,"mean",atr=c(2,6,13))**
  - **census.mdimp=ce.mimp(censusn,"median",atr=c(2,6,13),nomatr =c(2,4:10,14),"")**
  - **Missing values for nominal attributes are replaced by the mode.**

Imputation with k-Nearest Neighbor (KNNI)

Divide the data set D into two parts. Let D_m be the set containing the instances in which at least one of the features is missing. The remaining instances with complete feature information form a set called D_c.

For each vector x in D_m:

- **A) Divide the instance vector into observed and missing parts as $x = [xo; xm]$.**
- **B) Calculate the distance between the xo and all the instance vectors from the set D_c. Use only those features in the instance vectors from the complete set D_c, which are observed in the vector x.**
- **C) Use the K closest instances vectors (K-nearest neighbors) and perform a majority voting estimate of the missing values for categorical attributes. For continuous attributes replace the missing value by the mean of the attribute in the k-nearest neighborhood.** Also a weighted mean can be used.

Imputation with k-Nearest Neighbor (KNNI)

- Usually k is taken as 10

```
census.knn=ec.knn.imp(censusn,atr=c(2,6,13),k1=10)
census.knn=ec.knn.imp(censusn,natr=c(2,6,13),k1=10)
```

Other imputation methods

- Hot deck and Cold deck. [nces.ed.gov/statprog]
- Prediction model: Linear regresion (continuous attribute), Logistic regresion(Binary attribute), Polychotomous logistic (nominal attributes). The attribute with the missing value is used as the response variable and the remaining attributes are considered predictors. Drawbacks: It can created bias, requires high correlation among predictors. Slow computation.
- Multiple imputation
- The EM algorithm
- Decision trees have their own approach to treat missing values.

- #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
>
```

Missing Imputation in Weka

Weka performs only median imputation for numerical attributes and mode imputation for categorical attributes.
After opening an arff file follow the sequence
filters>unsupervised>attributes>ReplaceMissing\Values

Experimental Methodology

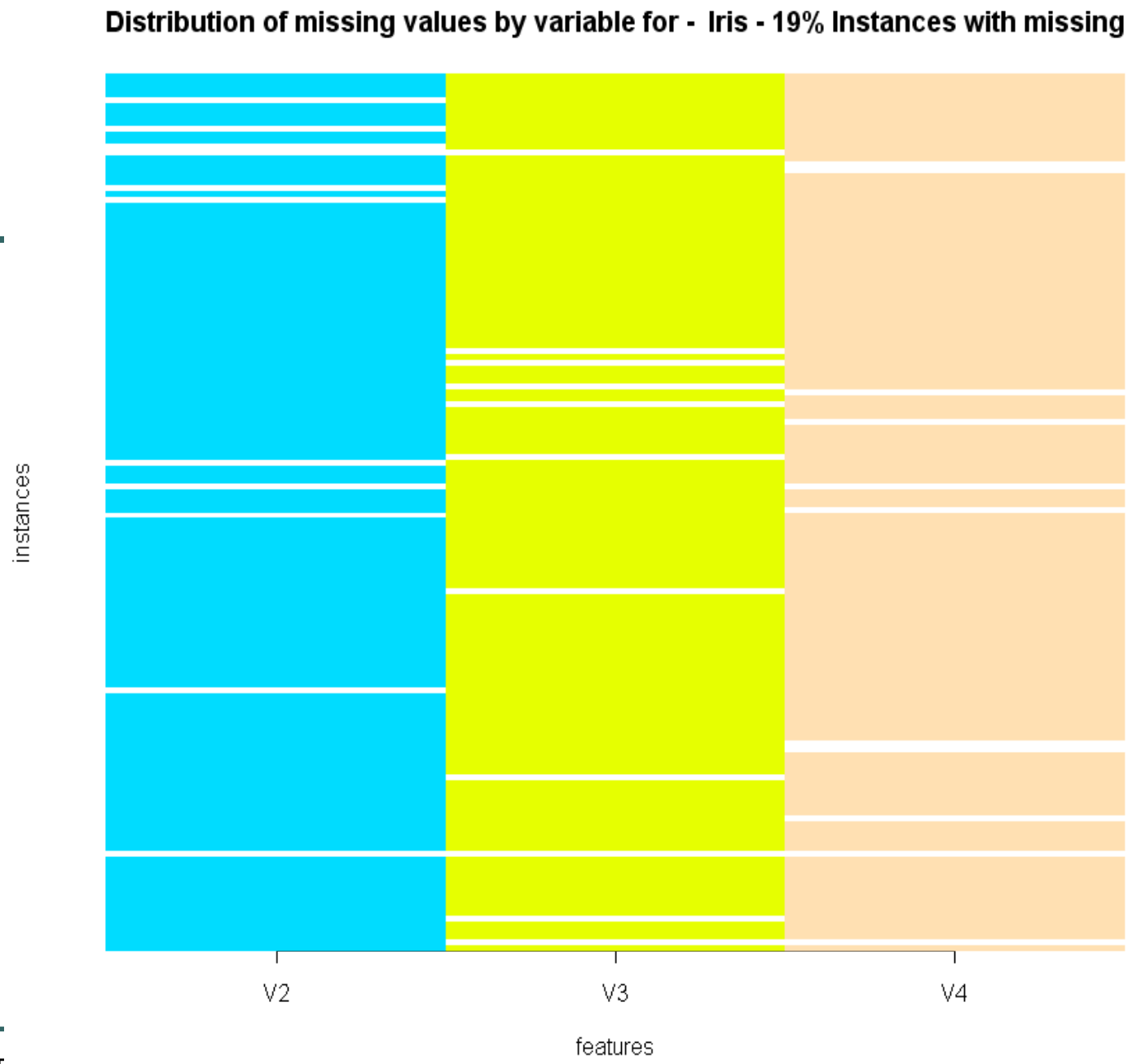
- **We use twelve datasets coming from the Machine Learning Database Repository at the University of California, Irvine, six of them containing missing values.**
- **A simulation experiment was carried out on the six complete datasets as follows:**
  - **A given percentage of missing values was inserted randomly in relevant variables. The percentage of missing values varied from 1 to 21%. The missing values were distributed proportionally according to the class sizes.**
  - **Case deletion and missing value imputation (mean, median, knn) were applied in each simulated set.**
  - **10-fold cross-validation estimates of the misclassification error rate for the LDA, knn, and Rpart classifiers were computed for each experiment.**
  - **Missing data mechanism for the datasets that contained simulated missing values: MAR**

Experimental Methodology

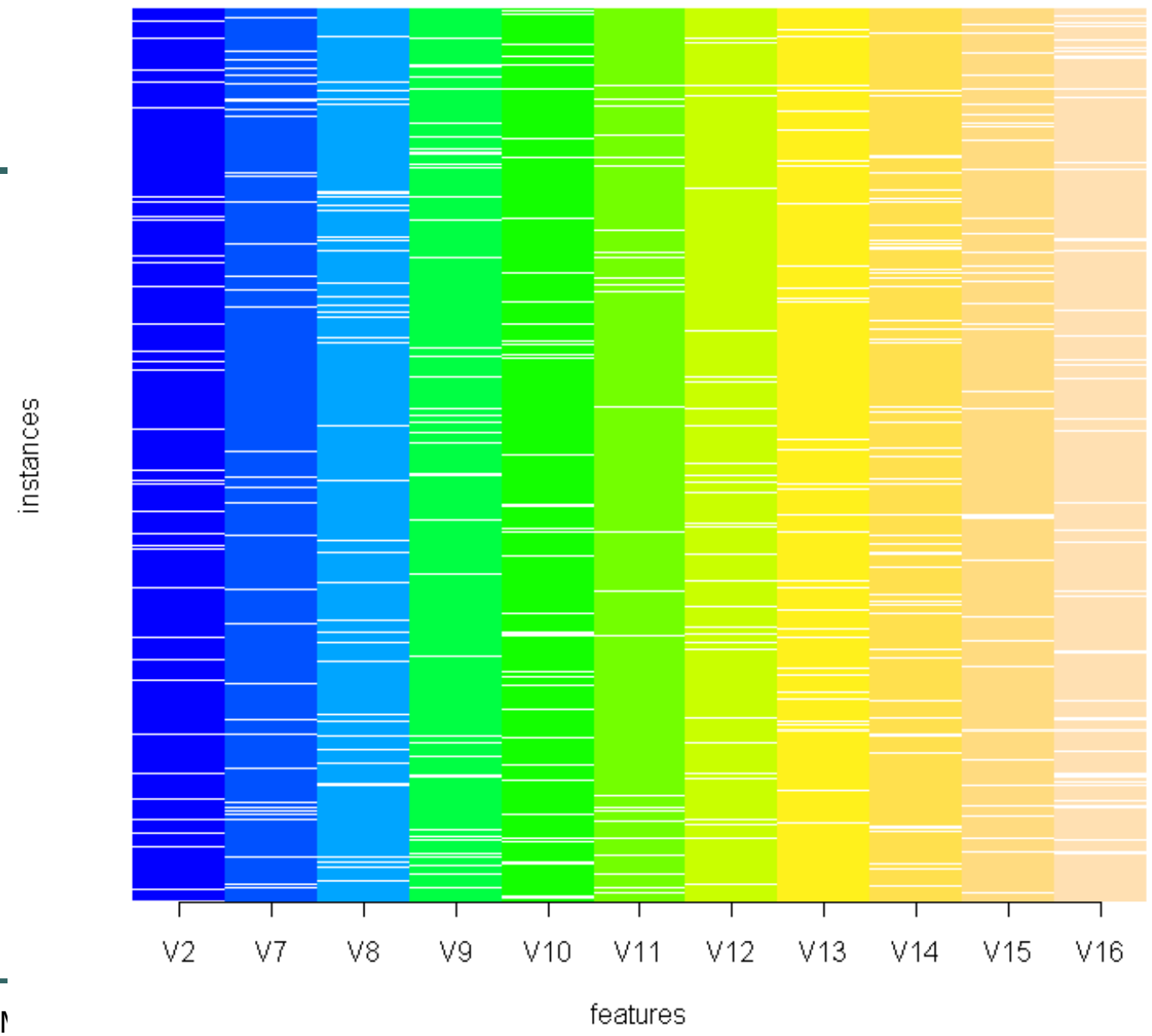
- **For the six datasets containing missing values:**
  - **Datasets were cleaned to eliminate variables with more than 50% missing data and instances with more than 30% missing data.**
  - **Missing values were imputed in the original data set using Mean, Median and K-nn imputation.**
  - **10-fold cross-validation estimates of the misclassification error rate for the LDA, Knn, and rpart classifiers were computed for each experiment.**
  - **missing data mechanism for the datasets that contained missing values: NMAR.**

Description of Datasets

Dataset	Instances	Classes (number, size)	Features (relevant)	Missing Values(%)	Instances with Missing (%)
Iris	150	3 (50,50,50)	4(3)	0	0
Hepatitis	155	2 (32, 123)	19	5.67	48.38
Heartc	303	2(164,139)	13(6)	0.15	1.98
Bupa	345	2 (145, 200)	6(3)	0	0
Crx	690	2 (383,307)	15(9)	0.64	5.36
Breastw	699	2 (458,241)	9(5)	0.25	2.28
Diabetes	768	2(500, 268)	8(5)	0	0
Vehicle	846	4 (218, 212, 217, 199)	18(10)	0	0
German	1000	2 (700, 300)	20(13)	0	0
Segment	2310	7 (330, 330, 330, 330, 330, 330, 330)	19(11)	0	0
Sick*	3772	2 (3541, 231)	28	2.17	29.93
Census	32561	2 (24720, 7841)	15	0.93	7.36


Distribution of missing values by variable for - Iris - 19% Instances with missing
instances
V2
V3
V4
features

COMP


Distribution of missing values by variable for - Segment - 55% missing
instances
V2
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
features

CON

Datasets with missings - Description

	Heartc (6)	Breastw (16)	CRX (67)	Census (4262)	Sick (2292)	Hepatitis (167)
% of missing by attribute	V12* (1.32)	V6* (2.29)	V1 (1.73)	V2 (5.63)	V1 (0.02)	V4 (0.65)
	V13* (0.66)		V2 (1.74)	V7 (5.76)	V2 (3.97)	V6 (0.65)
			V4 (0.87)	V14 (1.79)	V18 (9.78)	V7* (0.65)
			V5 (0.87)		V20* (20.38)	V8 (0.65)
			V6 (1.30)		V22 (2.45)	V9 (6.45)
			V7 (1.30)		V24 (10.25)	V10 (7.09)
			V14* (1.88)		V26* (10.20)	V11(3.23)
						V12 (3.23)
						V13 (3.23)
						V14 (3.23)
						V15 (3.87)
						V16* (18.71)
						V17 (2.58)
						V18* (10.32)
						V19*(43.23)
Missing values(%)	0.15	0.25	0.64	0.93	2.17	5,67
rows with missing(%)	1.98	2.29	5.36	7.36	29.93	48.38

Exploring the dataset using *imagmiss()*

> **imagmiss(data, name="dataname")**

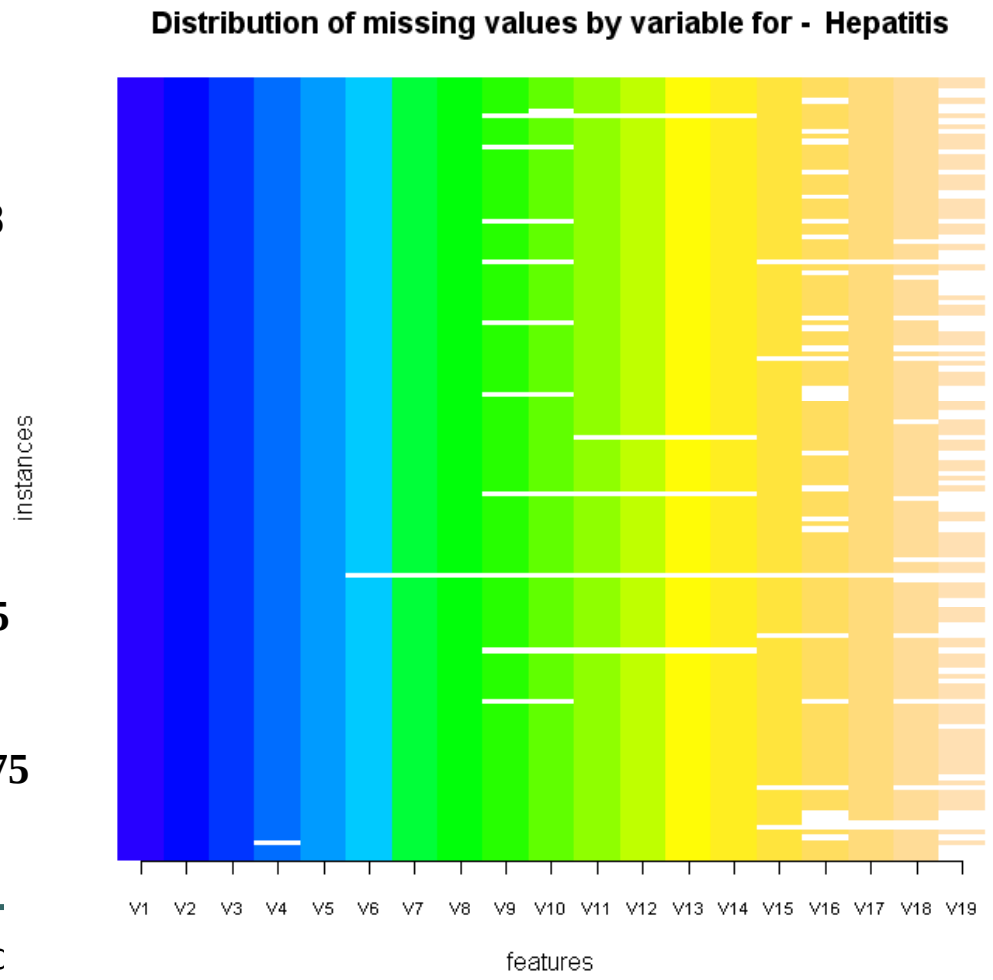
Report on missing values for Hepatitis :

Number of missing values overall: 167
Percent of missing values overall: 5.670628
Features with missing values (percent):

V4	V6	V7	V8	V9	V10	V11	V12	V13	V14
0.65	0.65	0.65	0.65	6.45	7.10	3.23	3.23	3.23	3.23

V15	V16	V17	V18	V19
3.87	18.71	2.58	10.32	43.23

Number of features with missing values: 15
Percent of features with missing values: 78.94737
Number of instances with missing values: 75
Percent of instances with missing values: 48.3871



Results LDA – Datasets with simulated Missing Values

Dataset	% Missing (overall)	% Missing (cases)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Iris	1	2	3.18	0.9088	1.2013	1.1698	1.1447
	5	13.33	3.18	0.9124	1.1424	1.1465	0.8242
	9	23.33	3.18	1.3590	1.2344	1.2470	1.1299
	13	32.67	3.18	1.2013	0.934	0.9937	0.956
Heartc*	5	26.93	16.51	0.785	0.8989	0.8546	0.9485
	11	50.16	16.51	1.0903	0.8807	0.8328	0.9219
	15	63.63	16.51	1.2292	0.7969	0.7047	0.8197
	21	75.75	16.51	0.7117	0.7838	0.6445	0.8262
Bupa	3	8.4	35.14	0.9807	1.0051	1.0062	0.9713
	9	25.21	35.14	1.0248	1.0811	1.0829	0.9983
	17	44.34	35.14	1.0780	1.0331	1.0545	0.9920
	21	51.30	35.14	0.7920	0.9975	1.0197	0.8686
Breastw*	3	13.32	3.66	0.9836	1.0055	0.9672	1.0683
	11	43.77	3.66	1.2787	0.9453	0.9809	1.0328
	15	55.93	3.66	1.2368	0.7492	0.7715	0.8634
	21	70.13	3.66	1.3798	0.8005	0.847	0.9481

Results LDA – Simulated Missing Values

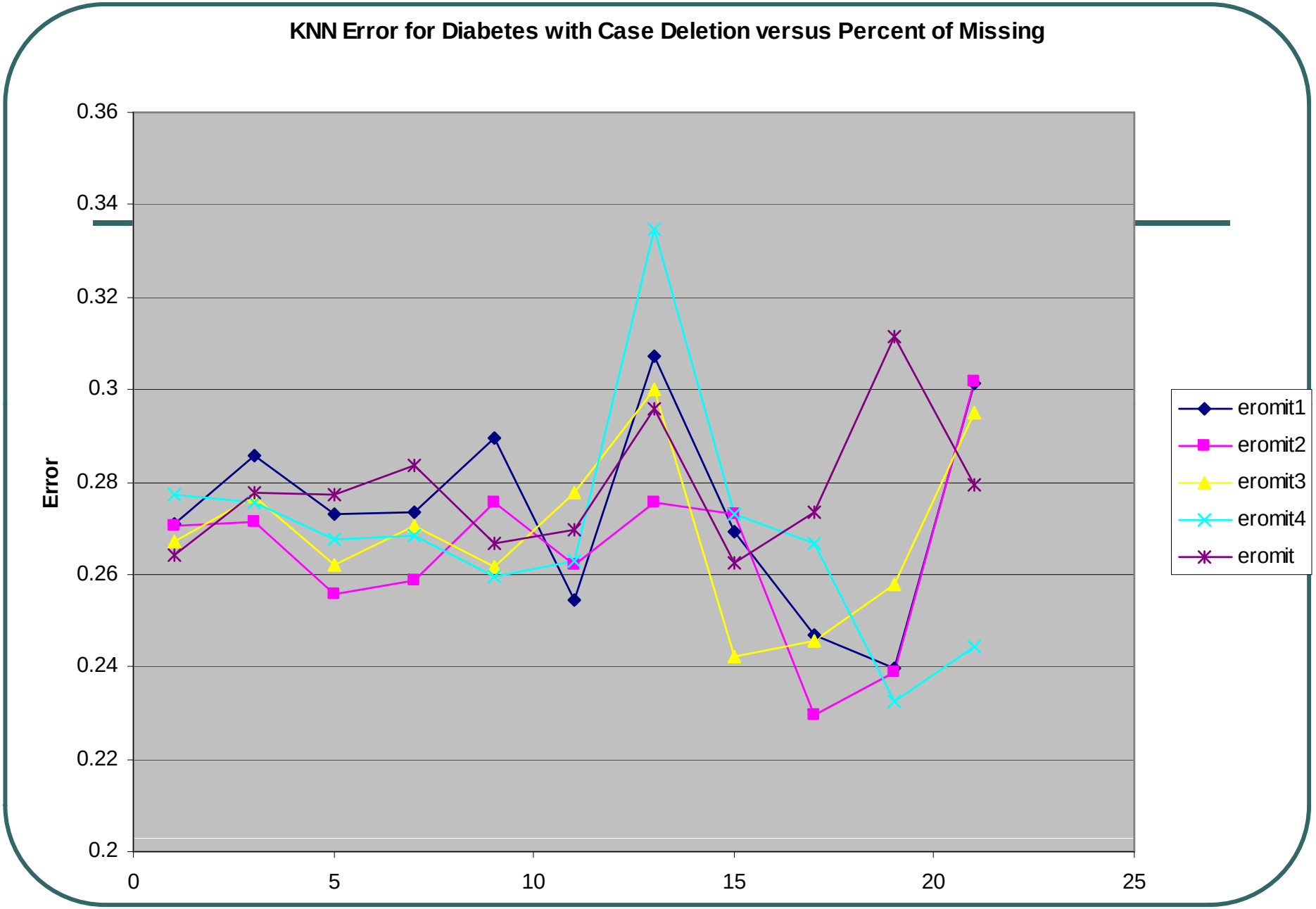
Dataset	% Missing (overall)	% Missing (cases)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Diabetes	3	14.32	24.64	1.0341	1.0142	1.011	1.011
	11	45.05	24.64	0.9826	0.9688	0.9692	0.9765
	15	55.34	24.64	0.8996	0.9430	0.9401	0.9380
	21	69.53	24.64	1.0629	0.9436	0.9224	0.9322
Vehicle	5	38.65	29.15	1.0621	1.0388	1.0415	0.9897
	13	74.11	29.15	1.0604	1.1832	1.1942	0.9883
	17	84.51	29.15	1.1346	1.1523	1.1432	0.9909
	21	90.89	29.15	1.1252	1.1979	1.1485	1.1235
German	5	48.4	24.38	1.0685	0.9934	0.9959	1.0008
	9	70.4	24.38	1.2250	0.9969	0.9743	0.9912
	13	84.8	24.38	1.0665	0.9795	0.9311	0.9828
	19	94.6	24.38	1.1952	0.9081	0.8831	0.9680
Segment	5	42.9	9.15	0.9705	1.0186	1.0240	1.0021
	9	64.89	9.15	1.2504	0.9546	0.9679	0.9626
	13	78.74	9.15	0.906	1.0219	1.0317	0.9661
	17	86.75	9.15	1.0352	0.8915	0.9006	0.9355
	21	92.16	9.15	0.9792	0.8536	0.8404	0.8175

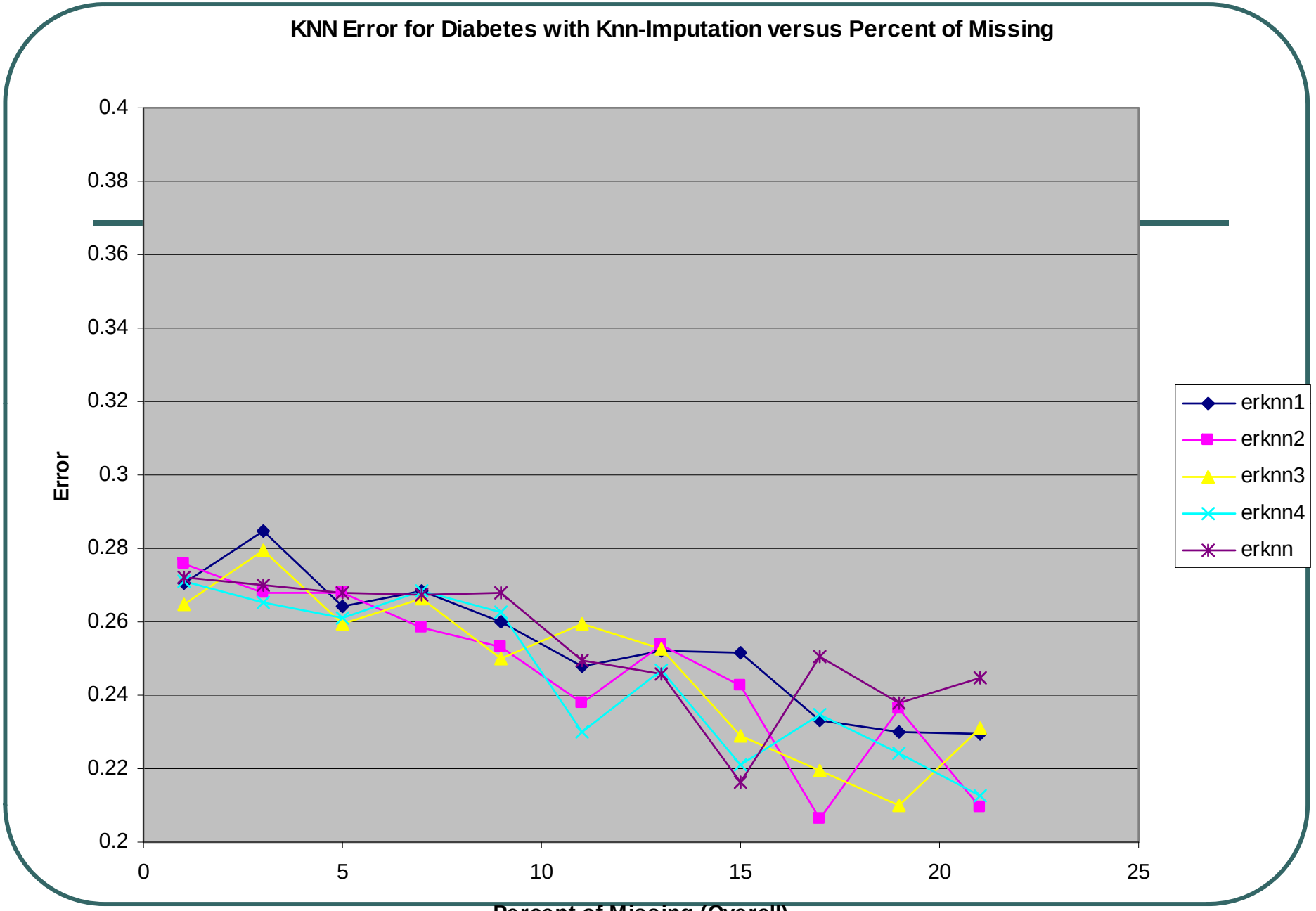
Results KNN classifier – Simulated Missing Values

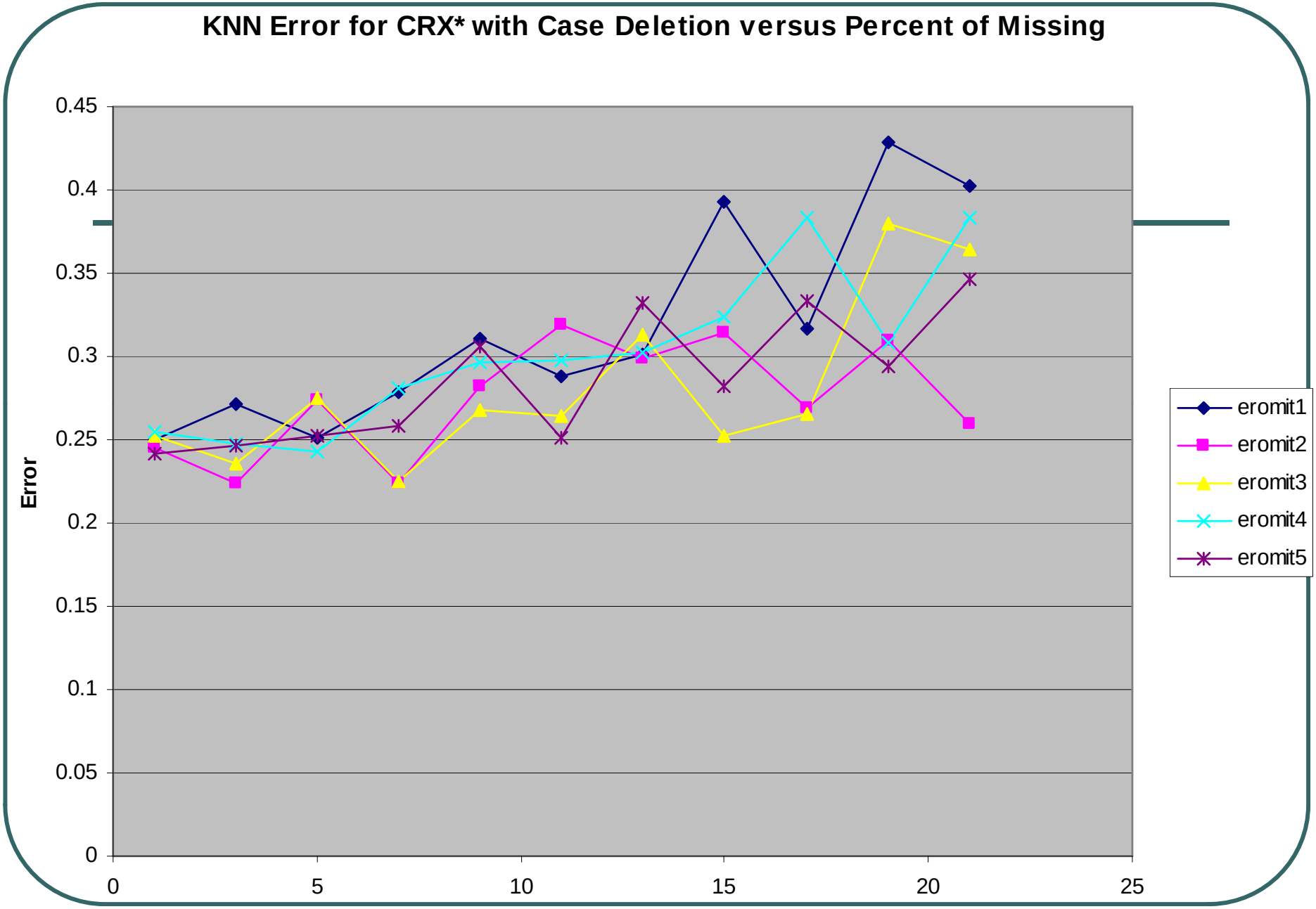
Dataset	% Missing (overall)	% Missing (instances)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Iris	1	2	4.68	1.0299	1.0278	1.0043	1.0385
	5	21.33	4.68	1.2586	1.0171	1.0021	0.9936
	13	33.33	4.68	0.9145	0.4872	0.5662	0.7350
Heartc*	5	31.64	18.55	1.0421	0.9342	0.9536	0.9369
	11	56.9	18.55	0.6367	0.9574	0.9655	0.834
	21	79.46	18.55	0.8216	0.7121	0.7596	0.8259
Bupa	3	8.69	36.49	0.9411	0.9792	0.9553	0.9553
	11	30.14	36.49	0.9619	0.9197	0.9164	0.9268
	19	46.95	36.49	1.0271	0.9720	1.0022	0.9942
Breastw	3	14.64	3.41	0.9677	0.9736	0.956	0.9765
	11	45.09	3.41	0.9795	0.827	0.827	0.8387
	21	66.18	3.41	0.6217	0.5631	0.5777	0.607

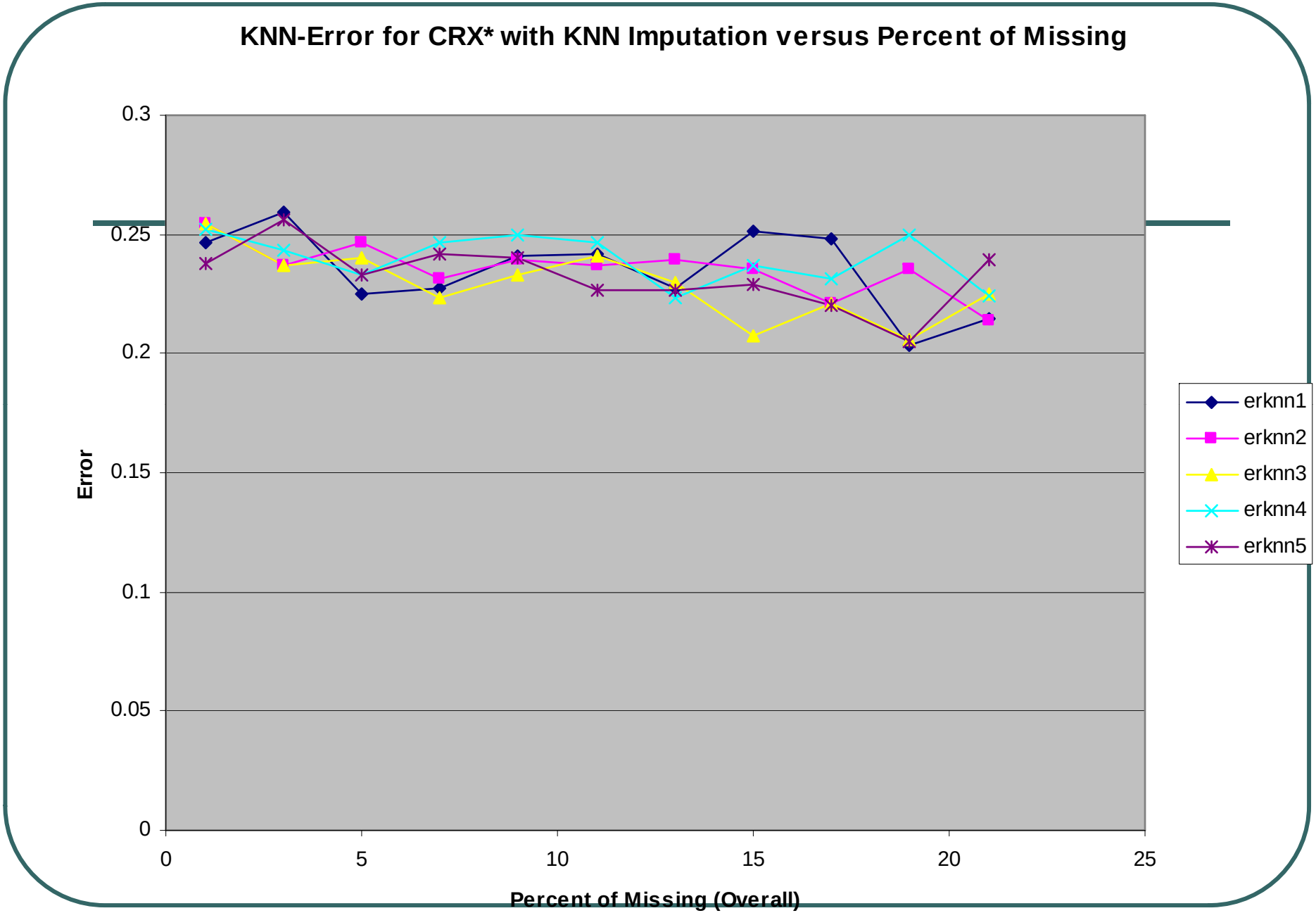
Results KNN – Simulated Missing Values

Dataset	% Missing (overall)	% Missing (instances)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Diabetes	3	14.32	27.47	1.0076	0.9687	0.9709	0.9651
	11	45.31	27.47	0.8580	0.8198	0.7674	0.7951
	21	66.88	27.47	0.9884	0.6858	0.5872	0.7477
Vehicle	5	42.08	34.87	1.1012	1.0419	1.0023	0.9535
	13	74.82	34.87	1.1557	0.9687	0.9415	0.9295
	21	92.55	34.87	1.2234	0.916	0.9036	0.8652
German	5	48.7	29.78	1.091	0.9681	0.9785	0.954
	13	83.7	29.78	1.0121	0.823	0.8251	0.9103
	19	93	29.78	1.1061	0.8103	0.7898	0.8408
Segment	5	43.03	4.64	1.403	1.3448	1.3168	0.9461
	13	77.96	4.64	2.028	1.5172	1.4224	1.1444
	21	92.46	4.64	1.9655	1.653	1.4677	1.0927


KNN Error for Diabetes with Case Deletion versus Percent of Missing
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
Error
0
5
10
15
20
25
Percent of Missing (Overall)
eromit1
eromit2
eromit3
eromit4
eromit


KNN Error for Diabetes with Knn-Imputation versus Percent of Missing
Error
0.4
0.38
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0
5
10
15
20
25
Percent of Missing (Overall)
erknn1
erknn2
erknn3
erknn4
erknn


KNN Error for CRX* with Case Deletion versus Percent of Missing
Error
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
Percent of Missing
eromit1
eromit2
eromit3
eromit4
eromit5


KNN-Error for CRX* with KNN Imputation versus Percent of Missing
0.3
0.25
0.2
0.15
0.1
0.05
0
Error
0
5
10
15
20
25
Percent of Missing (Overall)
erknn1
erknn2
erknn3
erknn4
erknn5

Results Rpart – Simulated Missing Values

Dataset	% Missing (overall)	% Missing (instances)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Iris	1	2	6.66	1.03904	1.003	1.02102	1.02102
	7	20	6.66	0.7958	0.43994	0.46246	0.43544
	13	33.33	6.66	0.98498	0.99099	1.06456	1.05706
Heartc*	5	29.62	17.75	0.97239	1.07211	1.04507	1.01915
	11	58.25	17.75	1.47606	0.92958	0.87268	1.03437
	21	83.16	17.75	1.36338	1.11944	1.08169	0.96789
Bupa	3	8.98	37.87	0.96884	0.96462	0.94243	0.98125
	11	30.72	37.87	0.90863	0.85556	0.88566	0.85001
	19	47.53	37.87	0.98495	0.70003	0.70267	0.76076
Breastw*	3	13.61	4.69	0.9979	0.9702	0.9787	0.9936
	11	44.5	4.69	1.2559	0.791	0.7932	0.7847
	21	68.37	4.69	2.0192	0.9574	0.9211	0.9424

Results Rpart – Simulated Missing Values

Dataset	% Missing (overall)	% Missing (instances)	Base error	CD_Ratio	MI_Ratio	MDI_Ratio	KNNI_Ratio
Diabetes	3	14.45	27	0.9756	0.98	0.9682	0.9733
	11	45.7	27	0.9822	0.7559	0.7707	0.7896
	21	68.22	27	1.0822	0.6537	0.6537	0.8107
Vehicle	5	39.59	34.58	1.0743	1.0234	1.0463	1.0272
	13	75.76	34.58	1.1212	0.9501	0.9286	0.9569
	21	90.3	34.58	1.4685	0.9069	0.895	0.9633
German	5	46.4	28.06	1.0321	0.9608	0.9797	0.9704
	13	84.3	28.06	1.1525	0.7719	0.788	0.8378
	19	95.3	28.06	1.1586	0.6508	0.729	0.7659
Segment	5	43.46	8.23	1.079	1.0316	1.0024	1.017
	13	79.3	8.23	0.9101	0.8566	0.7995	0.9405
	21	92.2	8.23	2.2819	0.7533	0.7436	0.932

Results – Datasets with Missing Values

	Heartc	Breastw	CRX	Census	Sick	Hepatitis
ECV10(lda) - NA omit	**16.44**	**3.96**	**13.48**	**17.49**	**6.47**	**17.97**
ECV10(knn) - NA omit	**33.31**	**2.81**	**30.41**	**20.85**	**7.97**	**17.12**
ECV10(rpart) -NA omit	**20.14**	**5.35**	**14.04**	**15.91**	**1.79**	**20.2**
ECV10(rpart)	**19.43**	**5.75**	**14.01**	**15.46**	**1.34**	**21.43**
Err_MI LDA ratio	**.9811**	**1.0523**	**1.0682**	**0.9748**	**0.7017**	**0.7045**
Err_MI KNN ratio	**1.018**	**1.1167**	**0.9914**	**0.9684**	**0.7229**	**1.3685**
Err_MI Rpart ratio*	**1.0190**	**1.0469**	**1.0178**	**0.9961**	**1.0895**	**0.4932**
Err_MDI LDA ratio	**0.9700**	**1.0404**	**1.0719**	**0.9742**	**0.7032**	**0.7033**
Err_MDI KNN ratio	**1.0127**	**1.1103**	**1.0065**	**0.9669**	**0.7892**	**1.1574**
Err_MDI Rpart ratio*	**1.0010**	**1.0375**	**1.0121**	**0.9954**	**1.0350**	**0.5412**
Err_KNNI LDA ratio	**0.9981**	**1.0648**	**1.0683**	**0.9742**	**0.7125**	**0.7256**
Err_KNNI KNN ratio	**0.9626**	**1.0747**	**.9858**	**0.9673**	**0.7791**	**1.2371**
Err_KNNI Rpart ratio*	**1.0159**	**1.0005**	**1.0214**	**0.9966**	**0.8583**	**0.5254**

Conclusions

- For datasets with a small amount of missing values little difference is observed between case deletion and other imputation methods.
- When case deletion is used the CV Error variability increases according to the percentage of missing.
- Only small differences can be observed between the results obtained with mean and median imputation.
- The effect of the missing values depend on their distribution on the data matrix and on their location on relevant variables. The percent of instances with missing appears to have a larger influence on the effect of classification accuracy than the percent of missing overall.
- The effect of the treatment of the missing values looks different for the rpart classifier than for the LDA and KNN classifiers

Normalization

- Data normalization consists in scaling the attribute values of the data into an small specified range, such as -1 to 1 or 0 to 1.
  Also it is known as range normalization.
- In microarray data a lot of time is spending doing variance normalization .

Reasons for normalizing

Normalizing the input data will help speed up the learning phase.

Attributes with initially large ranges will outweigh attributes with initially smaller ranges, then dominate the distance measure. For instance, the K-nearest neighbor classifier using a Euclidean distance measure depends on all input dimensions being scaled equally.

Some kind of data normalization also may be necessary to avoid numerical problems such as precision loss from arithmetic overflows.

The Bupa dataset

Number of instances: 345

Number of attributes: 7

Attribute information:

1. Mcv mean corpuscular volume
2. alkphos alkaline phosphotase
3. sgpt alamine aminotransferase
4. sgot aspartate aminotransferase
5. gammagt gamma-glutamyl transpeptidase
6. drinks number of half-pint equivalents of alcoholic beverages drunk per day
7. Class: 1 (liver disorder) and 2 (no liver disorder)

```
> bupa[1:20,]
   V1  V2 V3 V4 V5  V6 V7
1  85  92 45 27 31 0.0  1
2  85  64 59 32 23 0.0  2
3  86  54 33 16 54 0.0  2
4  91  78 34 24 36 0.0  2
5  87  70 12 28 10 0.0  2
6  98  55 13 17 17 0.0  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
```

Z-score normalization

The V values are normalized based on the mean and standard deviation.

V' = (V - mean) / std

This method works well in cases when you do not know the actual minimum and maximum of your input data or when you have outliers that have great effect on the range of the data.

- >zbupa=znorm(bupa)

Min-Max normalization

It performs a linear transformation on the original data V into the specified interval [newmin,newmax]

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

The advantage of this method is that it preserves all relationships of the data values exactly. It does not introduce any potential bias into the data. The disadvantage is that it will encounter an "out of bounds" error if a future input case falls outside of the original data range.

- > mmbupa=mmnorm(bupa)

Normalization by decimal scaling

It normalizes by moving the decimal point of values. The number of decimal points moved depends on the maximum absolute value.

V' = V / 10^j where j is the smallest integer such that Max(| V' |) < 1.

Only useful when attributes values are greater than 1 in absolute value

- > dsbupa=decscale(bupa)

Sigmoidal normalization

It transforms the input data nonlinearly into the range -1 to 1, using a sigmoid function.

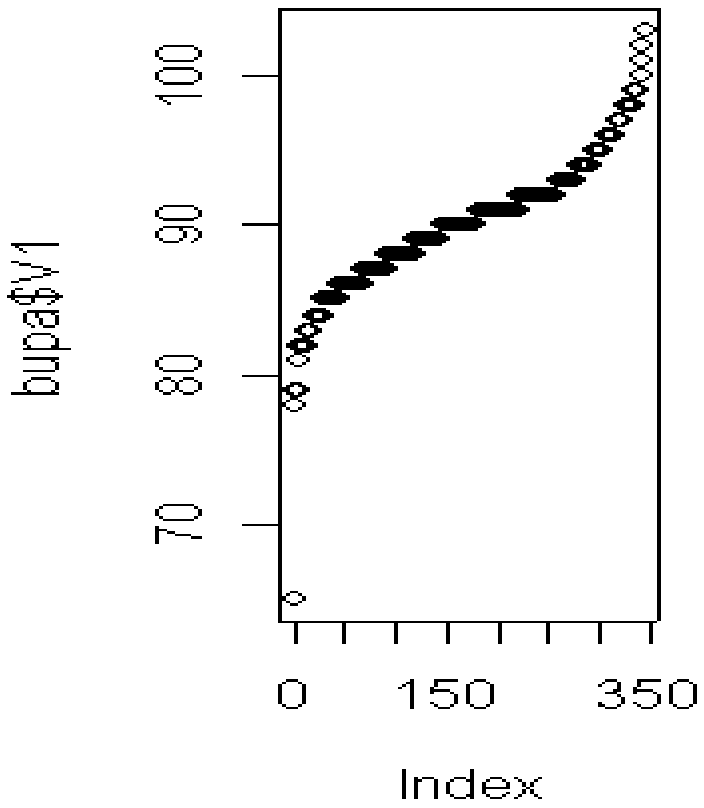
V' = (1 - e^(-a)) / (1 + e^(-a)) where a = (V - mean) / std

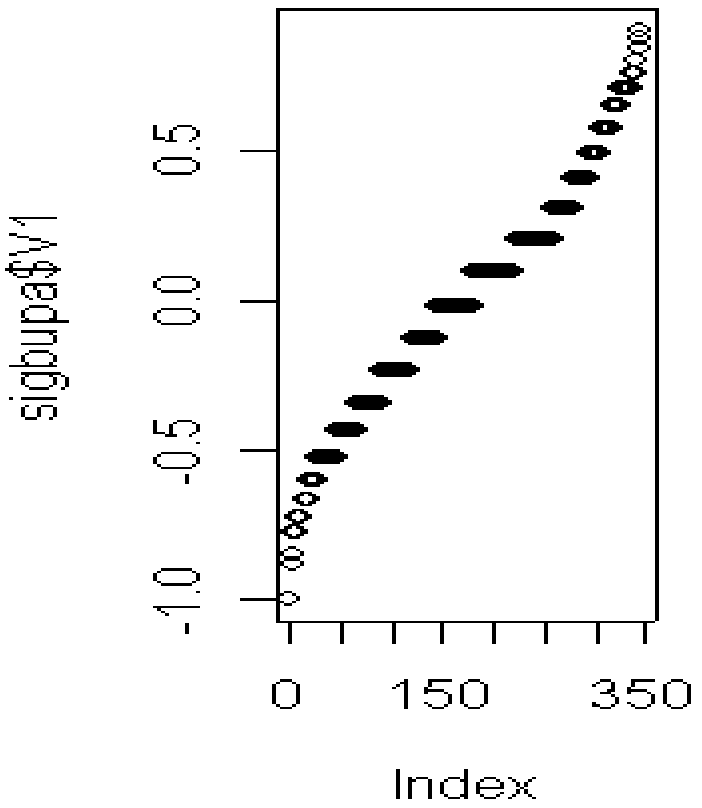
Data points within a standard deviation of the mean are mapped to the almost linear region of the sigmoid. Outlier points are compressed along the tails of the sigmoidal function.

Sigmoidal mormalization is especially appropriate when you have outlier data points that you wish to include in the data set. It prevents the most commonly occurring values from being compressed into essentially the same values without losing the ability to represent very large outlier values.

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

Visualizing the effect of the sigmoidal transformation





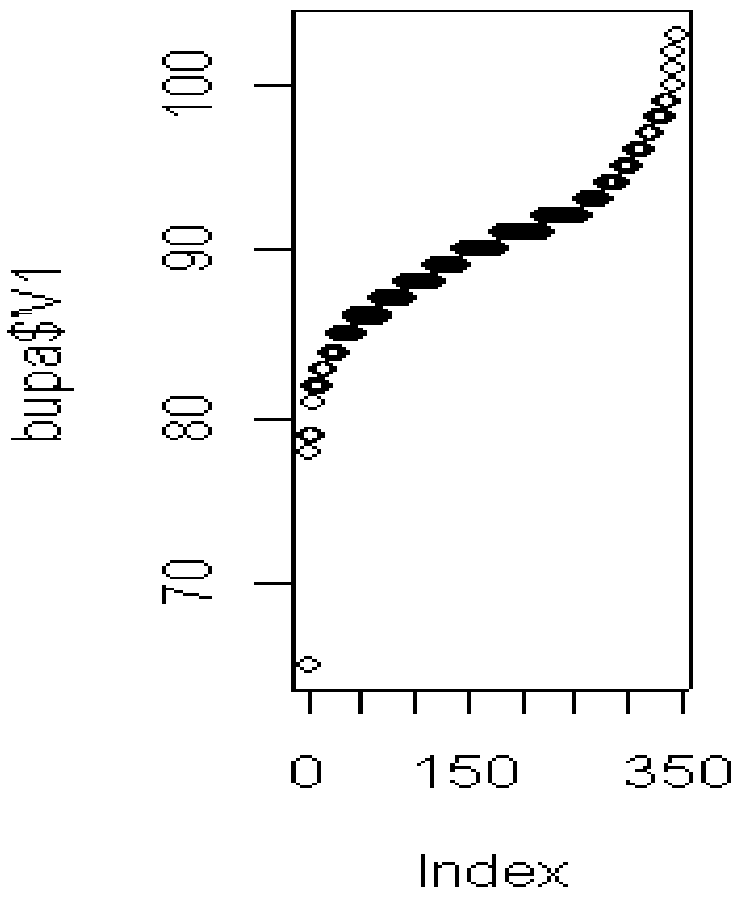
Softmax Normalization

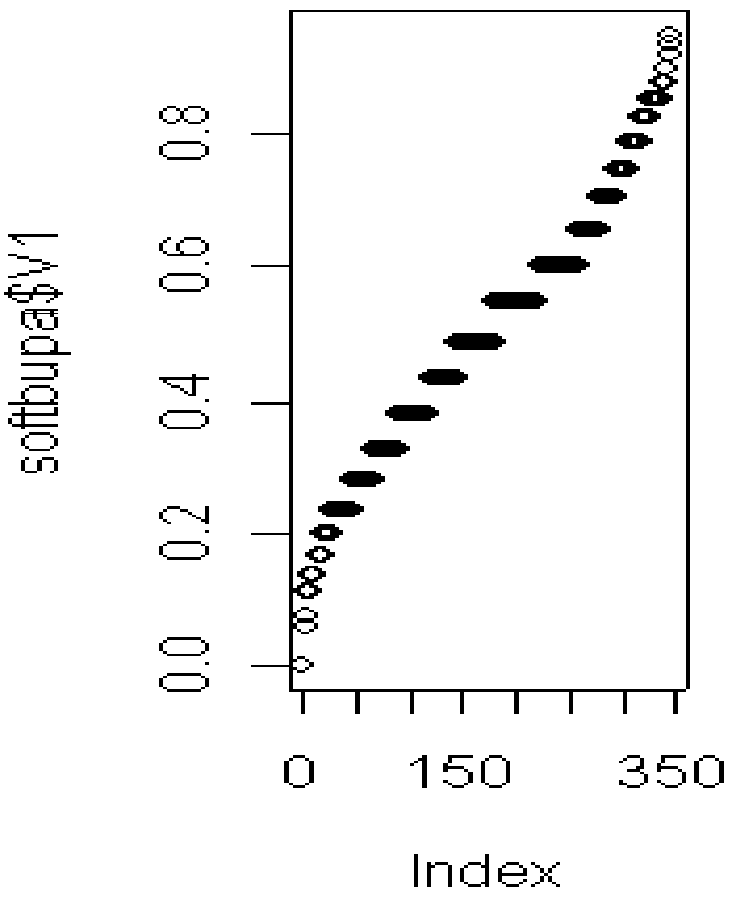
It is so called because it reaches "softly" toward its maximum and minimum value, never quite getting there. The transformation is more or less linear in the middle range, and has a smooth nonlinearity at both ends. The whole output range covered is 0 to 1 and the transformation assures that no present value lies outside this range.

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

where a= (V - mean) / std

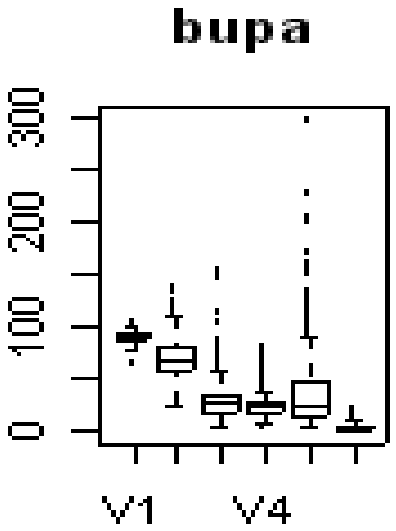
- softbupa=softmaxnorm(bupa)
- zbupa=rangenorm(bupa,method="znorm")

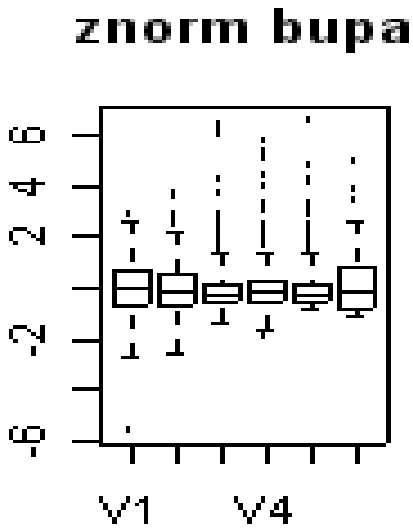

bupa$V1
70
80
90
100
0
150
350
Index

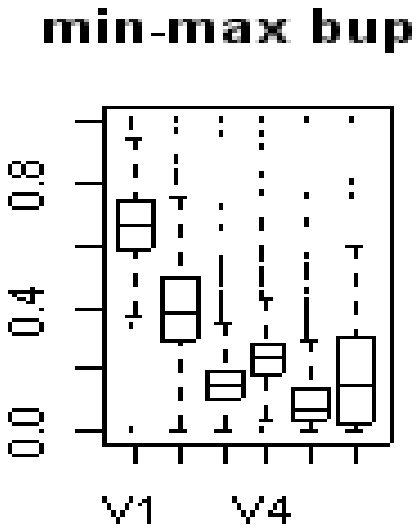

softbupa$V1
0.0
0.2
0.4
0.6
0.8
0
150
350
Index

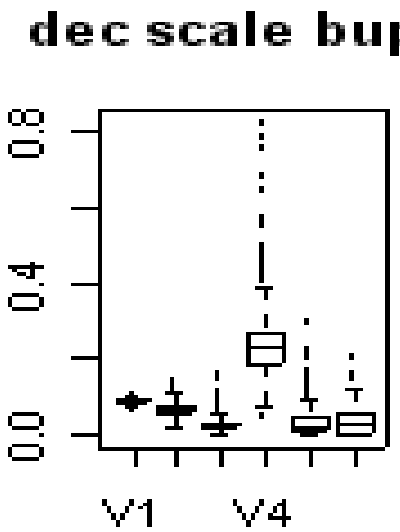
Boxplots to show the effect of the normalization

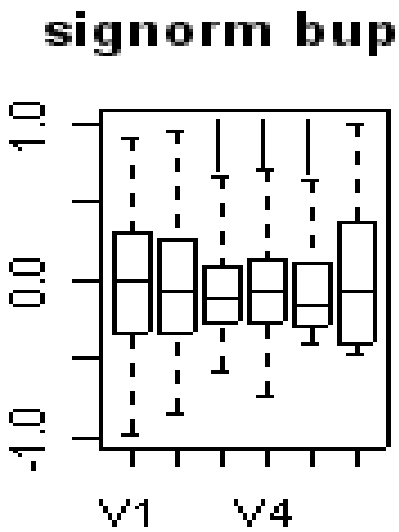
```
> par(mfrow=c(2,3))
> boxplot(bupa[,1:6],main="bupa")
> boxplot(zbupa[,1:6],main="znorm bupa")
> boxplot(mmbupa[,1:6],main="min-max bupa")
> boxplot(dsbupa[,1:6],main="dec scale bupa")
> boxplot(sigbupa[,1:6],main="signorm bupa")
> boxplot(softbupa[,1:6],main="softmax bupa")
```

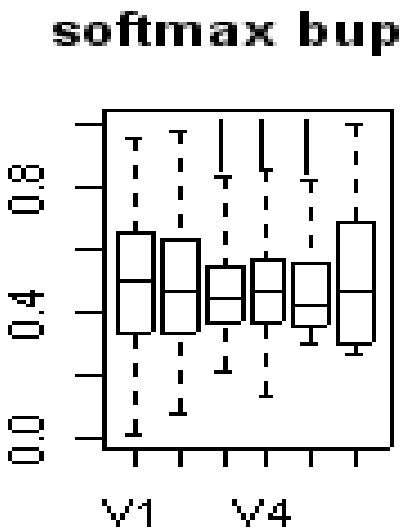

bupa
0
100
200
300
V1
V4


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

Normalization in Weka

Weka performs normalization for numerical attributes transforming the original values to the interval [0,1].
After opening an arff file follow the sequence filters>unsupervised>attributes>Normalize.
Z-normalization is carried out by the filter Standardize