CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

APLICAÇÃO DE RECONHECIMENTO DE PADRÕES EM UM EXPERIMENTO LINGUÍSTICO

APPLICATION OF PATTERN RECOGNITION IN A LINGUISTIC EXPERIMENT

Aluno: Ali Kamel Issmael Junior
Banca Examinadora:
Presidente, Professora Dra. Aline Gesualdi Manhães (CEFET/RJ) (orientador)
Dr. José Vicente Calvano (MARINHA DO BRASIL) (Coorientador)
Professora Dra. Marije Soto (UERJ)
Professor Dr. Thiago de Moura Prego (CEFET/RJ)
Professora Dra. Luciana Faletti Almeida (CEFET/RJ) - SUPLENTE

SUMÁRIO

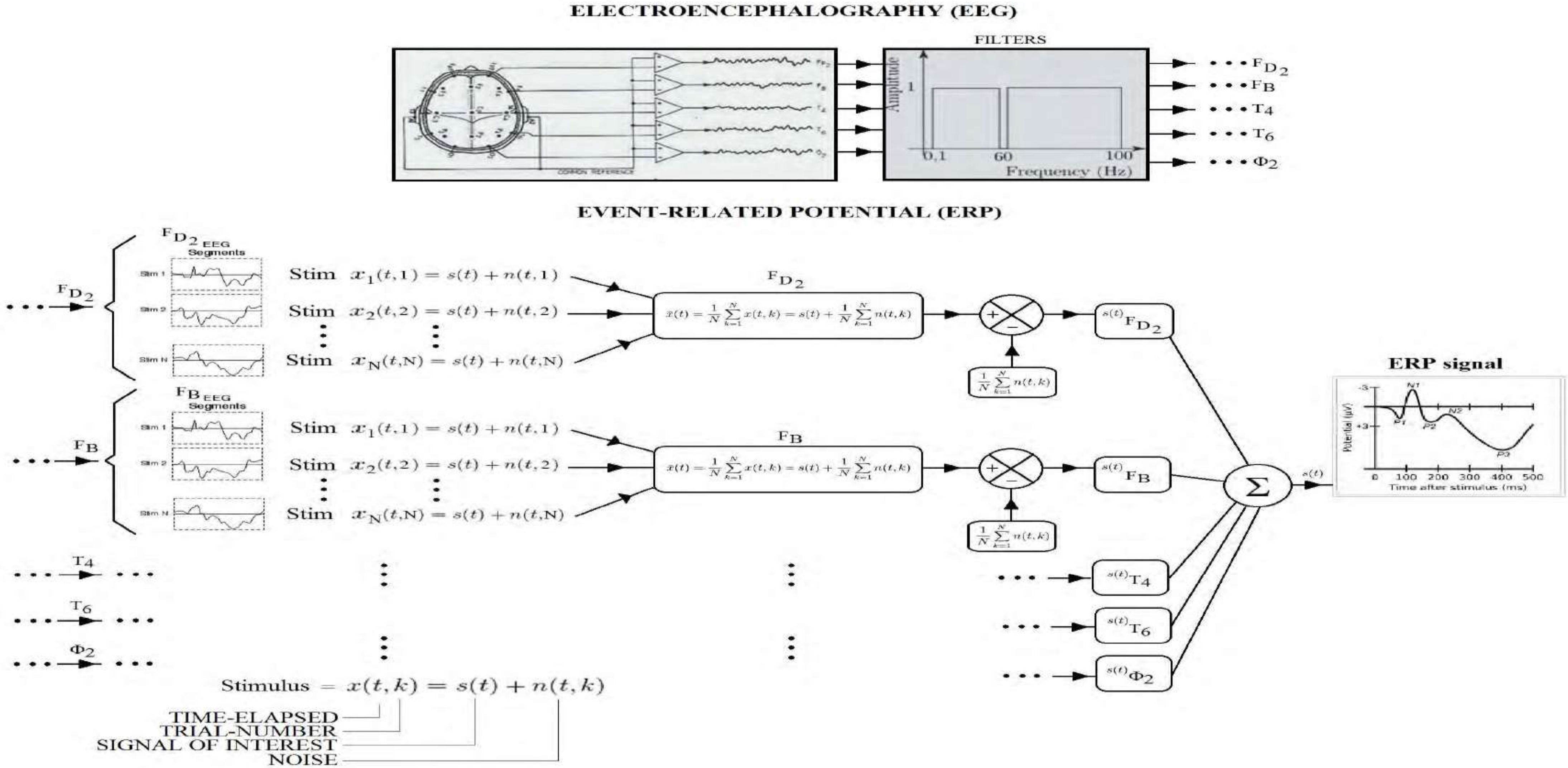
1. Introduction

The "Event-Related Potentials" (ERP) technique consists of the measurement of electrical biological signals obtained by electroencephalography (EEG), which are direct results of stimuli to sensory, cognitive or motor events. In this way, the ERP technique allows the non-invasive analysis of the brain functioning.

Based on the results of computational stimuli for words and sentences, obtained by Soto (2014), the treatment of these data and the extraction of ERP parameters, using the EEGLAB® and ERPLAB® tools, based on the Matlab® simulation program, and the clustering analysis of the obtained parameters, the result of the research is the obtaining of supervised and unsupervised pattern recognition algorithms, for the classes proposed for the mentioned experiment, and the comparative study and discussion of the classification results found, using the Webb (2002) methodology.

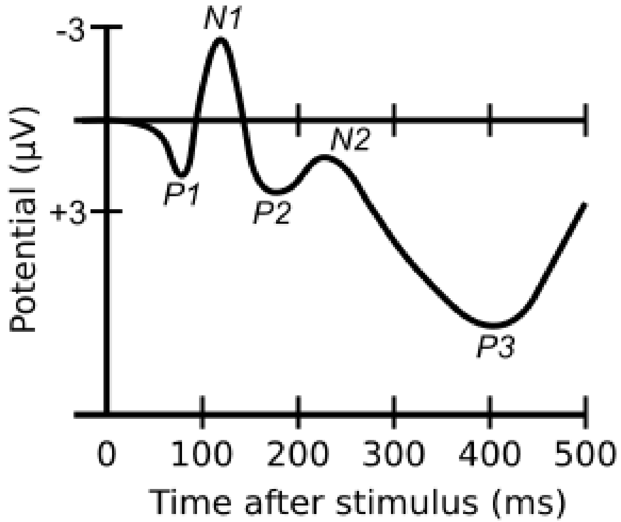
For this work, the proposed question is: if applying the pattern recognition methodology proposed by Webb (2002) in the ERP results from the Soto (2014) data experiment, is it possible to obtain good classification paradigms considering each type of stimulus for the epochs previously labeled (using supervised classification methods) and not labeled (unsupervised classification and clustering methods)?

2. Theoretical References - EEG & ERP

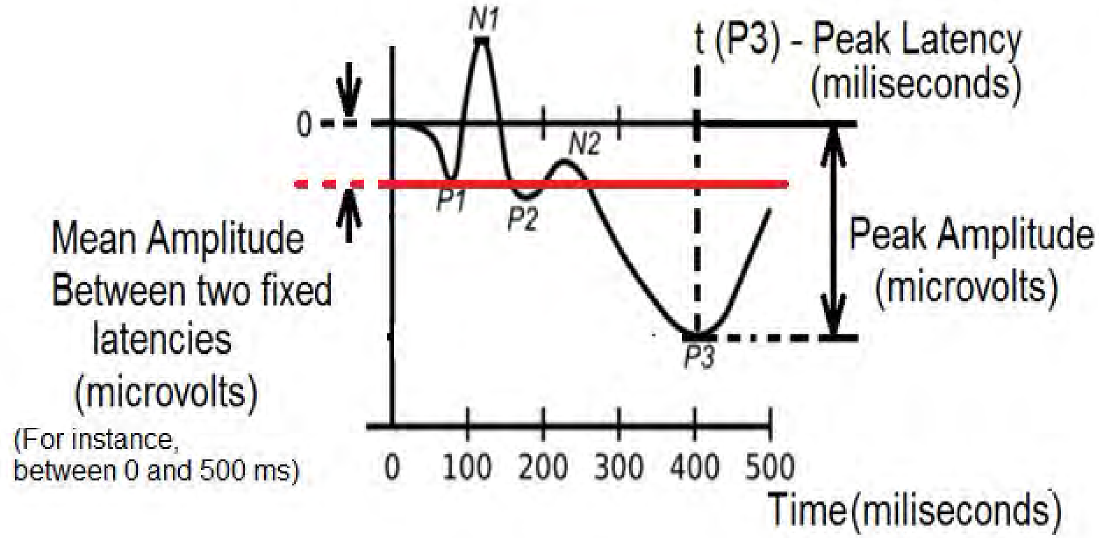


Simplified schematics for ERP experiment (GESUALDI, 2011)

2. Theoretical References - EEG & ERP

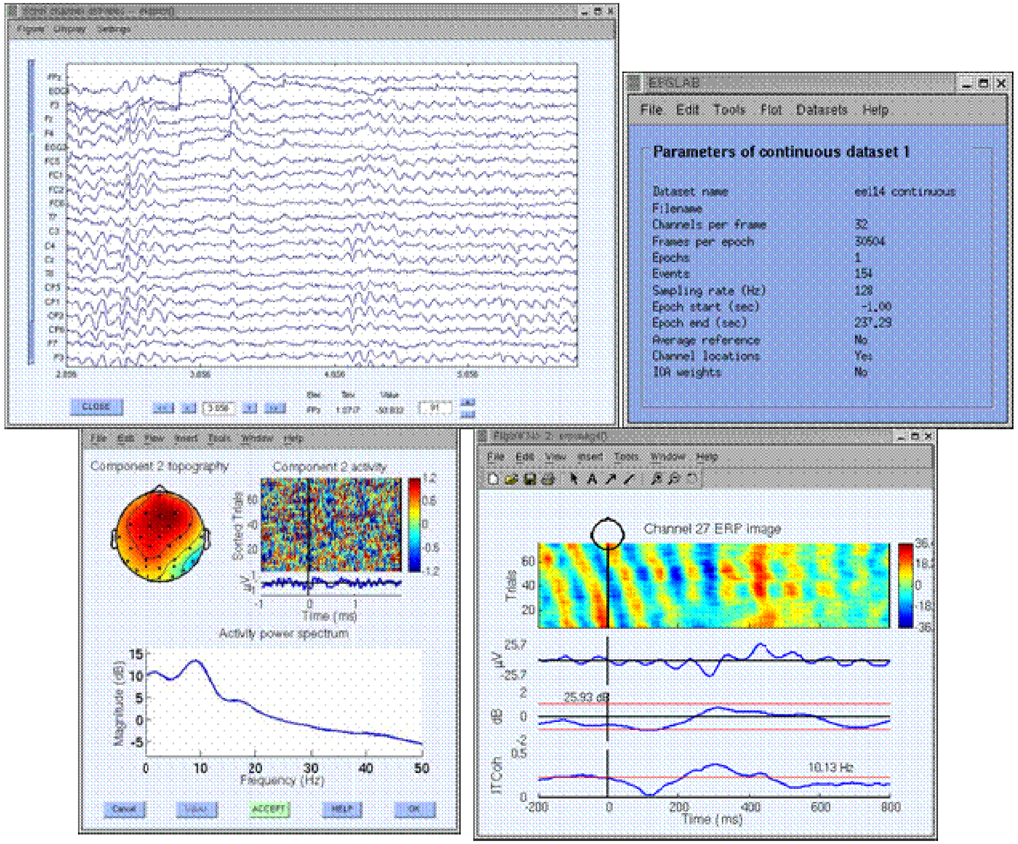


A waveform showing several ERP components as P1 (P100), N1(N100), P2(P200), N2 (N200) and P3 (P300) (WIKIPEDIA, consulted on April, 15th, 2016)



ERP parameters extracted from the ERP waveform (ISSMAEL JUNIOR, A.K. adapted from Wikipedia (2016))

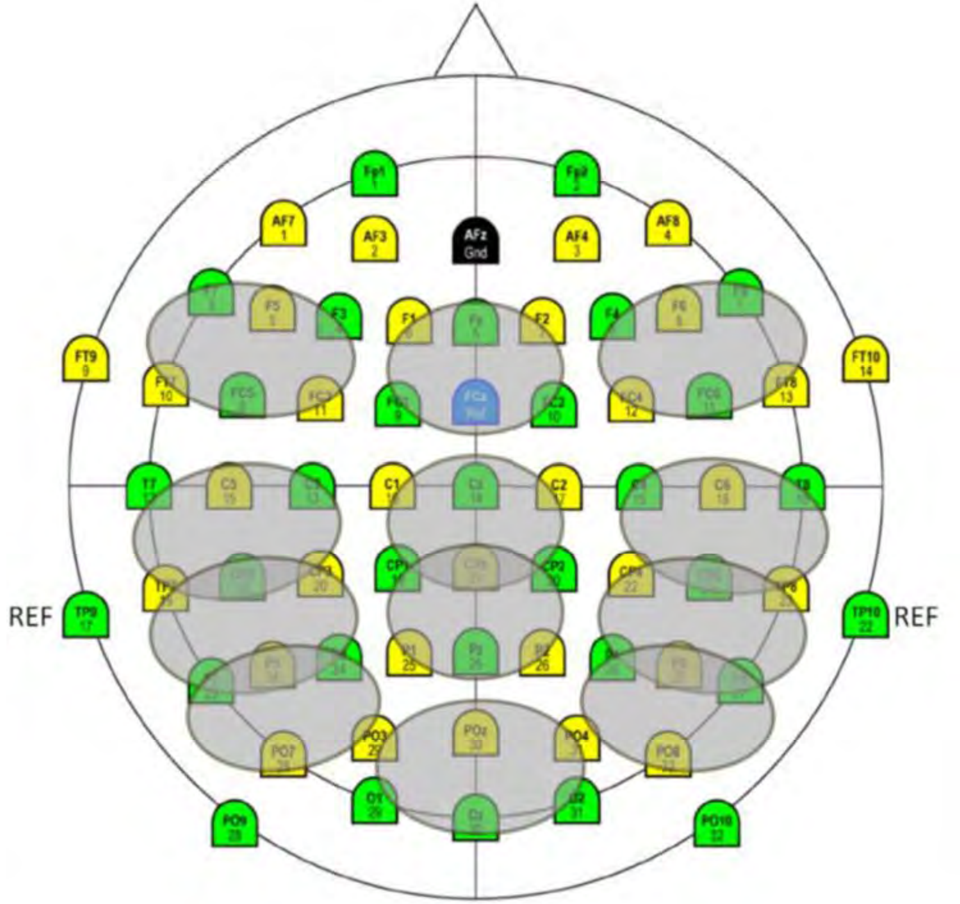
2. Theoretical References - EEG/ERP Data Software toolboxes and Matlab® platform



Example of Graphical Interfaces of the EEGLAB® and ERPLAB® toolboxes
(EEGLAB® Tutorial site, consulted on April, 15th, 2016)

2. Theoretical References - Soto (2014) Experiment

Electrode set up during recording (SOTO,2014)



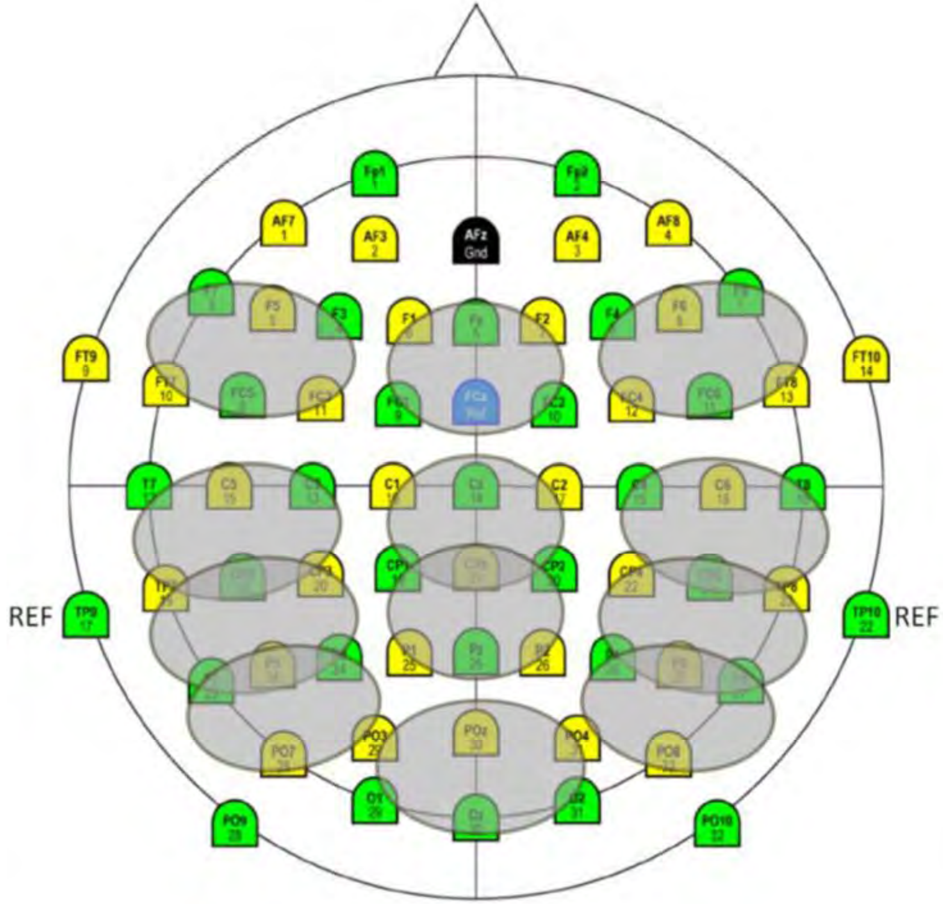
ROI definition as based on anatomical proximity (SOTO,2014)

2. Theoretical References - Soto (2014) Experiment

The ROIs along the mid-line were:
Frontal (F1, F2, FC1, FC2, FCz and Fz);
Central (C1, C2, CP1, CP2, CPz and Cz),
Parietal (CP1, CP2, CPz, P1, P2, and Pz), and
Occipital (O1, O2, Oz, PO3, PO4, and POz).

On the left hemisphere, they were:
Frontal (F3, F5, F7, FC3, FC5 and FT7);
Central (C3, C5, CP3, CP5, T7 and TP7),
Parietal (CP3, CP5, P3, P5, P7 and TP7), and
Occipital (P3, P5, P7, PO3 and PO7).

And on the right hemisphere, they were:
Frontal (F4, F6, F8, FC4, FC6 and FT8);
Central (C4, C6, CP4, CP6, T8 and TP8),
Parietal (CP4, CP6, P4, P6, P8 and TP8), and
Occipital (P4, P6, P8, PO4 and PO8).



ROI definition as based on anatomical proximity (SOTO,2014)

2. Theoretical References - Soto (2014) Experiment

Sentence Task				
Condition	**context**	**congruence**	**Stimulus example (n=30 for each condition)**	Repeated item
1: CSC	supportive	congruous	Até sem capacete, João dirige ↑ a moto feito louco	dirige a moto
2: CNSC	non-supportive	congruous	Todos os dias, João dirige ↑ a moto feito louco	dirige a moto
3: ISC	supportive	incongruous	Até sem capacete, João dirige ↑ a pera feito louco	dirige -
4: INSC	non-supportive	incongruous	Todos os dias, João dirige ↑ a pera feito louco	dirige -

Word Task				
Condition	relation	Stimulus example (n=30 for each condition) Prime	Target	Repeated item
1: SSR	Syntactic and Semantic	CAPACETE	moto	moto
2: ASR	Associative Semantic	ÔNIBUS	moto	moto
Control 1: UR	Unrelated Words	FACA	nuvem	-
Control 2: PW	(Pseudo Word Target)	FILTRO	garufa	-

Abbreviations: congruous supportive-context (CSC); congruous non-supportive context (CNSC); incongruous supportive-context (ISC); incongruous non-supportive context (INSC); associative semantic relation (ASR); syntactic and semantic relation (SSR); unrelated pair (UR); pair with pseudo word target (PW)

Experimental conditions and sample stimuli for the ERP experiment (SOTO,2014)

2. Theoretical References - Soto (2014) Experiment

Presentation protocol ERP Experiment: sentence task

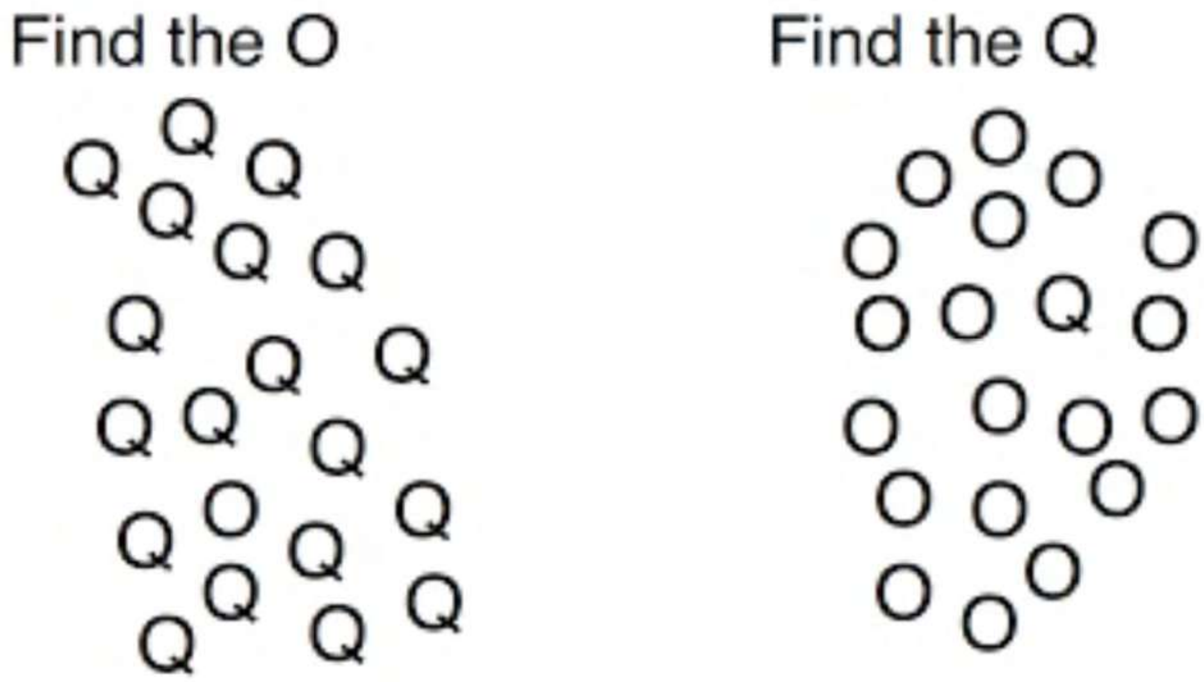
Presented:	+	Até sem capacete,	(blank)	João	(...)	a moto	(blank)	feito louco	(blank)	RESPONDA
Action:						*Target*				*Congruent Y/N?*
Timing: (ms)	1500	300	100	250	(...)	250	100	250	350	1500

Presentation protocol ERP Experiment: word task

Presented:	+	(blank)	CAPACETE	(blank)	moto	(blank)	muito veloz	(blank)	RESPONDA
Action:			*Prime*		*Target*				*Lexical Decision Y/N*
Timing: (ms)	1500	100	250	100	250	100	250	350	1500

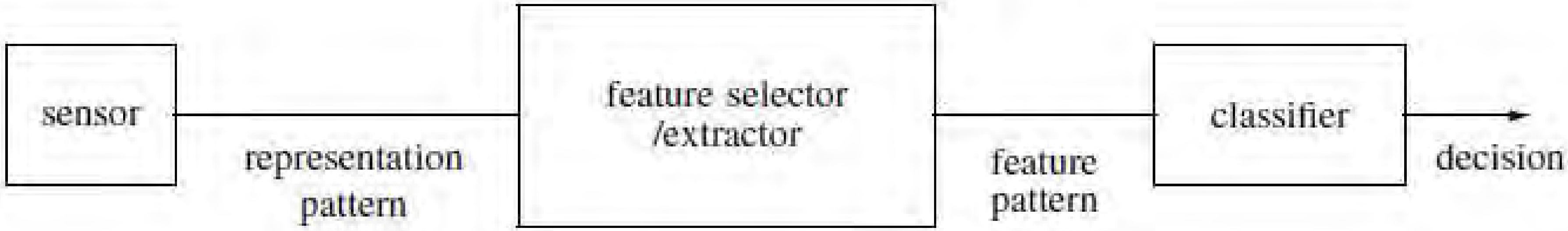
Presentation protocol and SOA for the ERP Experiment
(SOTO,2014)

2. Theoretical References - Pattern Recognition Theory



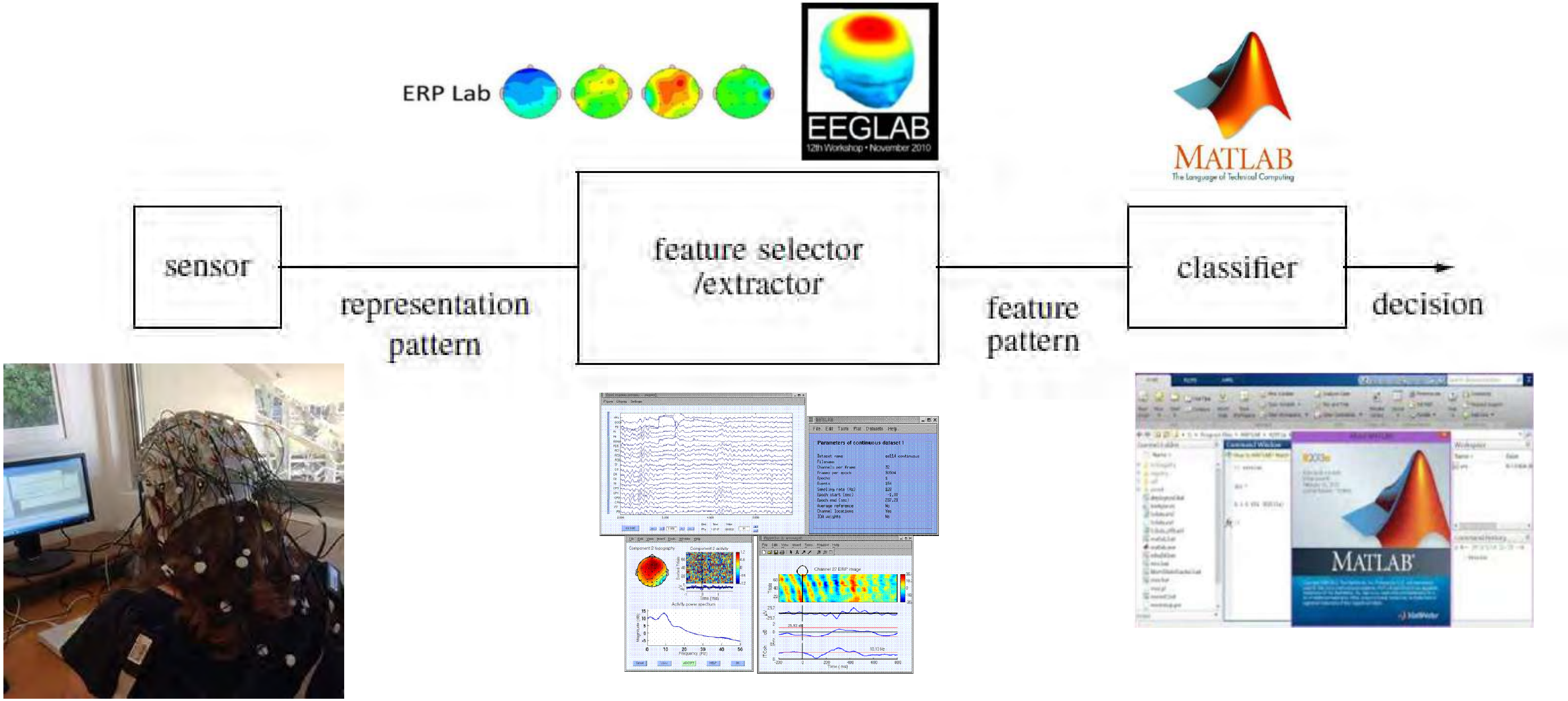
O & Q classification

2. Theoretical References - Pattern Recognition Theory



Pattern Recognition Method (WEBB, 2002)

2. Theoretical References - Pattern Recognition Theory



Pattern Recognition Method (WEBB, 2002)

2. Theoretical References - Pattern Recognition Theory

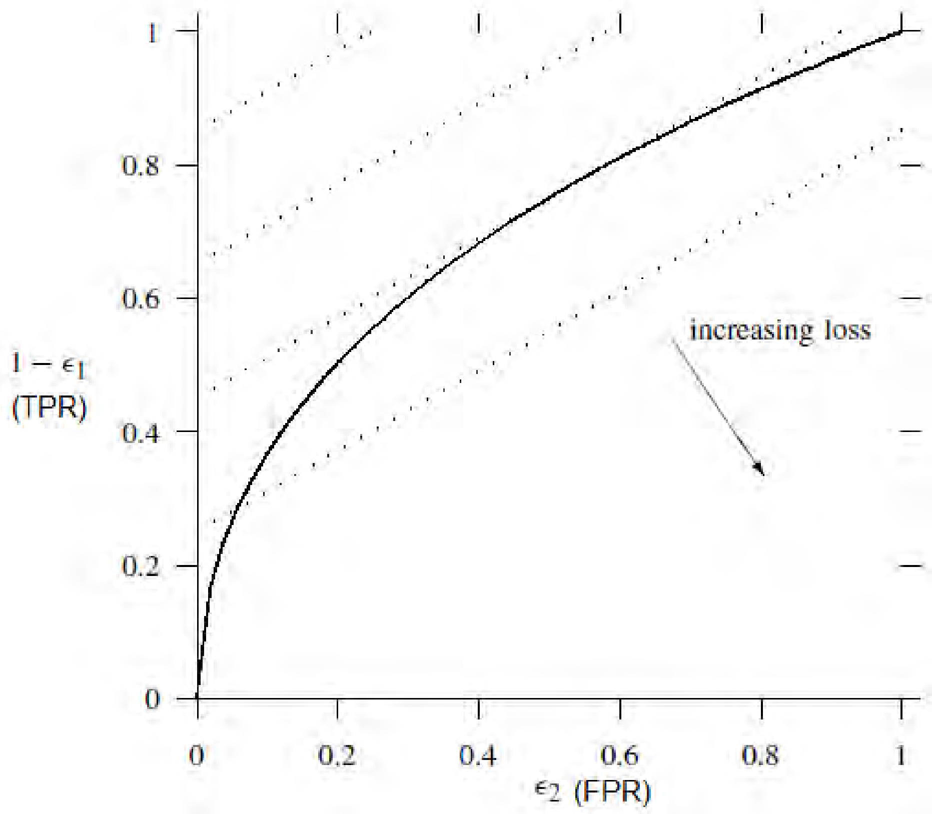
Performance of Classifiers

		Predicted	
		Negative	Positive
Actual	Negative	**a**	**b**
	Positive	**c**	**d**

Example of Confusion Matrix for 2 classes

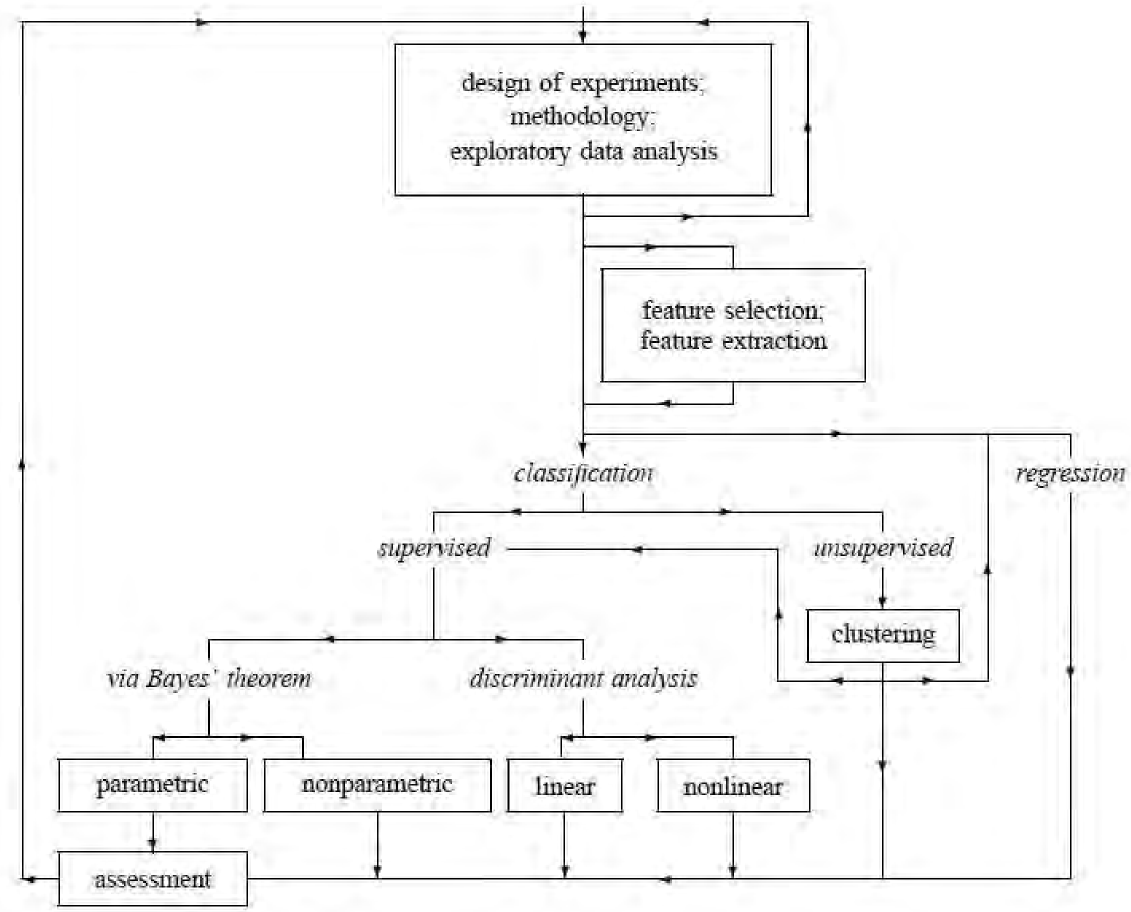
$$AC = \frac{a+d}{a+b+c+d}$$

Accuracy



ROC curve with selected loss contours (straight lines) superimposed (WEBB, 2002)

3. Methodology, Results and Discussion



Pattern Recognition methodology (WEBB, 2002)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering
- k-means
- Gaussian Mixture Models

Apply discrimination (Supervised Classification)

- Naïve Bayes
- Multiclass Support Vector Machine (SVM)
- Neural Network
- Random Forest

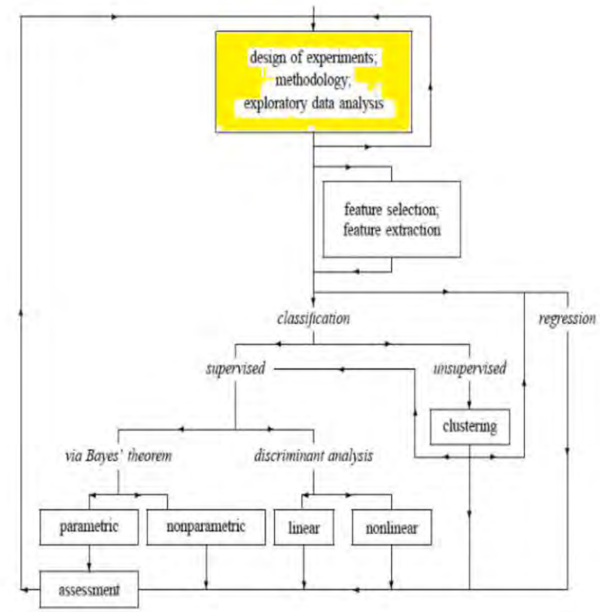
3. Methodology, Results and Discussion

1. Formulation of the problem, Data collection and Initial examination of the data;

For this work, the experimental question is: if applying the pattern recognition methodology proposed by Webb (2002) in the ERP results from the Soto (2014) data experiment, is it possible to obtain good classification paradigms considering each type of stimulus for the epochs previously labeled (using supervised classification methods) and not labeled (unsupervised classification and clustering methods)?

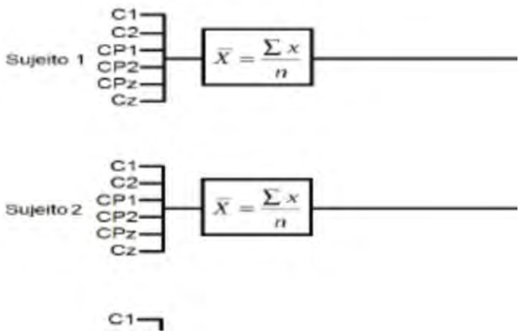
Considering the visual inspection of the ERP segmented signals for the 21 original subjects done by Soto (2014) it were eliminated 7 subjects because the low quality of these signals. Using the sequential order of the experiment, the subjects 2, 3, 4, 5, 6, 7, 9, 10, 13, 15, 16, 17, 18, 19, 20 and 21 are being used in this work. For each subject, from their EEG raw data, it is necessary to create a specific ERPLAB® dataset to organize this data in order to allow their treatment and analysis by MATLAB®.

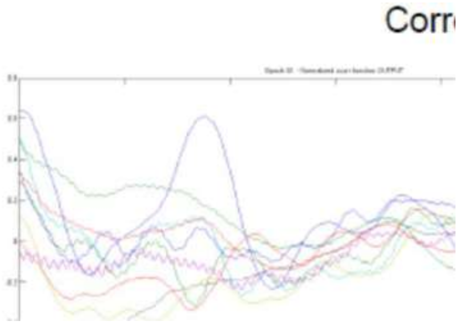
It was used the version v13.6.5b for EEGLAB® and the version v5.0.0.0 for ERPLAB®.

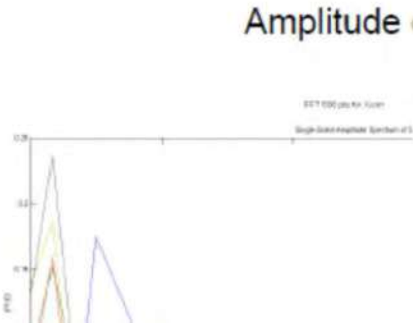


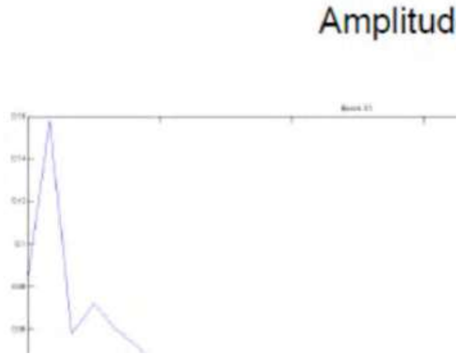
3. Methodology, Results and Discussion

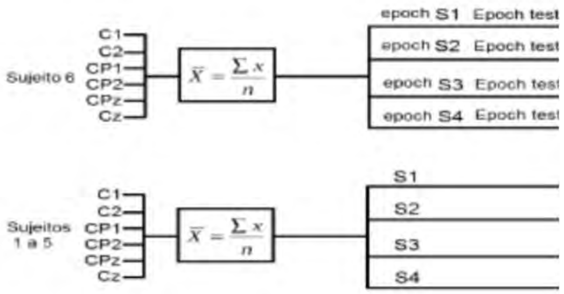
First Try (presented during the Qualify in May 2016) - There were used time x frequency signal processing analysis as spectrogram, FFT and correlation, but the results are not good for Words Task.

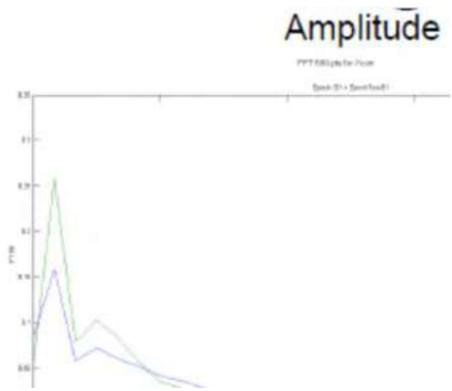












Confusion Matrix for the Training

Class	S1	S2
S1	6	0
S2	3	0
S3	2	0
S4	4	1

Acc = 43,3%

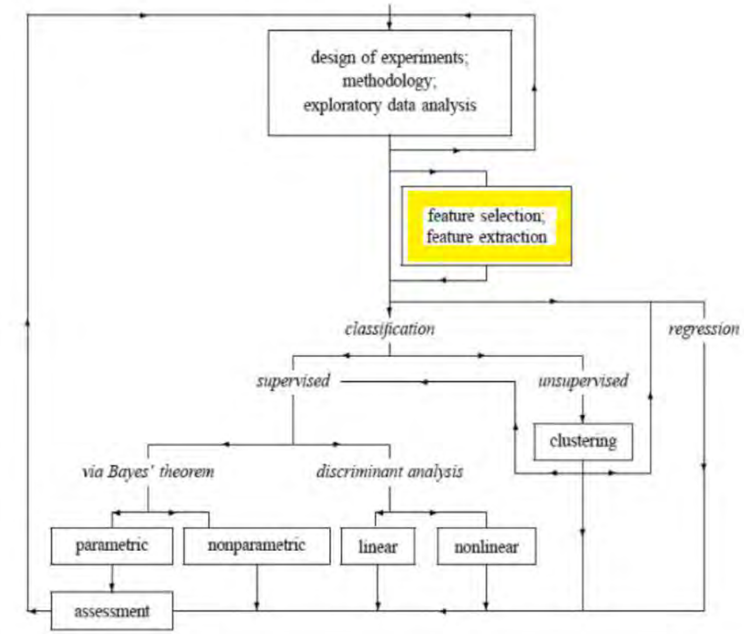
Confusion Matrix for the test

Class	S1	S2
S1	7	2

Acc = 23,14%

3. Methodology, Results and Discussion

2. Feature selection or feature extraction;



After the fist try, the Patern Recognition approach using Neuro linguistic parameters was implemented.

Features:

Mean Amplitude Between two fixed latencies;
Peak Amplitude;
Peak Latency;
ERP Time Range;
Range of Interest (ROI);
Human subject index related to each measurement.

Classes for the sentences task:

S1 (CSC), S2 (CNSC), S3 (ISC), S4 (INSC) and S5 (Control)

Classes for the words task:

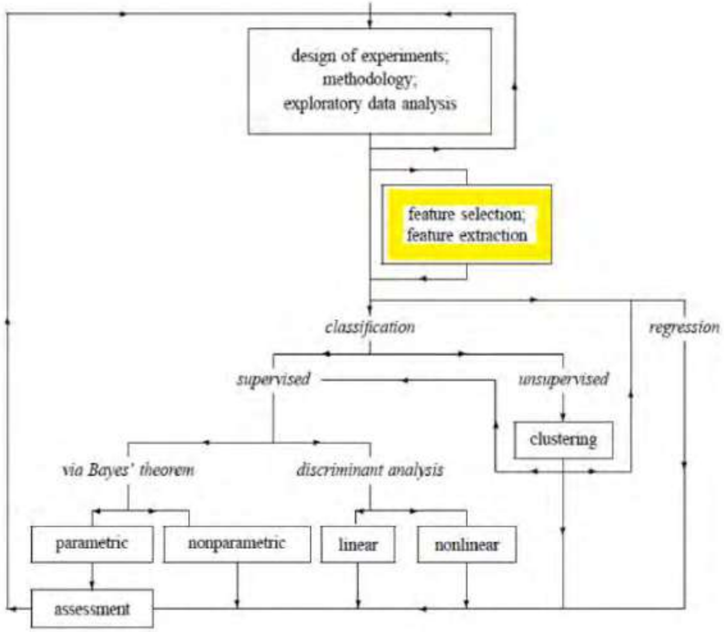
S1 (SSR), S2 (ASR), S3 (Control 1 - UR) and S4 (Control 2 - PW).

3. Methodology, Results and Discussion

2. Feature selection or feature extraction;

Words and Sentences Task Organization and coding for features

Features	Real Value	Code for Matlab® algorithm
ERP Time Range	150-300ms	1
	300-500ms	2
	500-700ms	3
Region of Interest (ROI)	Frontal Mid Line	1
	Central Mid Line	2
	Pariental Mid Line	3
	Occiptal Mid Line	4
	Frontal Left Side	5
	Central Left Side	6
	Pariental Left Side	7
	Occiptal Left Side	8
	Frontal Right Side	9
	Central Right Side	10
	Pariental Right Side	11
	Occiptal Right Side	12
Subject	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	9	9
	10	10
	13	13
	15	15
	16	16
	17	17
	18	18
	19	19
	20	20
	21	21

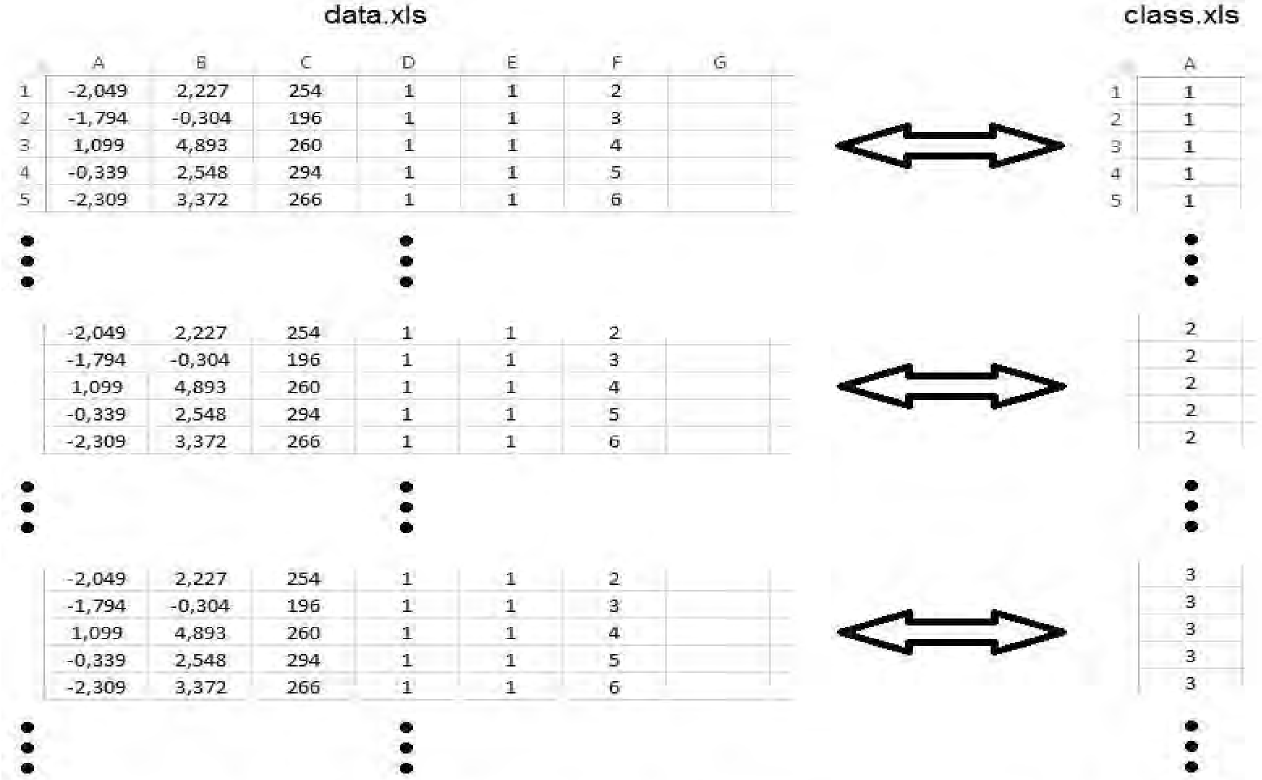


Words and Sentences Task organization and coding for classes

Task	Classes	Coding
Words	S1 (SSR)	1
	S2 (ASR)	2
	S3 (Control 1 – UR)	3
	S4 (Control 2 – PW)	4
Sentences	S1 (CSC)	1
	S2 (CNSC)	2
	S3 (ISC)	3
	S4 (INSC)	4
	S5 (Control)	5

3. Methodology, Results and Discussion

2. Feature selection or feature extraction;



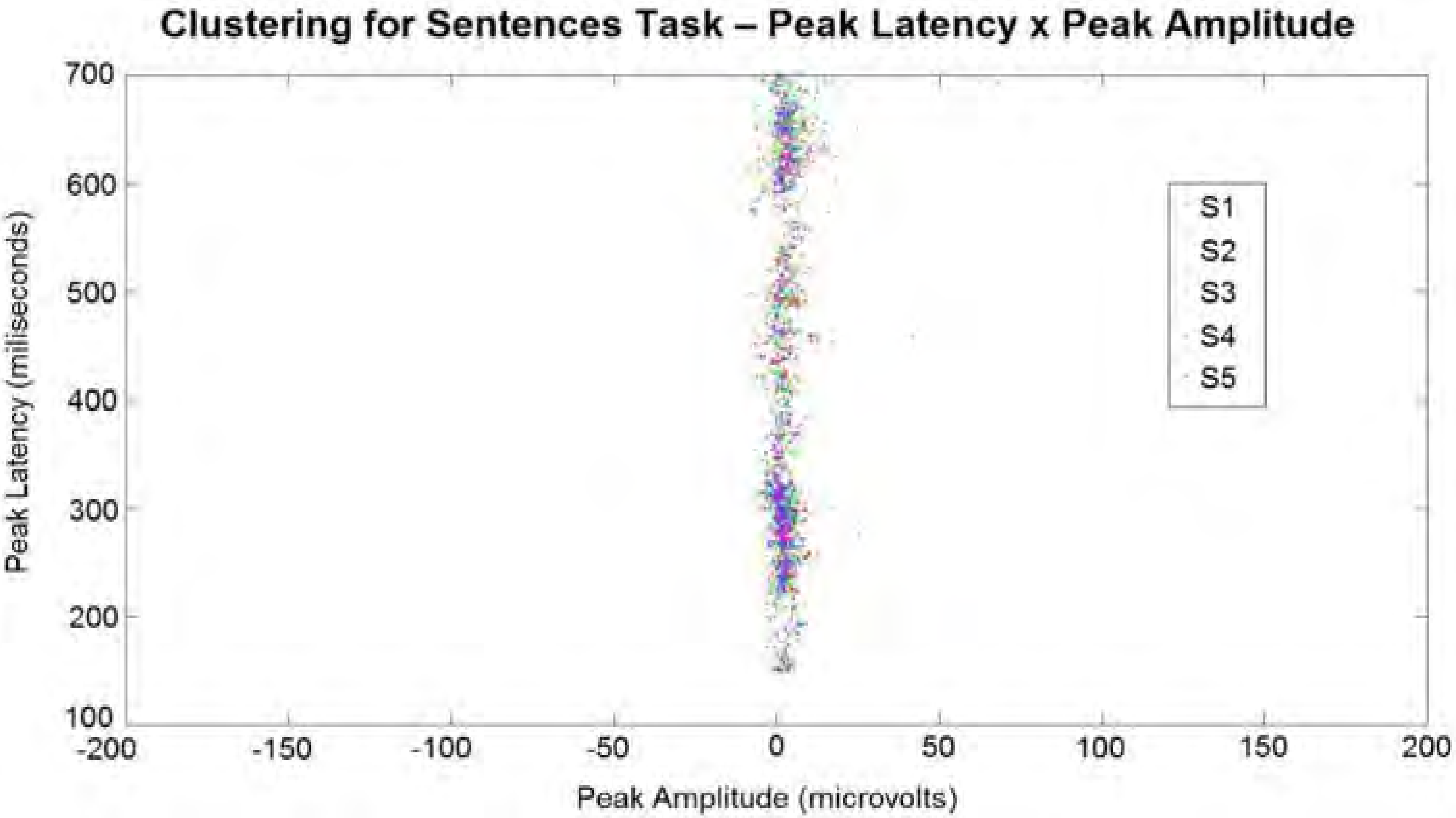
Column A - Mean Amplitude Between two fixed latencies
Column B - Peak Amplitude
Column C - Peak Latency
Column D - ROI
Column E - ERP time range
Column F - Subject

Column A - classes

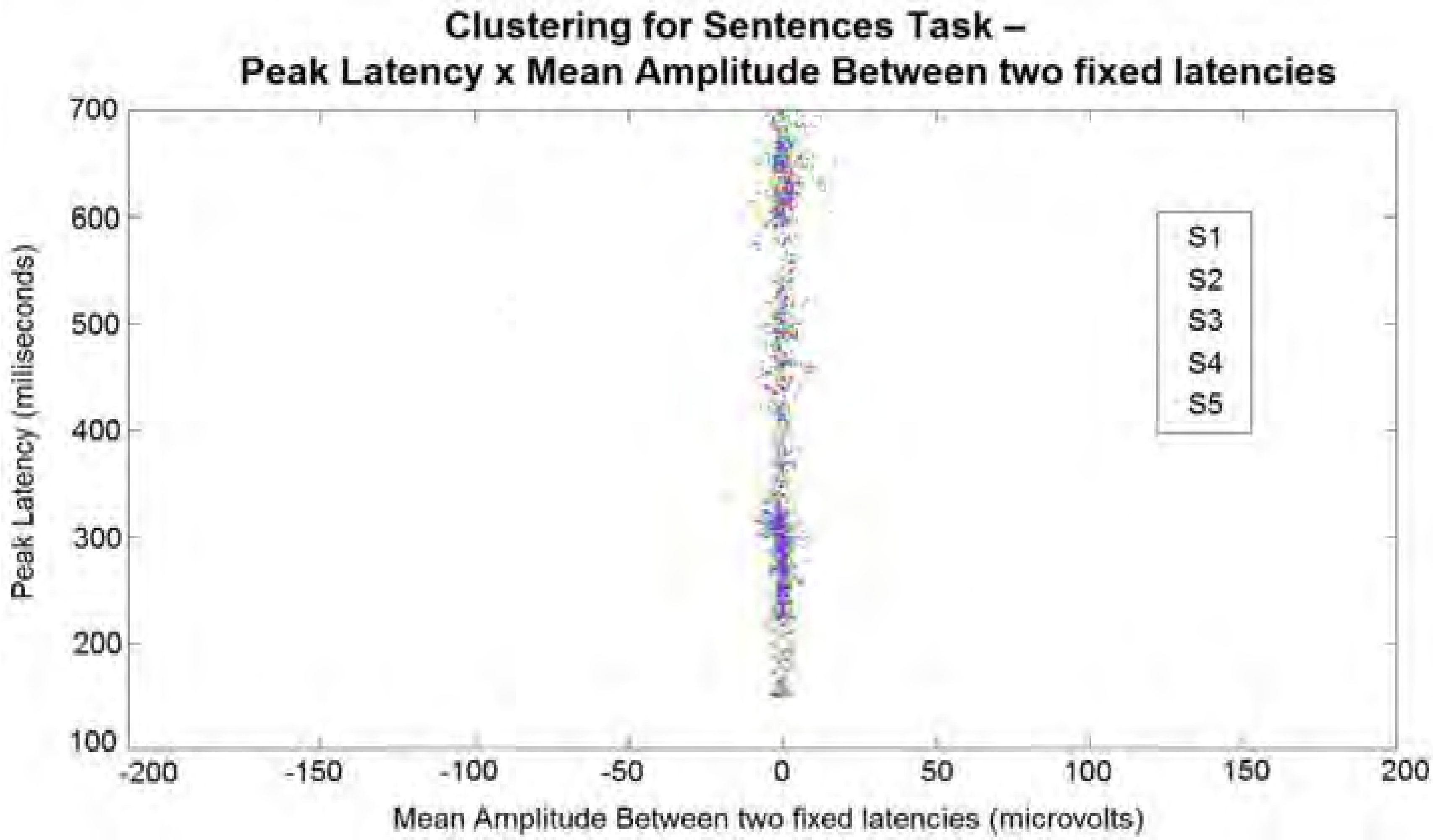
Afterthat, to do the initial examination of the data, it was done the scattering plotting of the features Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency, combined 2-by-2, labelled by classes for each task, to do an initial check of the distribution of the data. The scatter plots will be presented in the Results, for each task

Microsoft Excel® sheets format with the features (data.xls) and classes (class.xls) for both Words and Sentences task with increasing order by classes

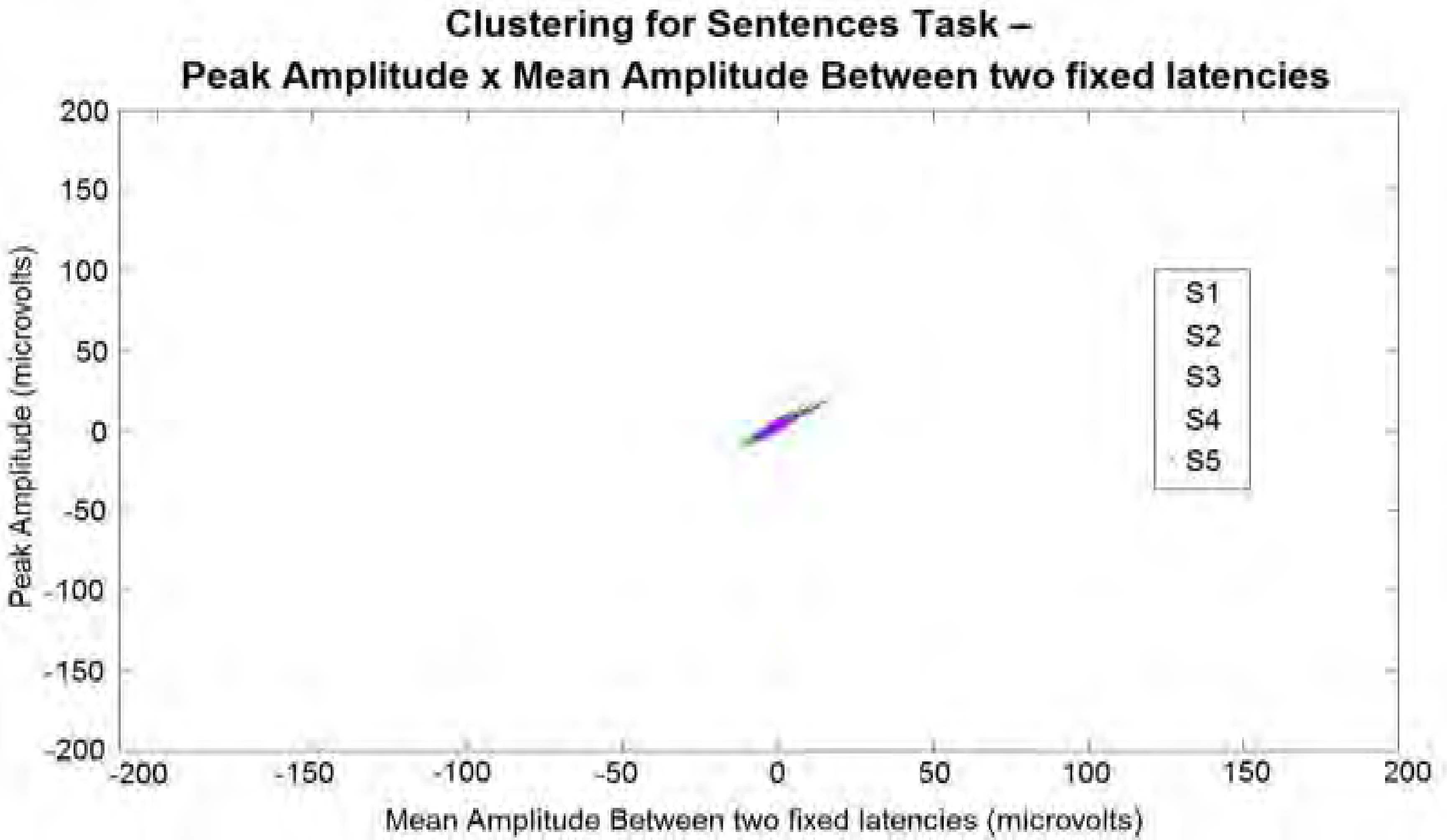
3. Methodology, Results and Discussion



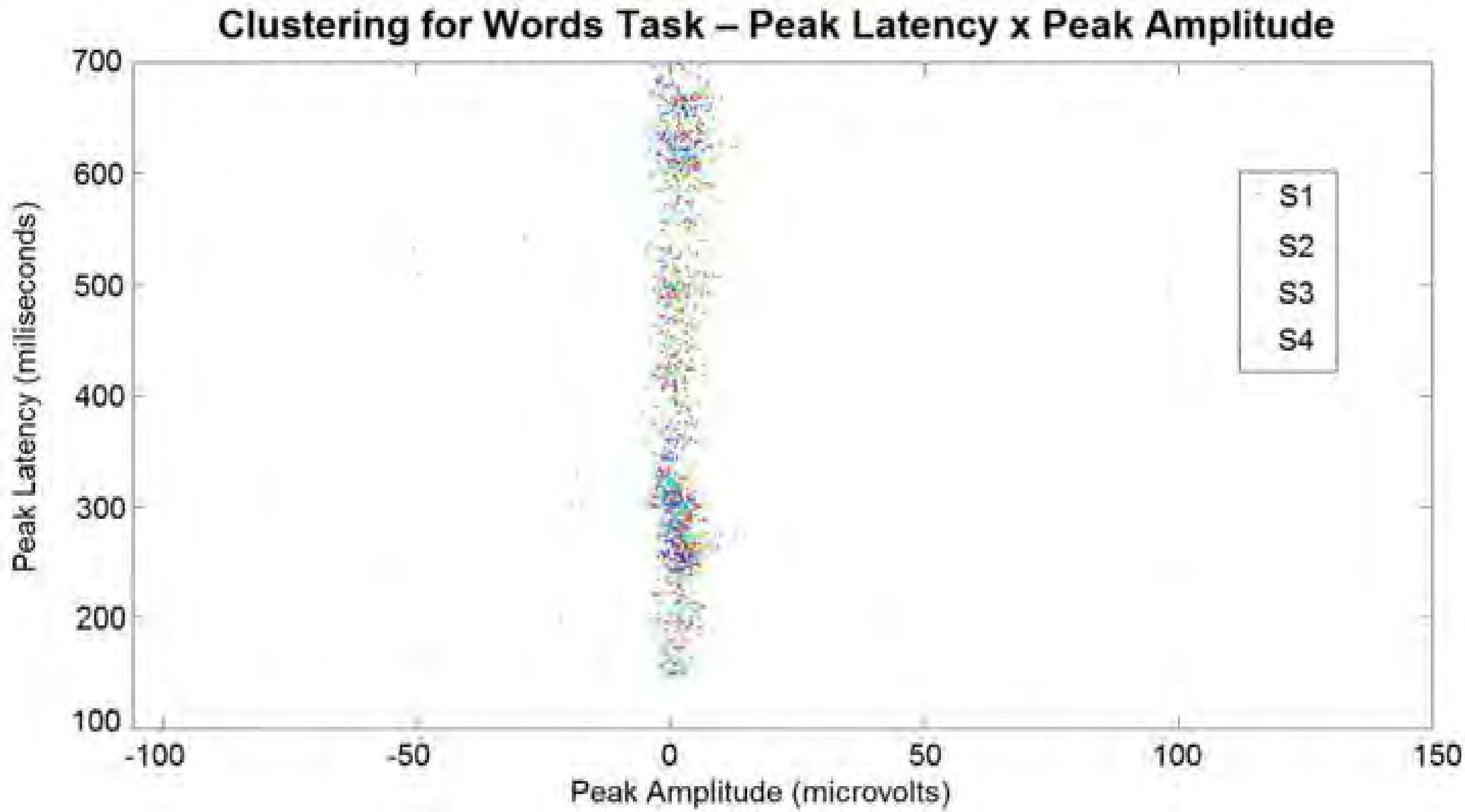
3. Methodology, Results and Discussion



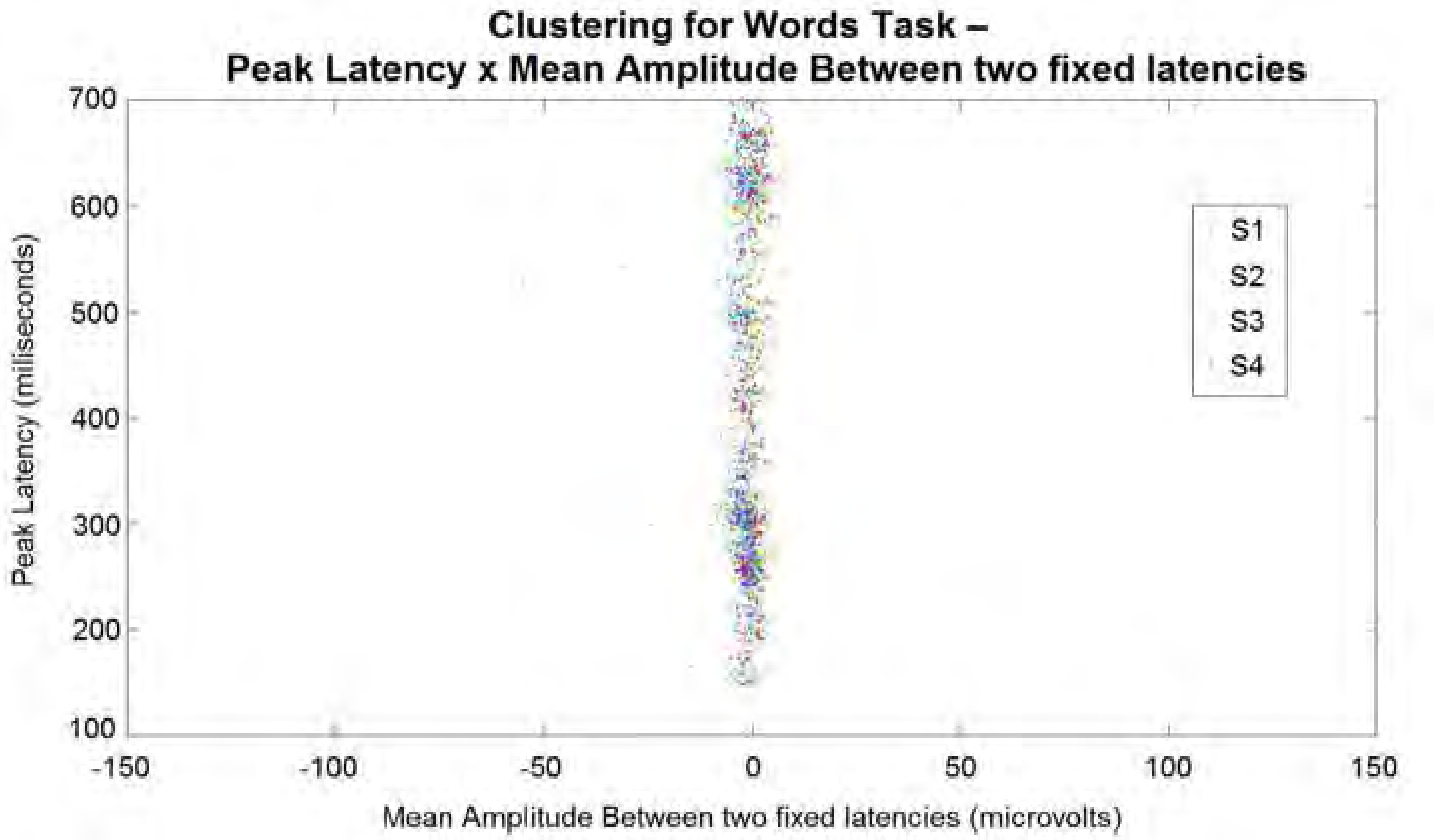
3. Methodology, Results and Discussion



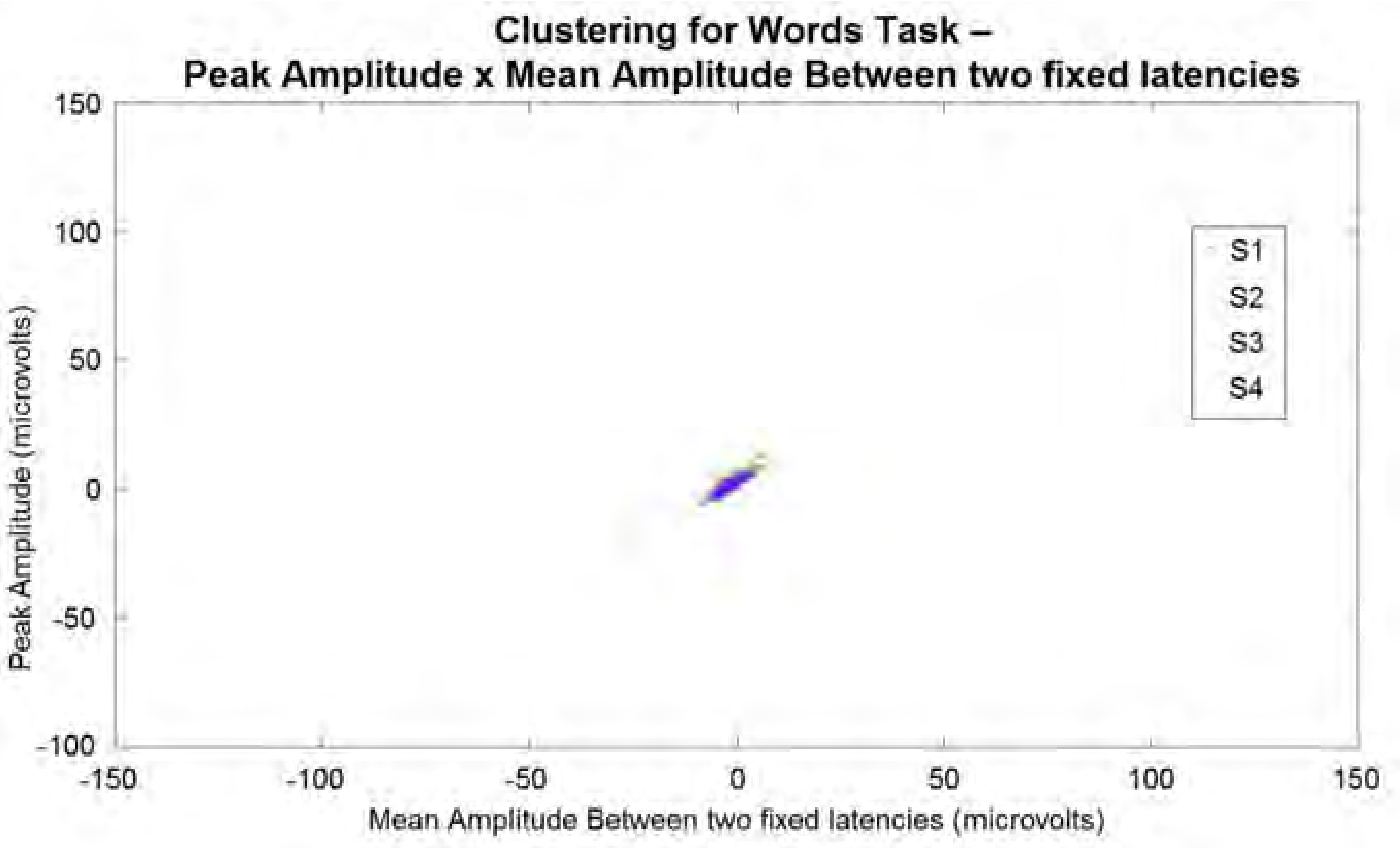
3. Methodology, Results and Discussion



3. Methodology, Results and Discussion



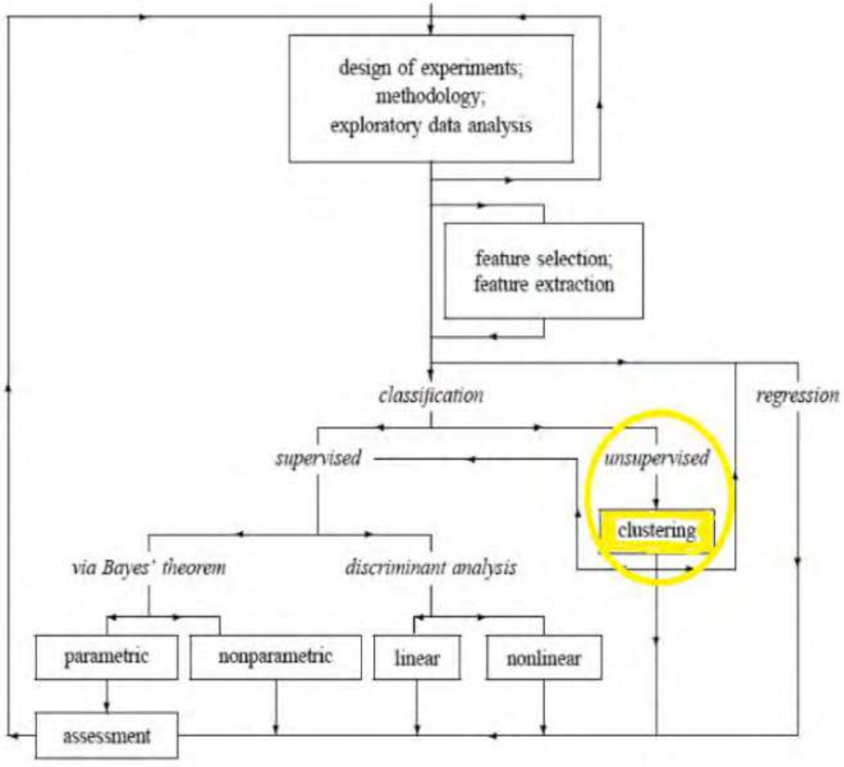
3. Methodology, Results and Discussion



3. Methodology, Results and Discussion

3. Unsupervised pattern classification or clustering;

Concerning the dataset split for the test campaign, due the to the time available, this study consider, for the unsupervised classification and clustering, all the data is treated in one single test set for the classifiers done, not being divided in subsets.



3. Methodology, Results and Discussion

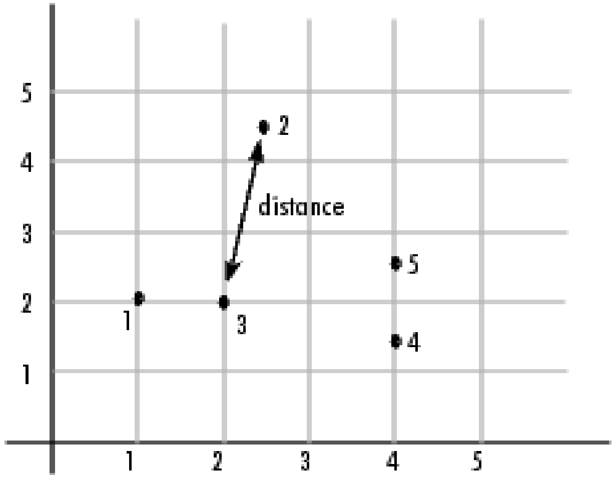
Unsupervised pattern classification and clustering

- Hierarchical Clustering

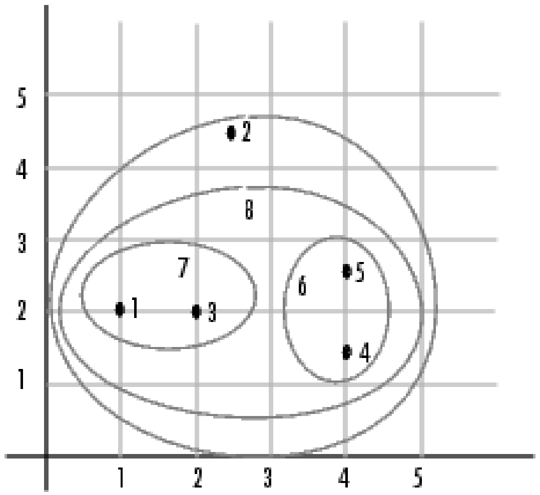
Step 1 - Find the similarity or dissimilarity between every pair of objects in the data set - In this step, you calculate the distance between objects using the “pdist” function.

Step 2 - Group the objects into a binary, hierarchical cluster tree - - In this step, you link pairs of objects that are in close proximity using the “linkage” function. The “linkage” function uses the distance information generated in step 1 to determine the proximity of objects to each other.

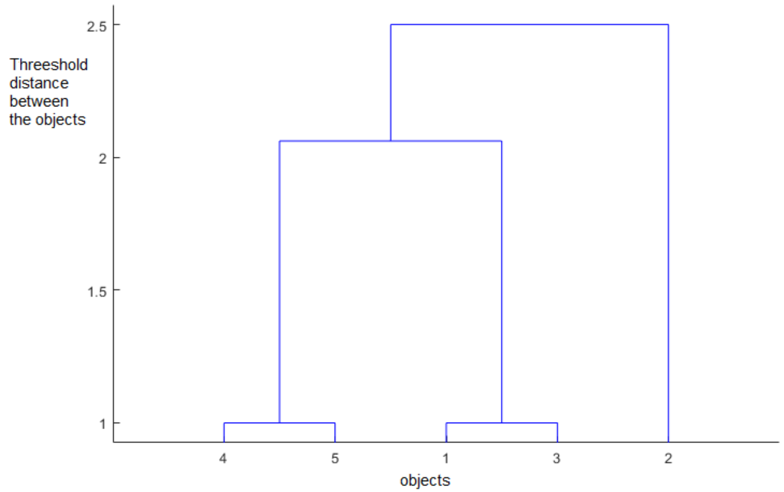
Step 3 - Determine where to cut the hierarchical tree into clusters. In this step, you use the “cluster” function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster. This creates a partition of the data.



Distance Information (MATLAB® site, 2016a)



Linkage (MATLAB® site, 2016a)



Dendogram (MATLAB® site, 2016a)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering

Metric	Description
'euclidean'	Euclidean distance (default).
'squaredeuclidean'	Squared Euclidean distance. (This option is provided for efficiency only. It does not satisfy the triangle inequality.)
'seuclidean'	Standardized Euclidean distance. Each coordinate difference between rows in X is scaled by dividing by the corresponding element of the standard deviation `S=nanstd(X)`. To specify another value for `S`, use `D = pdist(X,'seuclidean',S)`.
'cityblock'	City block metric.
'minkowski'	Minkowski distance. The default exponent is 2. To specify a different exponent, use `D = pdist(X,'minkowski',P)`, where `P` is a scalar positive value of the exponent.
'chebychev'	Chebychev distance (maximum coordinate difference).
'mahalanobis'	Mahalanobis distance, using the sample covariance of `X` as computed by `nancov`. To compute the distance with a different covariance, use `D = pdist(X,'mahalanobis',C)`, where the matrix `C` is symmetric and positive definite.
'cosine'	One minus the cosine of the included angle between points (treated as vectors).
'correlation'	One minus the sample correlation between points (treated as sequences of values).
'spearman'	One minus the sample Spearman's rank correlation between observations (treated as sequences of values).
'hamming'	Hamming distance, which is the percentage of coordinates that differ.
'jaccard'	One minus the Jaccard coefficient, which is the percentage of nonzero coordinates that differ.

custom distance function	A distance function specified using @:
	`D = pdist(X,@distfun)`
	A distance function must be of form
	`d2 = distfun(XI,XJ)`
	taking as arguments a 1-by-*n* vector `XI`, corresponding to a single row of `X`, and an *m*2-by-*n* matrix `XJ`, corresponding to multiple rows of `X`. `distfun` must accept a matrix `XJ` with an arbitrary number of rows. `distfun` must return an *m*2-by-1 vector of distances `d2`, whose *k*th element is the distance between `XI` and `XJ(k,:)`.

“pdist” function metrics (MATLAB® site, 2016b)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering

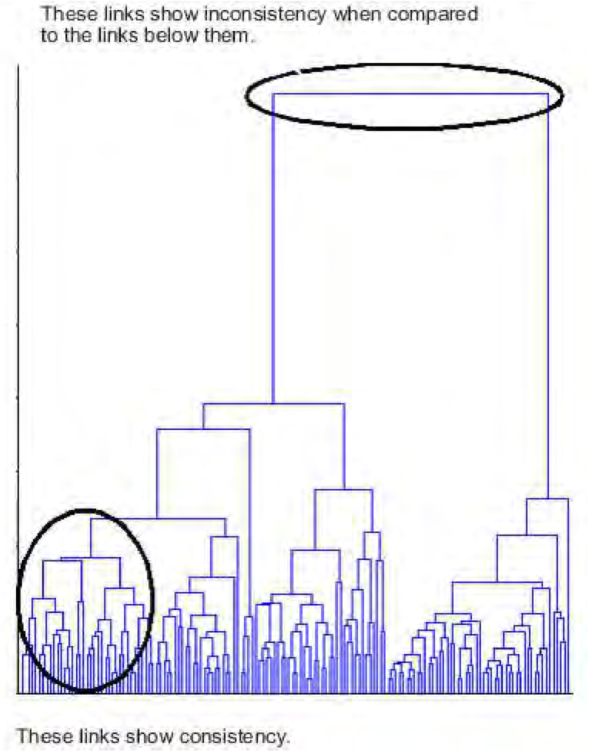
Method	Description
'average'	Unweighted average distance (UPGMA)
'centroid'	Centroid distance (UPGMC), appropriate for Euclidean distances only
'complete'	Furthest distance
'median'	Weighted center of mass distance (WPGMC), appropriate for Euclidean distances only
'single'	Shortest distance
'ward'	Inner squared distance (minimum variance algorithm), appropriate for Euclidean distances only
'weighted'	Weighted average distance (WPGMA)

“linkage” function methods (MATLAB® site, 2016c)

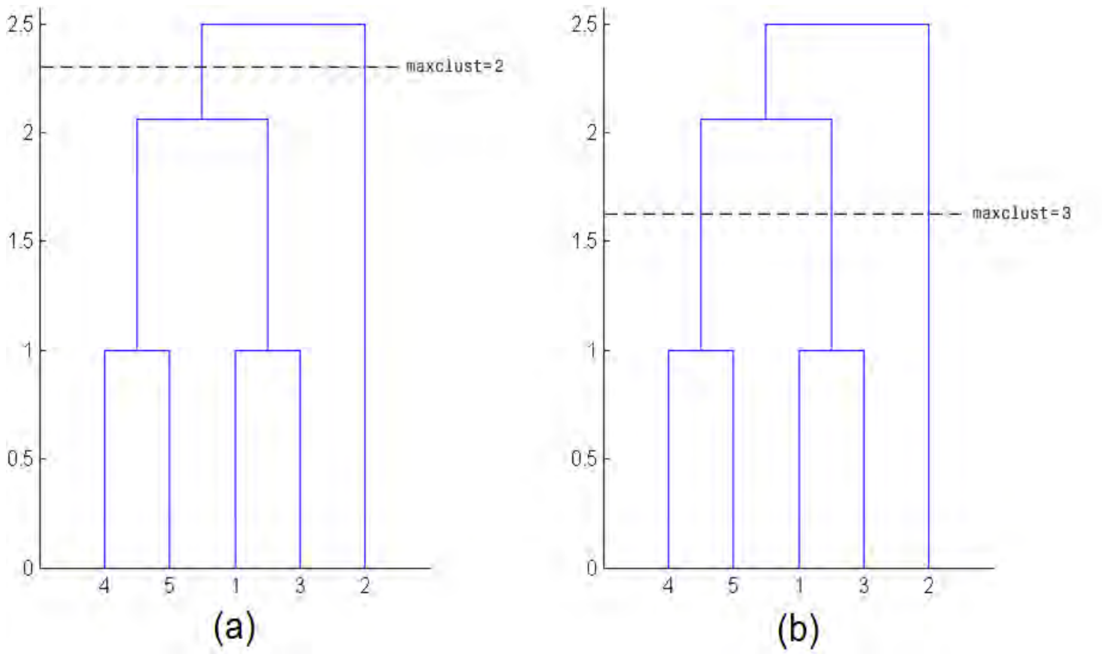
3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering



Consistency in a dendogram (MATLAB® site, 2016a)



Examples of Arbitrary Clusters for: a) 2 clusters; and b) 3 clusters, respectively (MATLAB® site, 2016a)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering for Sentences Task (best result)

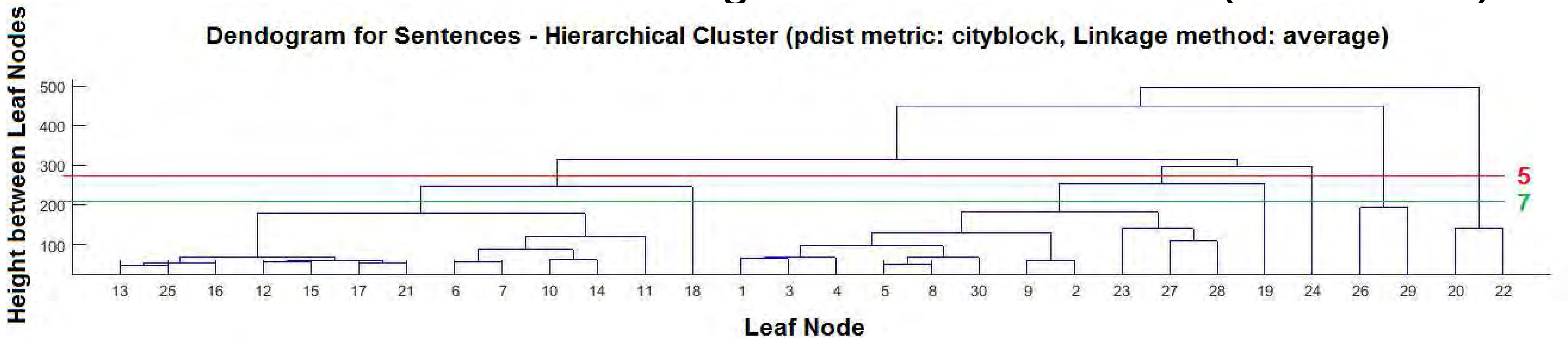
```
Hierarchical Cluster (pdist metric: cityblock Linkage Method: average):
accuracy = 21.63%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    311     265     0       0       0
     S2 |    263     311     0       2       0
     S3 |    260     315     1       0       0
     S4 |    262     313     1       0       0
     S5 |    245     331     0       0       0

Hierarchical Cluster (pdist metric: cityblock Linkage Method: centroid)
accuracy = 21.63%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    311     265     0       0       0
     S2 |    263     311     0       2       0
     S3 |    260     315     1       0       0
     S4 |    263     312     1       0       0
     S5 |    245     331     0       0       0
```

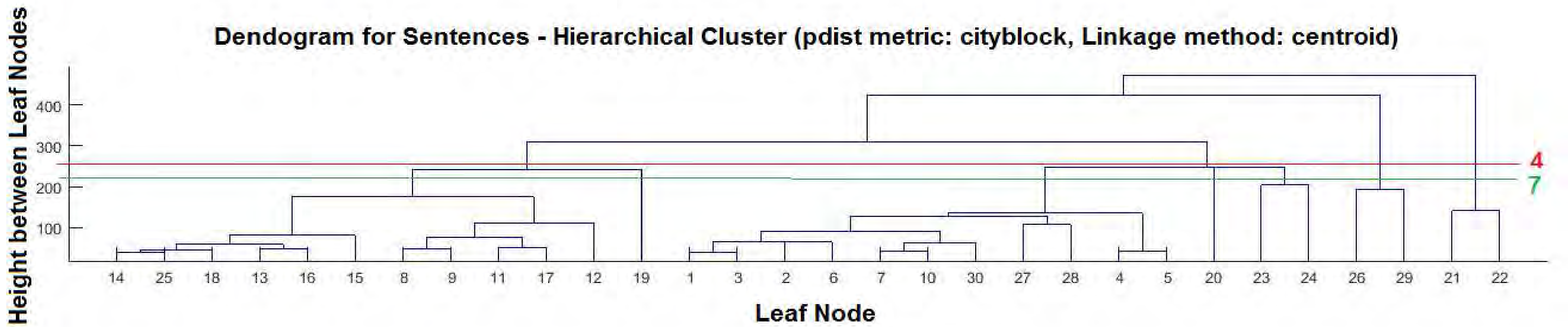
Confusion Matrix and accuracy for the best results of Hierarchical Clustering and Unsupervised Classifiers for Sentences Task

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering for Sentences Task (best result)





Dendograms for the best results of Hierarchical Clustering and Unsupervised Classifiers for Sentences Task

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering for Words Task (best result)

Hierarchical Cluster (pdist metric: spearman Linkage Method: complete):
accuracy = 28.21%
Confusion Matrix for the test

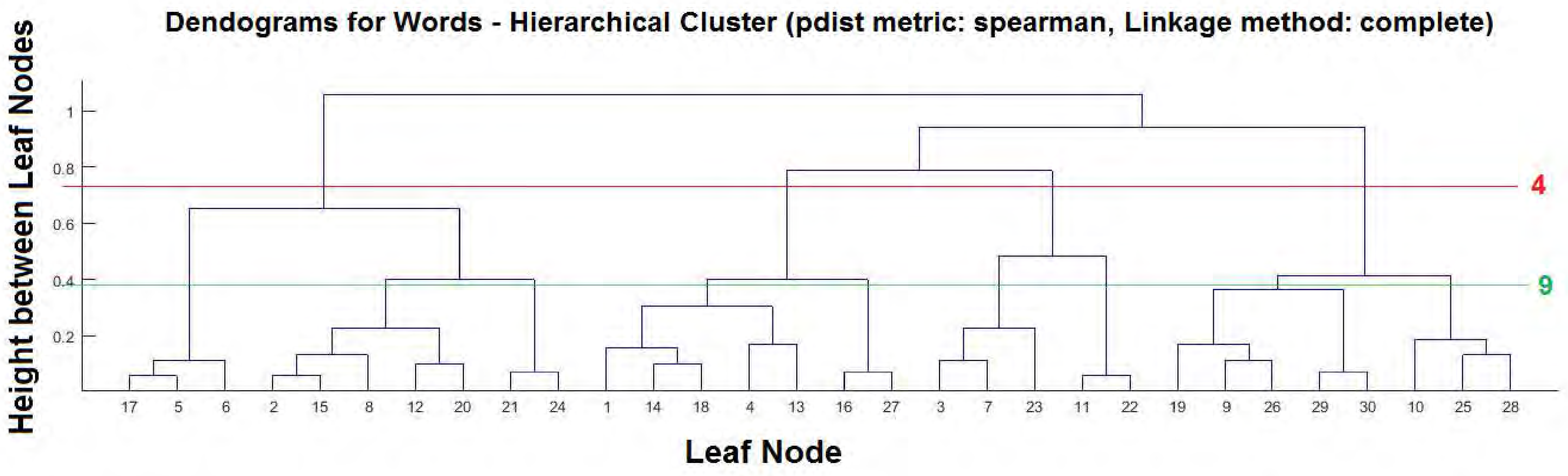
T	S1	S2	S3	S4
S1	65	41	30	440
S2	52	57	26	441
S3	42	13	27	494
S4	47	18	10	501

Confusion Matrix and accuracy for the best results of Hierarchical Clustering and Unsupervised Classifiers for Words Task

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Hierarchical Clustering for Words Task (best result)



Dendograms for the best results of Hierarchical Clustering and Unsupervised Classifiers for Words Task

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means

“kmeans” function do the partitions of data into k mutually exclusive clusters, and returns the index of the cluster to which it has assigned each observation.

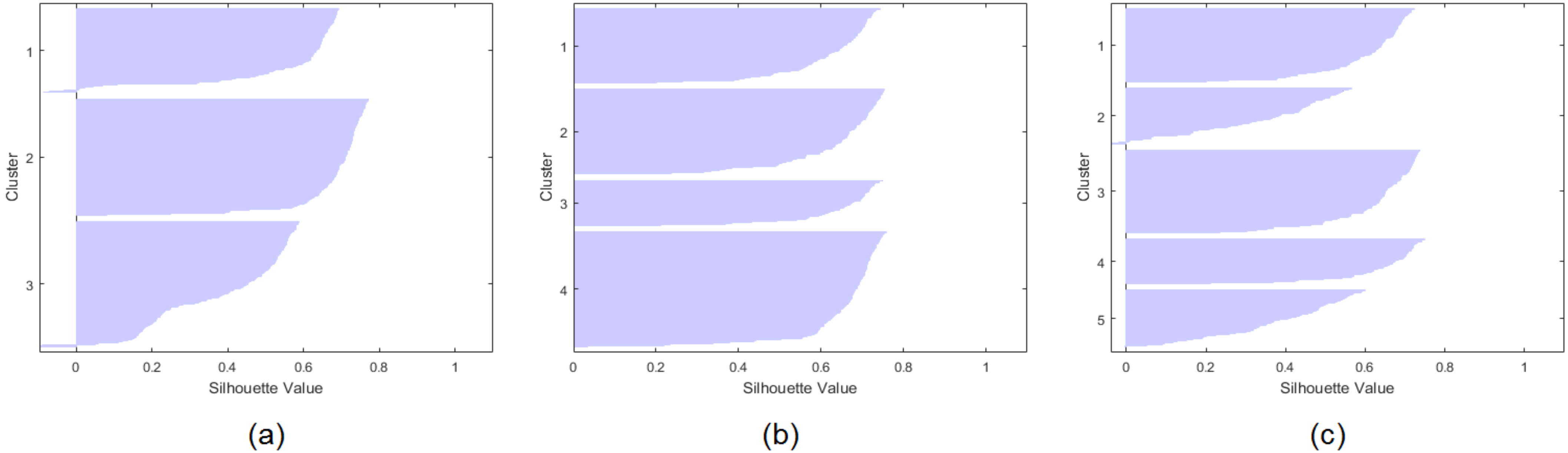
Distance Measure	Description	Formula
'sqeuclidean'	Squared Euclidean distance (default). Each centroid is the mean of the points in that cluster.	$d(x,c) = (x-c)(x-c)'$
'cityblock'	Sum of absolute differences, i.e., the $L1$ distance. Each centroid is the component-wise median of the points in that cluster.	$d(x,c) = \sum_{j=1}^{p} \lvert x_j - c_j \rvert$
'cosine'	One minus the cosine of the included angle between points (treated as vectors). Each centroid is the mean of the points in that cluster, after normalizing those points to unit Euclidean length.	$d(x,c) = 1 - \frac{xc'}{\sqrt{(xx')(cc')}}$
'correlation'	One minus the sample correlation between points (treated as sequences of values). Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.	$d(x,c) = 1 - \frac{(x-\vec{\bar{x}})(c-\vec{\bar{c}})'}{\sqrt{(x-\vec{\bar{x}})(x-\vec{\bar{x}})'}\sqrt{(c-\vec{\bar{c}})(c-\vec{\bar{c}})'}},$ where • $\vec{\bar{x}} = \frac{1}{p}\left(\sum_{j=1}^{p} x_j\right)\vec{1}_p$ • $\vec{\bar{c}} = \frac{1}{p}\left(\sum_{j=1}^{p} c_j\right)\vec{1}_p$ • $\vec{1}_p$ is a row vector of p ones.
'hamming'	This measure is only suitable for binary data. It is the proportion of bits that differ. Each centroid is the component-wise median of points in that cluster.	$d(x,y) = \frac{1}{p}\sum_{j=1}^{p} I\{x_j \neq y_j\},$ where I is the indicator function.

“kmeans” function metrics (MATLAB® site, 2016e)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means



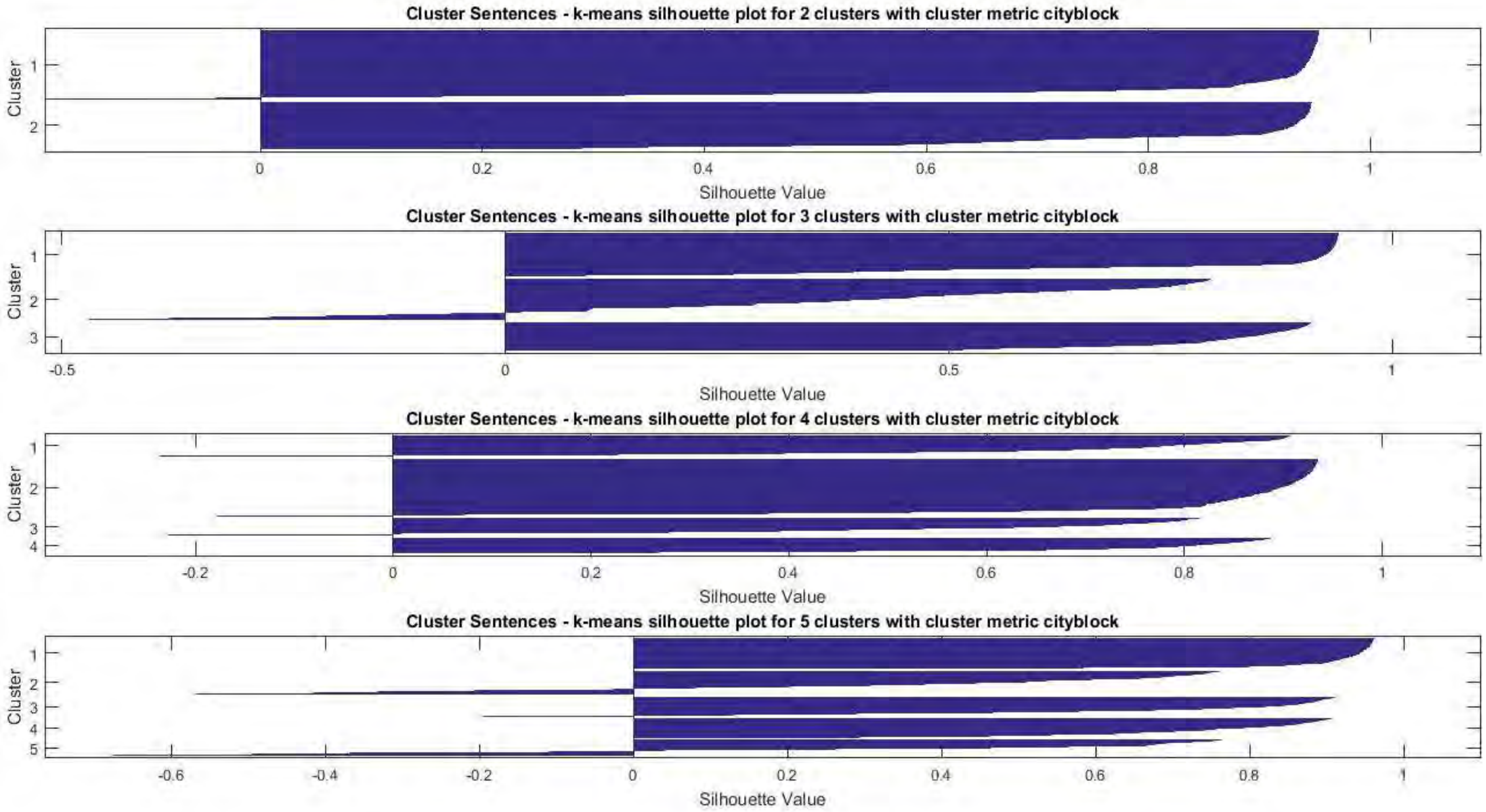
Silhouette for k-means clustering example for: a) 3 clusters; b) 4 clusters; and c) 5 clusters (MATLAB® site, 2016d)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Sentences Task (metric “cityblock”)

Silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric “cityblock”



3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Sentences Task (metric “cityblock”)

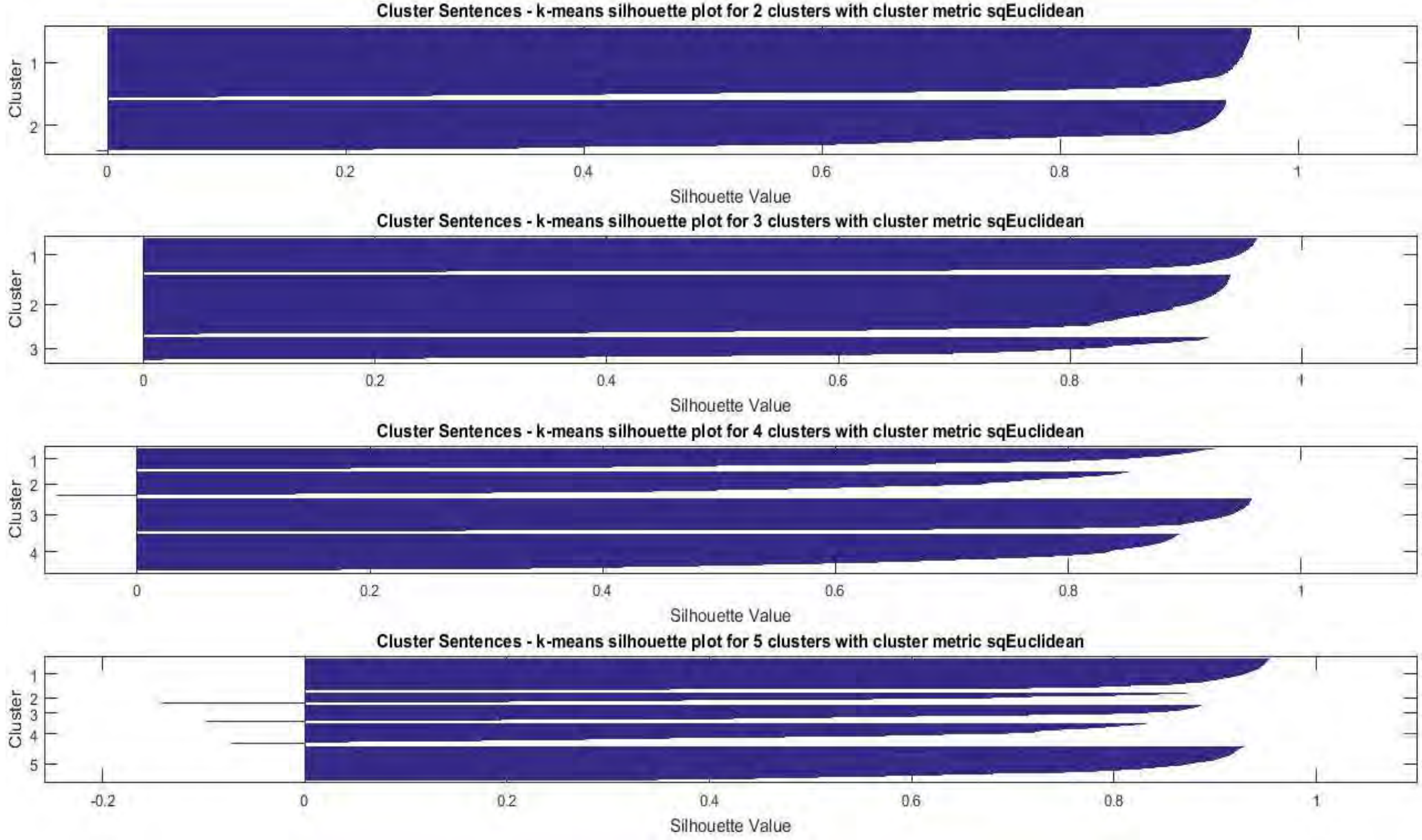
```
kmeans cityblock with 2 clusters:
accuracy = 52.92%
Confusion Matrix for the test
     T  |     S1       S2
     S1 |     487      665
     S2 |     691      1037

kmeans cityblock with 5 clusters:
accuracy = 19.44%
Confusion Matrix for the test
     T  |     S1       S2       S3       S4       S5
     S1 |     135      129      82       83       147
     S2 |     170      141      72       93       100
     S3 |     184      130      118      60       84
     S4 |     186      120      85       85       100
     S5 |     209      122      74       90       81
```

For the verification of the 2 clusters and to maintain the coherence with the 5 original classes of Soto(2014) experiment, the classes S1 and S2 (congruous) are jointed in the first cluster and S3, S4 and S5 (incongruous) are jointed in the second cluster to the verification.

K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 5 clusters (real number of classes) of Sentences Task with k-means metric “cityblock”

Unsupervised pattern classification and clustering

- k-means for Sentences Task (metric “sqEuclidean”)



Silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric “sqEuclidean”

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Sentences Task (metric “sqEuclidean”)

kmeans sqEuclidean with 2 clusters:

accuracy = 53.33%

Confusion Matrix for the test

T	S1	S2
S1	522	630
S2	714	1014

kmeans sqEuclidean with 5 clusters:

accuracy = 18.13%

Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	143	158	62	108	105
S2	160	158	51	83	124
S3	180	172	43	63	118
S4	198	162	52	80	84
S5	200	156	55	67	98

For the verification of the 2 clusters and to maintain the coherence with the 5 original classes of Soto(2014) experiment, the classes S1 and S2 (congruous) are jointed in the first cluster and S3, S4 and S5 (incongruous) are jointed in the second cluster to the verification.

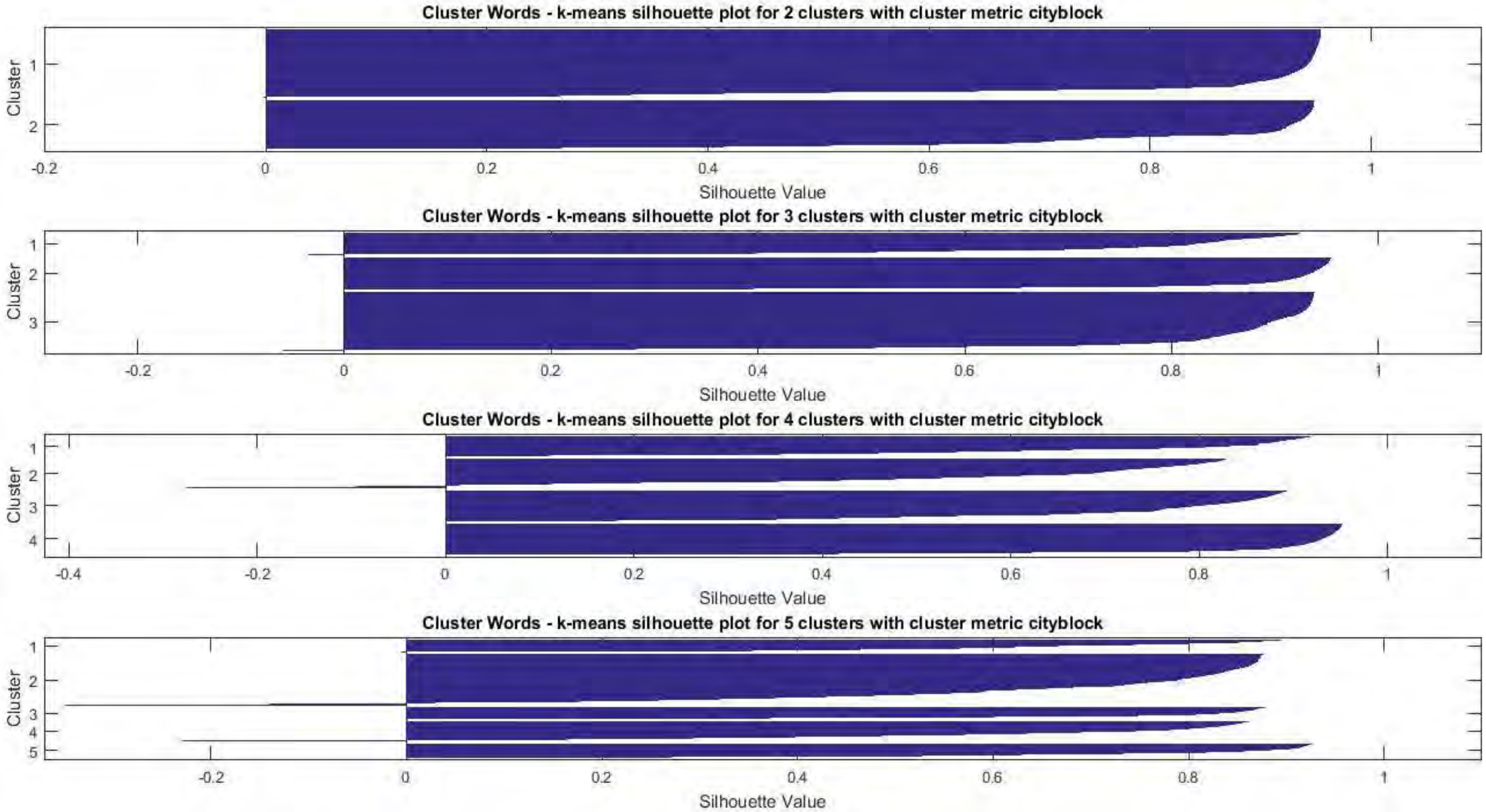
K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 5 clusters (real number of classes) of Sentences Task with k-means metric “sqEuclidean”

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Words Task (metric “cityblock”)

Silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric “cityblock”



3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Words Task (metric “cityblock”)

kmeans cityblock with 2 clusters:

accuracy = 48.44%

Confusion Matrix for the test

T	S1	S2
S1	658	494
S2	694	458

kmeans cityblock with 4 clusters:

accuracy = 23.26%

Confusion Matrix for the test

T	S1	S2	S3	S4
S1	103	282	55	136
S2	98	286	67	125
S3	119	319	51	87
S4	93	307	80	96

For the verification of the 2 clusters and to maintain the coherence with the 4 original classes of Soto (2014) experiment, the classes S1 and S2 (semantic) are jointed in the first cluster and S3 and S4 (no semantic) are jointed in the second cluster to the verification.

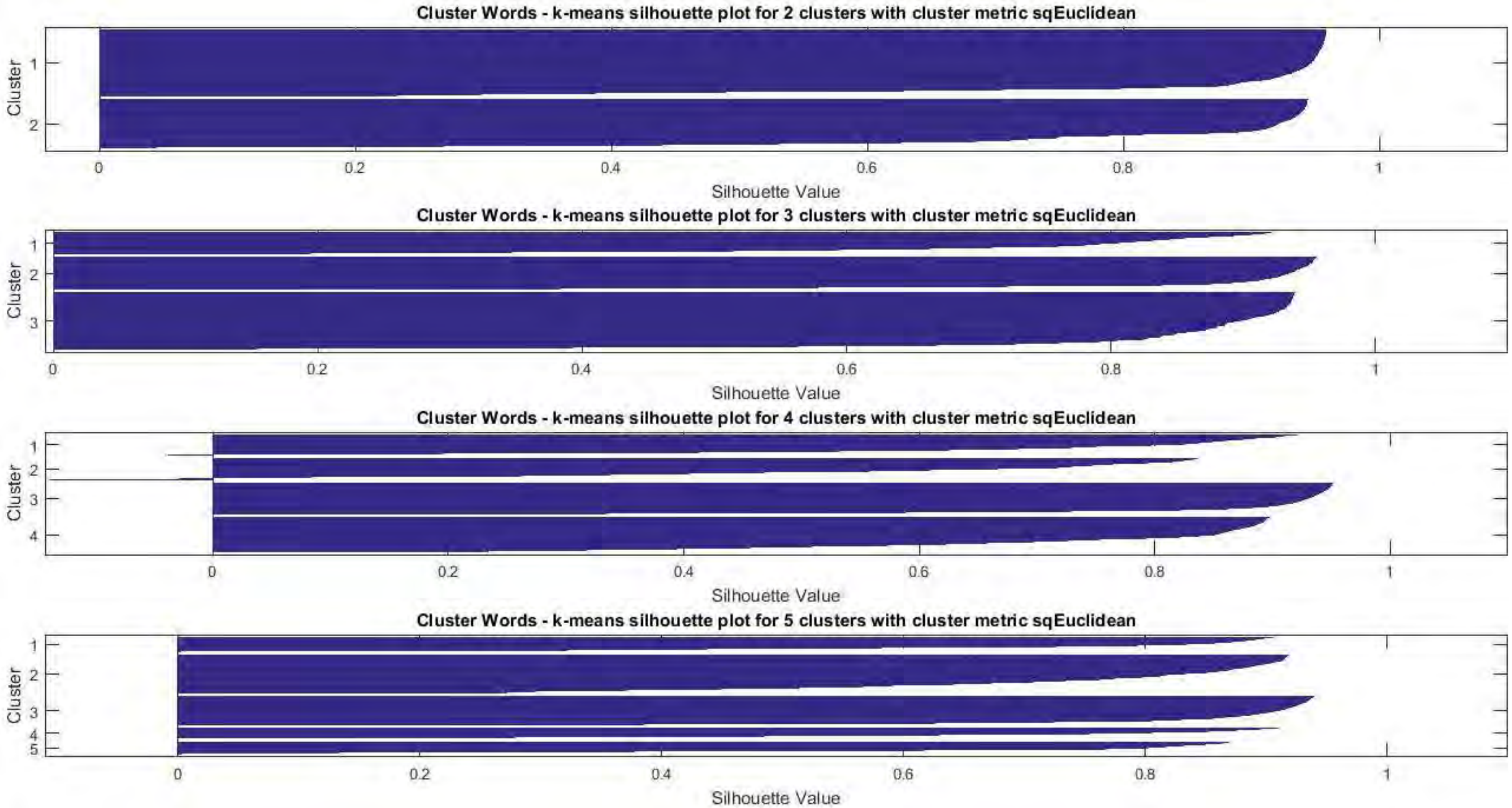
K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 4 clusters (real number of classes) of Words Task with k-means metric “cityblock”

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Words Task (metric “sqEuclidean”)

Silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric “sqEuclidean”



3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- k-means for Words Task (metric “sqEuclidean”)

kmeans sqEuclidean with 3 clusters:
accuracy = 32.68%
Confusion Matrix for the test

T	S1	S2	S3
S1	465	158	145
S2	407	165	196
S3	302	343	123

kmeans sqEuclidean with 4 clusters:
accuracy = 25.00%
Confusion Matrix for the test

T	S1	S2	S3	S4
S1	134	100	158	184
S2	124	91	164	197
S3	88	108	169	211
S4	94	127	173	182

For the verification of the 3 clusters and to maintain the coherence with the 4 original classes of Soto (2014) experiment and observing the Siulhouette plot distribution, the classes S1 and S2 (semantic) are keep separated and S3 and S4 (no semantic) are jointed in the third cluster to the verification.

K-Means unsupervised classifiers for 3 clusters (best silhouette result) and for 4 clusters (real number of classes) of Sentences Task with k-means metric “sqEuclidean”

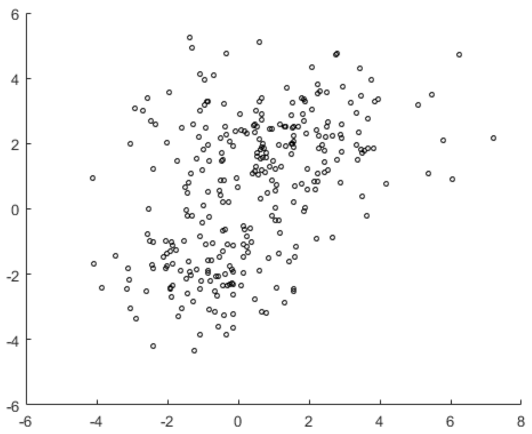
3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- **Gaussian Mixture Models**

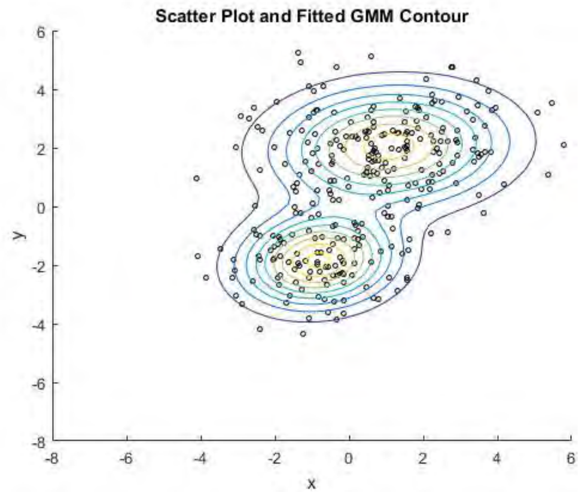
In the mixture method of clustering, each different group in the population is assumed to be described by a different probability distribution that may belong to the same family but differ in the values they take for the parameters of the distribution.

Step 1 - Fit a two-component Gaussian mixture model (GMM)



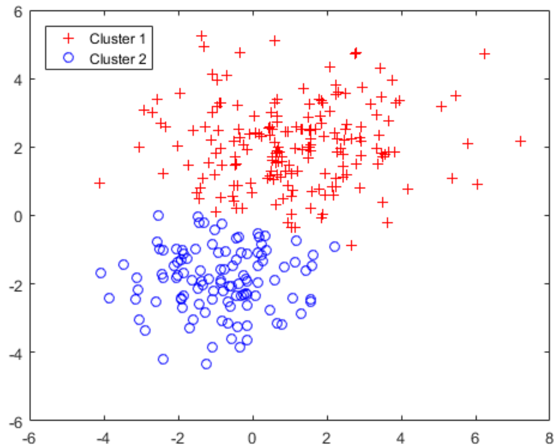
Simulate data from a mixture of two bivariate Gaussian distribution (MATLAB® site, 2016g)

Step 2 - plot the estimated probability density contours for the two-component mixture distribution.



Scatter Plot and Fitted GMM Contour (MATLAB® site, 2016g)

Step 3 - Cluster the Data Using the Fitted GMM.

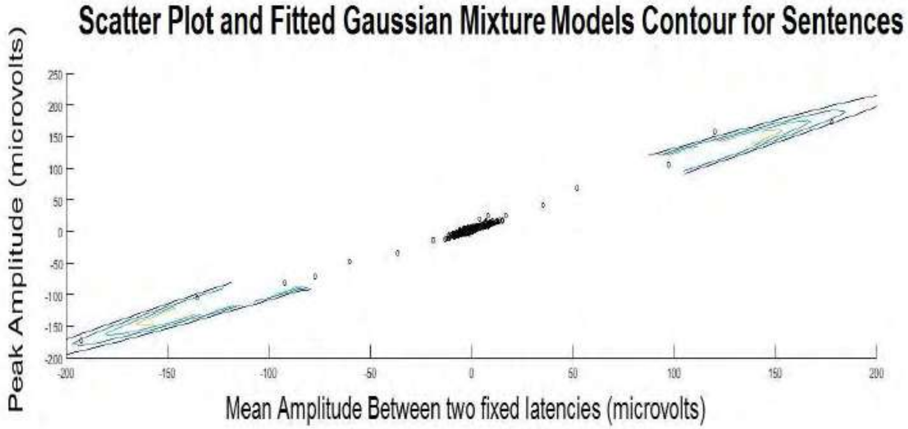


Scattering plot of the GMM fitted clusters (MATLAB® site, 2016g)

3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Gaussian Mixture Models for Sentences Task

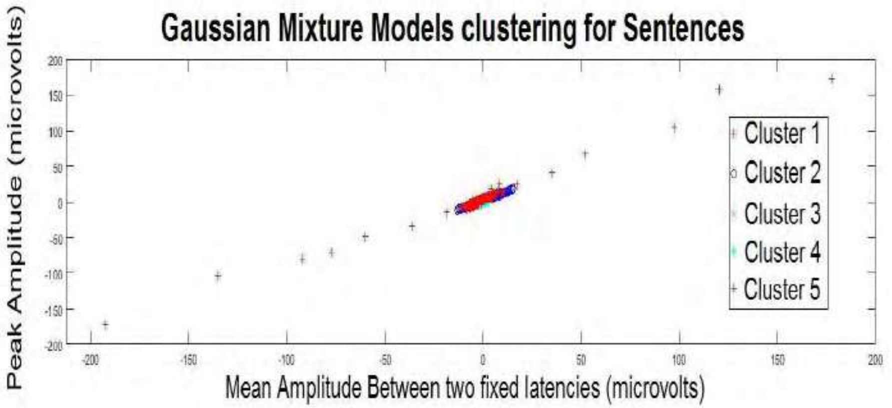


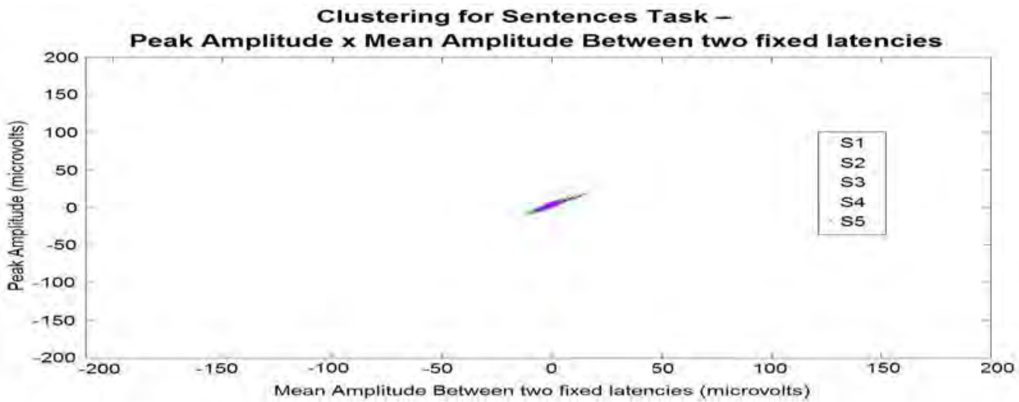
Gaussian Mixture Models Cluster for Sentences
MeanAmp2FixedLat and PeakAmp Attributes:

accuracy = 19.24%

Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	3	64	208	227	74
S2	3	37	188	242	106
S3	4	35	229	217	91
S4	3	40	189	243	101
S5	2	31	143	358	42

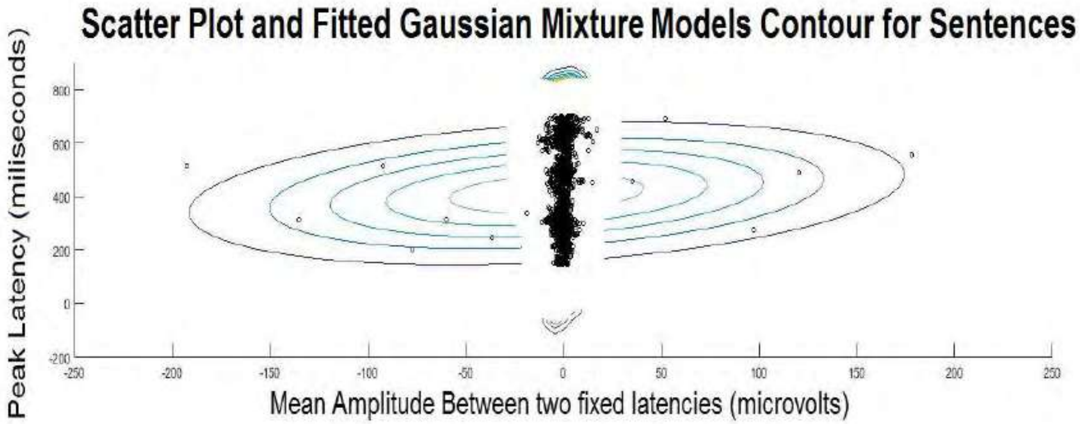




3. Methodology, Results and Discussion

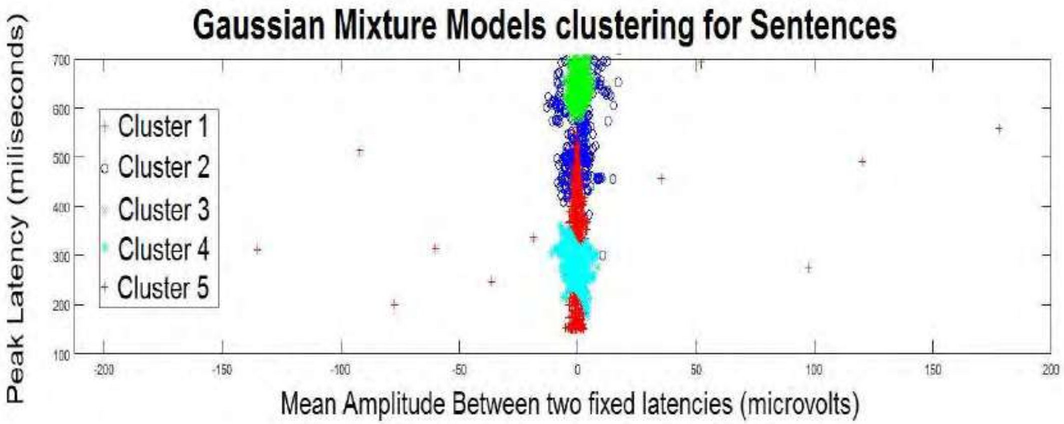
Unsupervised pattern classification and clustering

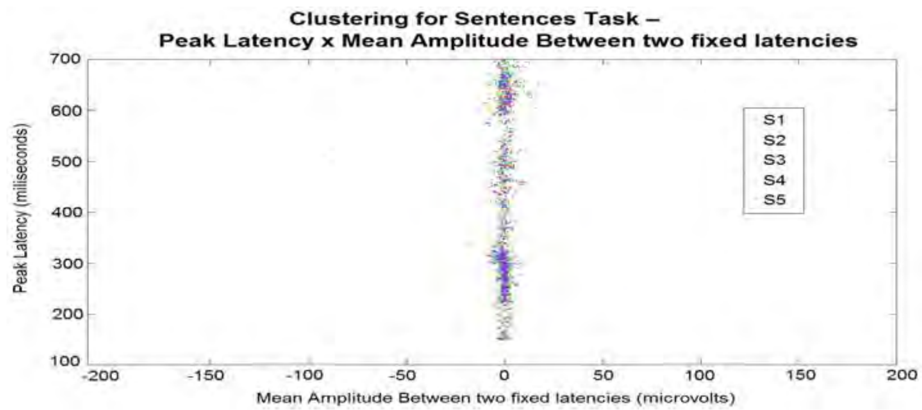
- Gaussian Mixture Models for Sentences Task



Gaussian Mixture Models Cluster for Sentences
MeanAmp2FixedLat and PeakLat Attributes:
accuracy = 21.18%
Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	158	70	137	209	2
S2	146	65	131	231	3
S3	147	32	157	237	3
S4	151	43	150	229	3
S5	148	51	136	240	1

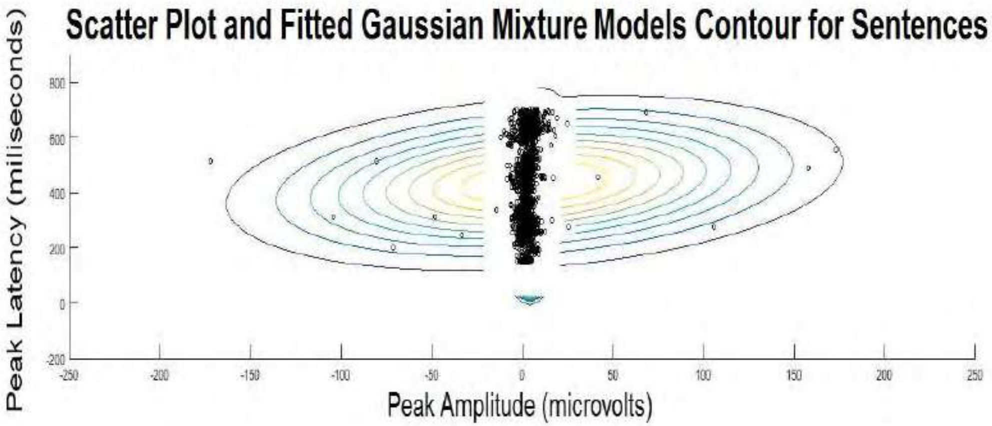




3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Gaussian Mixture Models for Sentences Task

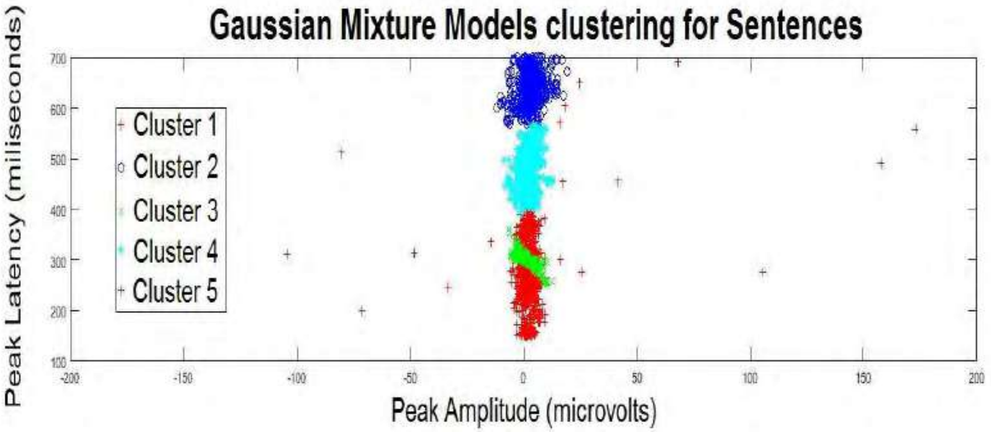


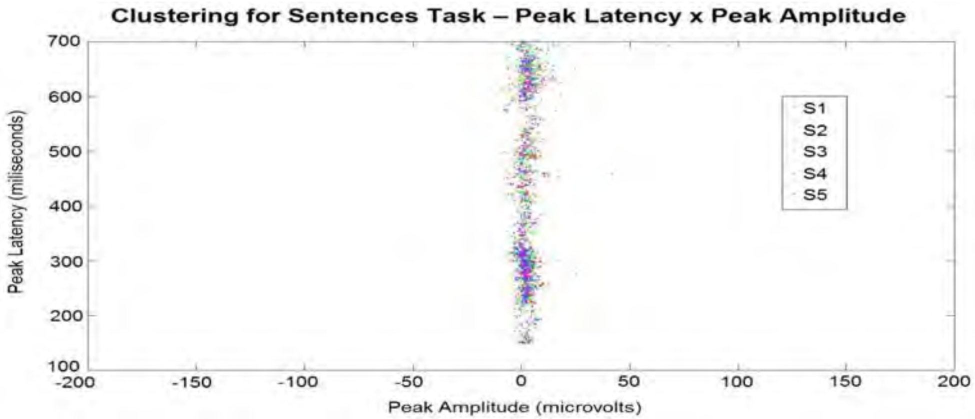
Gaussian Mixture Models Cluster for Sentences
PeakAmp and PeakLat Attributes:

accuracy = 19.24%

Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	202	156	62	150	6
S2	228	158	85	102	3
S3	220	172	94	87	3
S4	217	161	96	98	4
S5	245	155	86	88	2

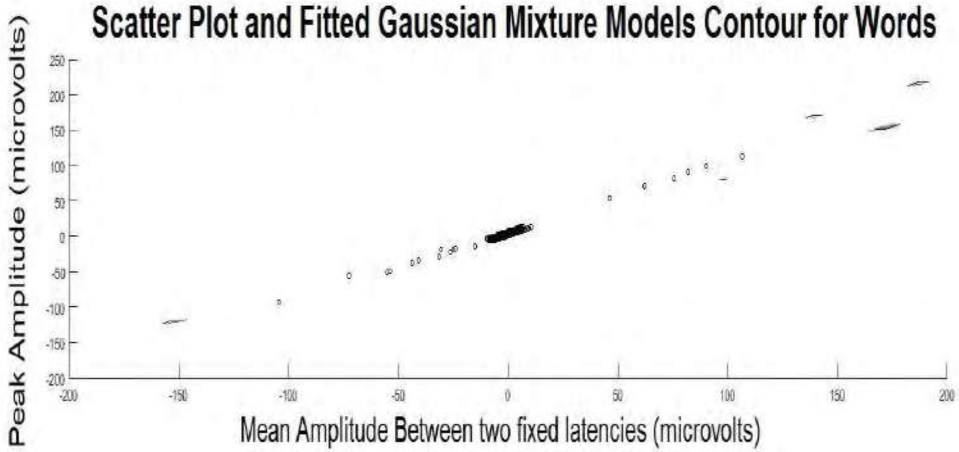




3. Methodology, Results and Discussion

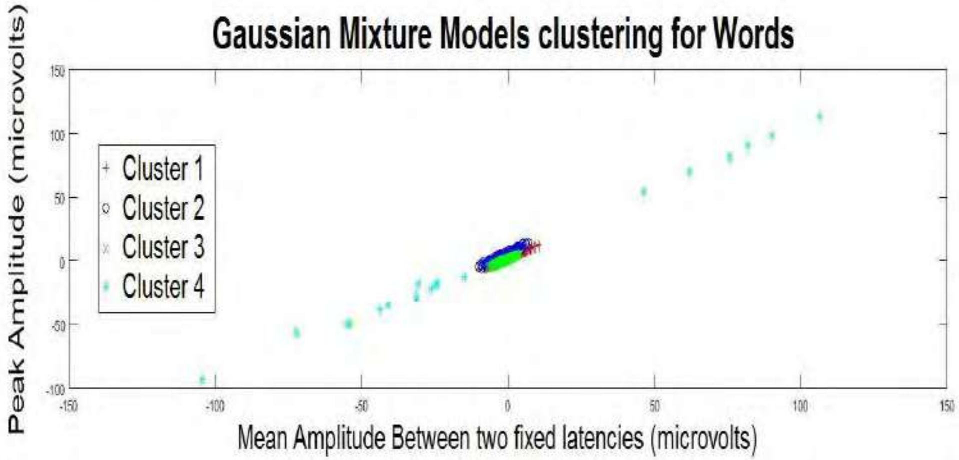
Unsupervised pattern classification and clustering

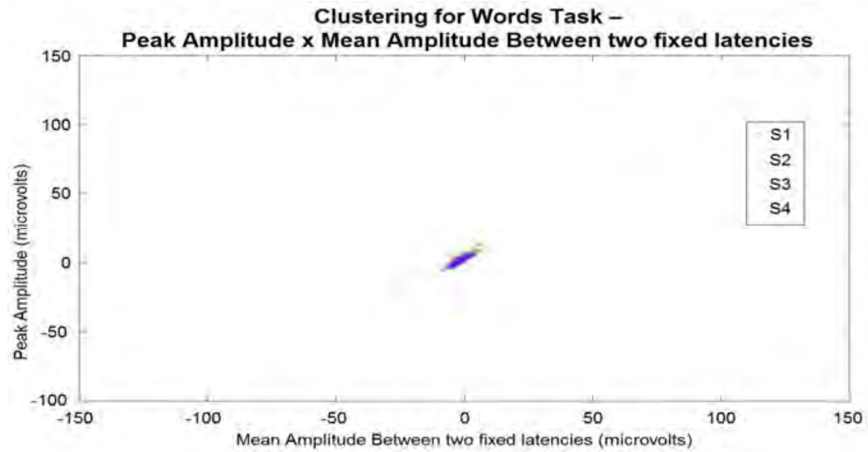
- Gaussian Mixture Models for Words Task



Gaussian Mixture Models Cluster for Words
MeanAmp2FixedLat and PeakAmp Attributes):
accuracy = 24.91%
Confusion Matrix for the test

T	S1	S2	S3	S4
S1	256	96	218	6
S2	268	79	226	3
S3	260	80	233	3
S4	202	52	316	6

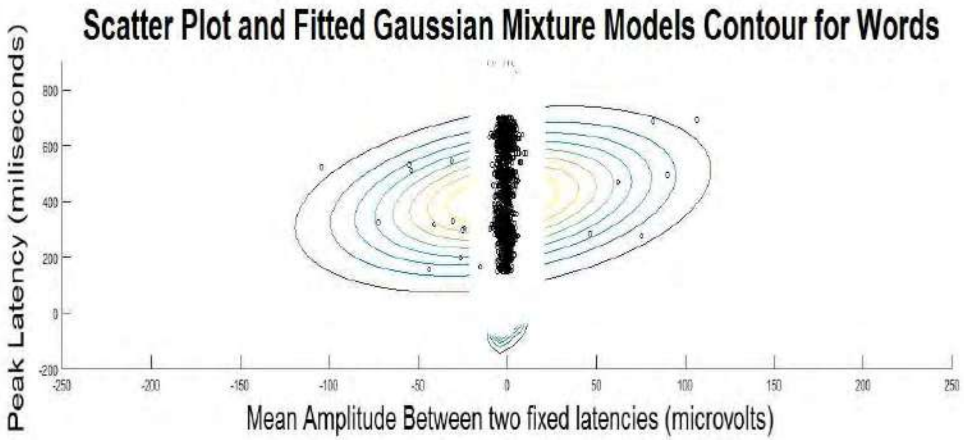




3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Gaussian Mixture Models for Words Task



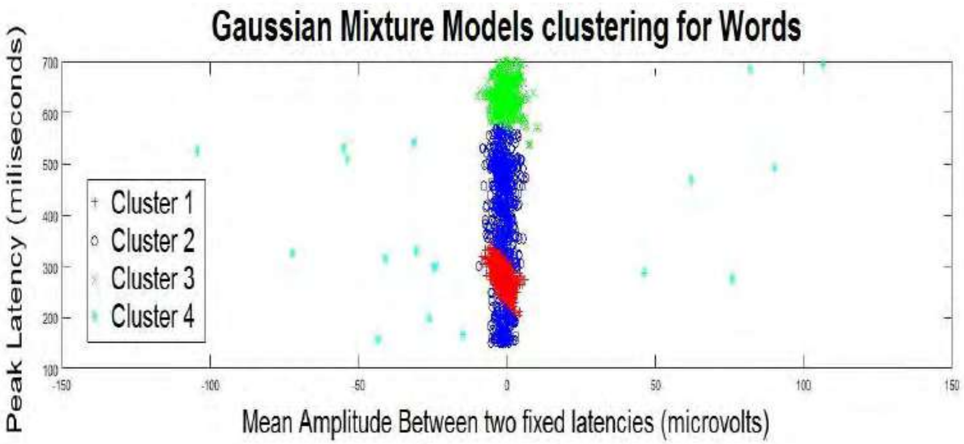
Gaussian Mixture Models Cluster for Words

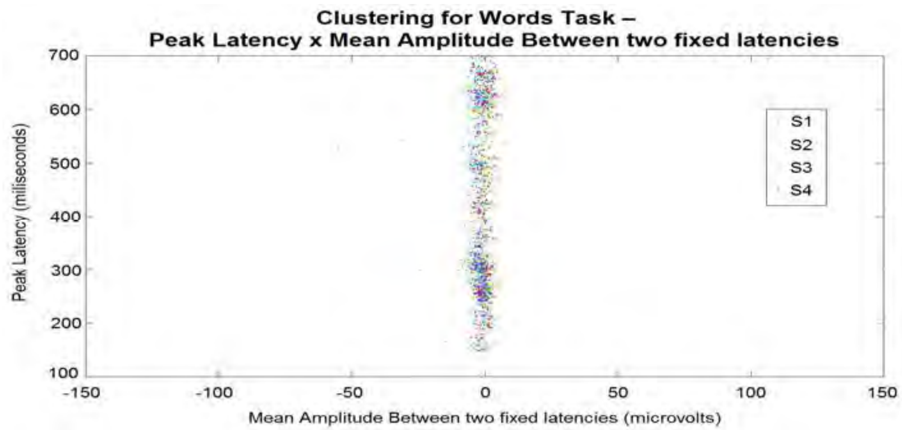
MeanAmp2FixedLat and PeakLat Attributes:

accuracy = 24.78%

Confusion Matrix for the test

T	S1	S2	S3	S4
S1	213	212	145	6
S2	224	195	154	3
S3	239	177	157	3
S4	226	184	160	6

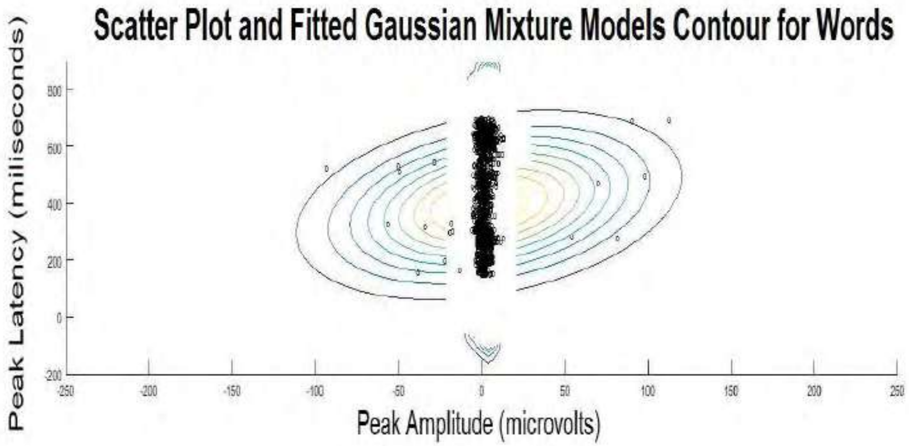




3. Methodology, Results and Discussion

Unsupervised pattern classification and clustering

- Gaussian Mixture Models for Words Task



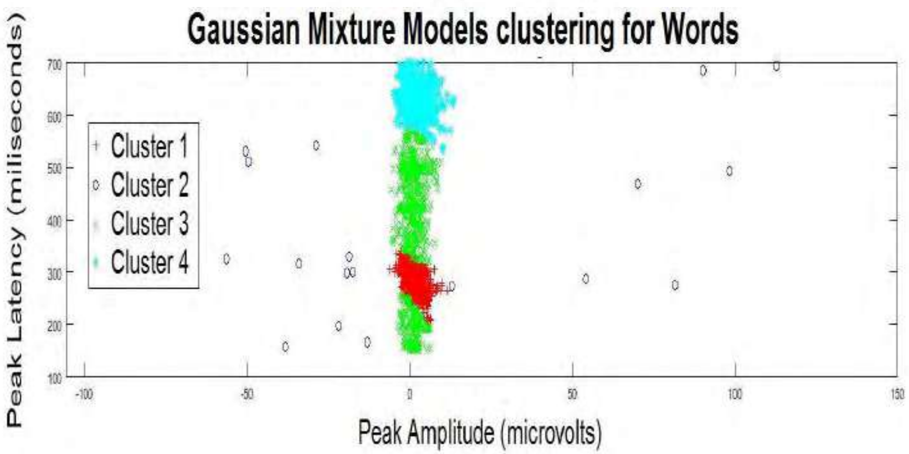
Gaussian Mixture Models Cluster for Words

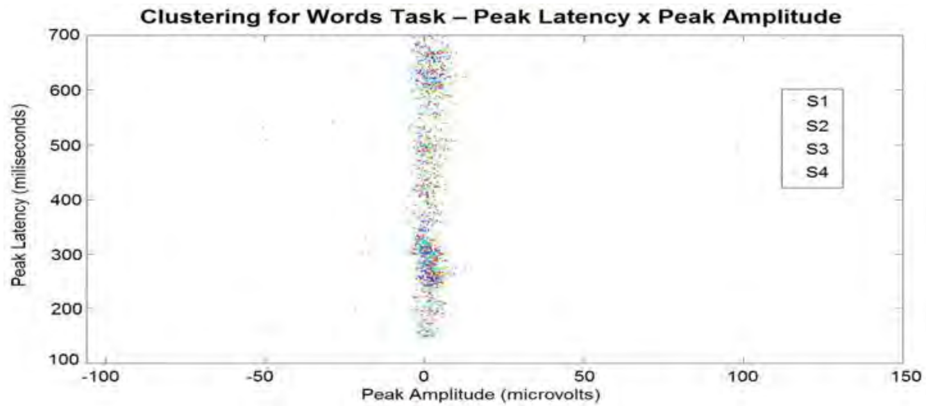
PeakAmp and PeakLat Attributes:

accuracy = 24.39%

Confusion Matrix for the test

T	S1	S2	S3	S4
S1	205	7	216	148
S2	216	3	200	157
S3	223	3	194	156
S4	213	6	197	160





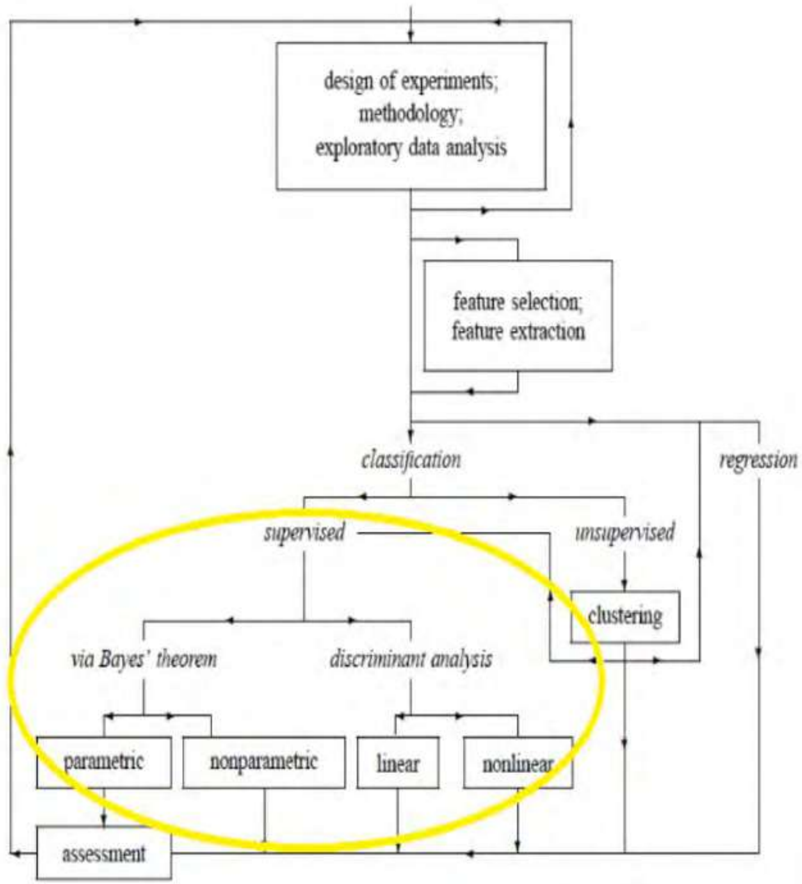
3. Methodology, Results and Discussion

4. Supervised pattern classification;

For supervised classification:

the data were splitted in three sets with the same amount of data with all features for each class coming from the Words and Sentence Task. The sets are defined as training set, validation set and test set, respectively, with 1/3 of the total amount.

it were used to assess the performances of the classifiers, for each try, the confusion matrixes, the accuracies and the receiver operating characteristic (ROC) curve.



3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Naïve Bayes

Naive Bayes classifiers assign observations to the most probable class (in other words, the maximum a posteriori decision rule). Explicitly, the algorithm:

a. Estimates the densities of the predictors within each class;

b. Models posterior probabilities according to Bayes rule. That is, for all k = 1,...,K,

$$\hat{P}(Y = k \mid X_1, \ldots, X_P) = \frac{\pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)}{\sum_{k=1}^{K} \pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)},$$

where:

Y is the random variable corresponding to the class index of an observation.

$X_1, \ldots, X_P$ are the random predictors of an observation.

$\pi(Y=k)$ is the prior probability that a class index is k.

c. Classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

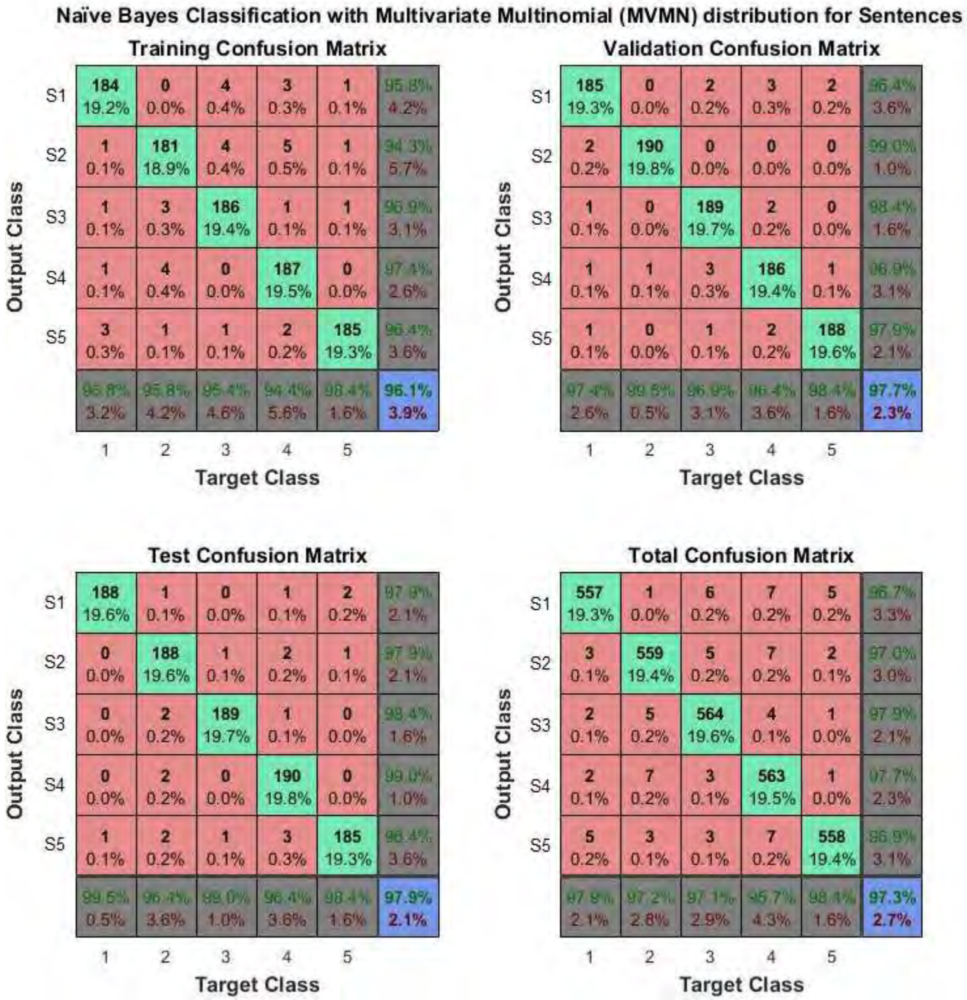
- Naïve Bayes

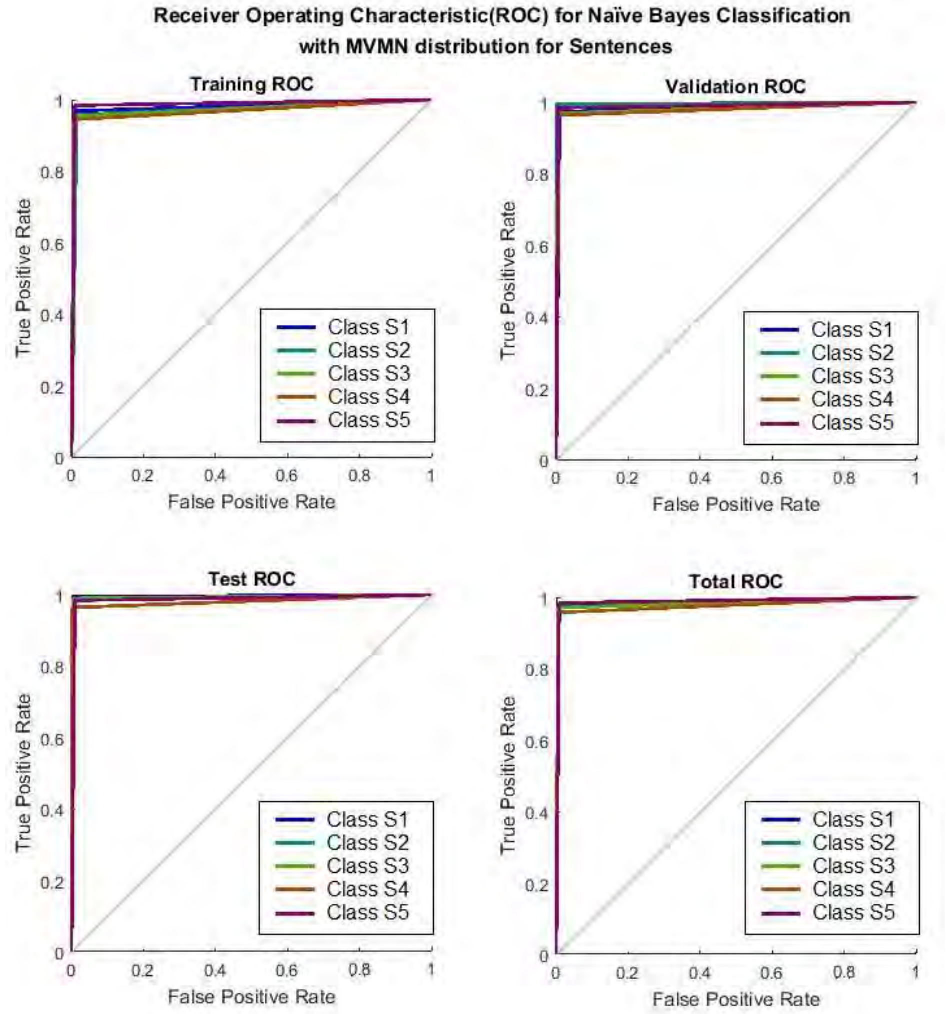
Probability distribution values for “fitNaiveBayes” function (MATLAB® site, 2016i)

Value	Description
'kernel'	Kernel smoothing density estimate.
'mn'	Multinomial distribution. If you specify mn, then all features are components of a multinomial distribution. Therefore, you cannot include 'mn' as an element of a cell array of character vectors.
'mvmn'	Multivariate multinomial distribution.
'normal'	Normal (Gaussian) distribution.

3. Methodology, Results and Discussion

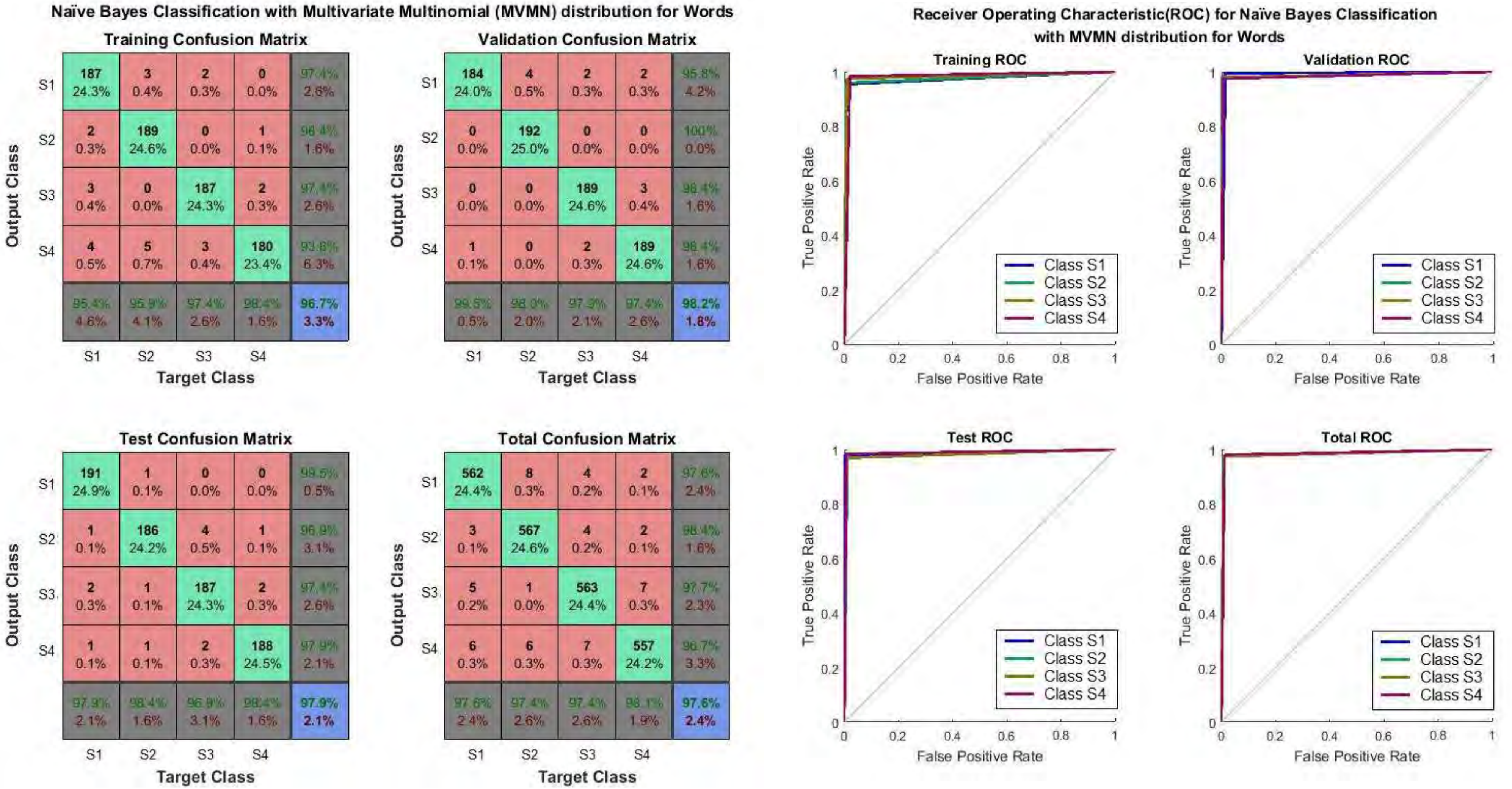
Naïve Bayes Sentences (best result)





3. Methodology, Results and Discussion

Naïve Bayes Words (best result)

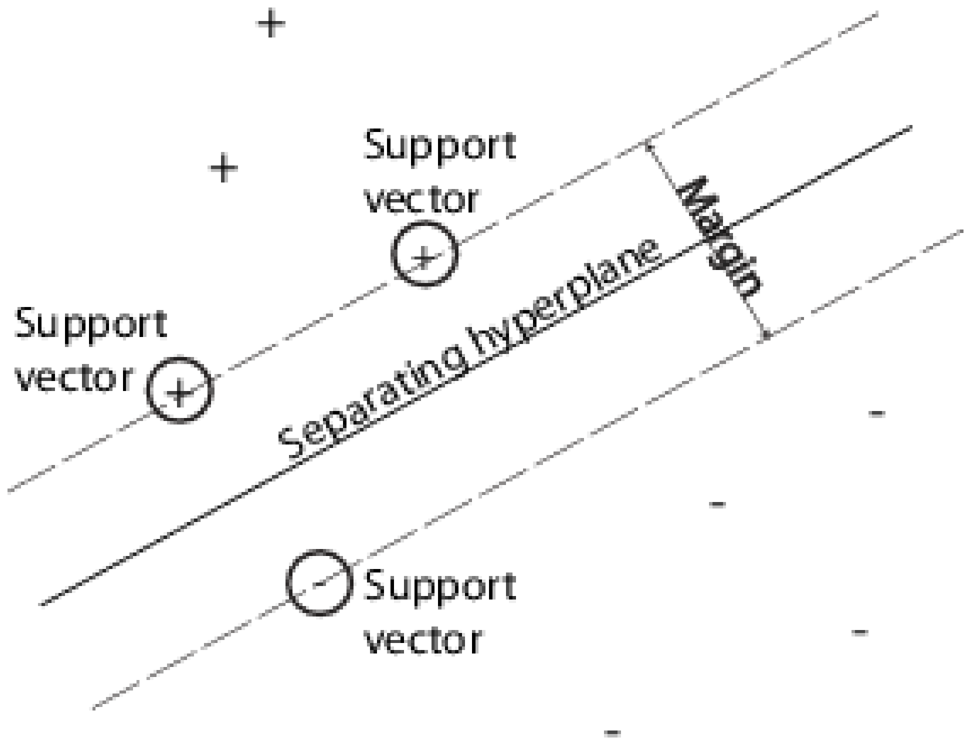


3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Multiclass Support Vector Machine (SVM)

The support vectors are the data points that are closest to the separating hyperplane; these points are on the boundary of the slab. The following figure illustrates it, with + indicating data points of type 1, and – indicating data points of type –1.



Support Vectors (MATLAB® site, 2016j)

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- ## Multiclass Support Vector Machine (SVM)

The mainly parameters (MATLAB® site, 2016l) used in this study were:

- Box Constraint – A parameter that controls the maximum penalty imposed on margin-violating observations, and aids in preventing overfitting (regularization). If you increase the box constraint, then the SVM classifier assigns fewer support vectors. However, increasing the box constraint can lead to longer training times.

- Kernel Function – Kernel function is used to compute the Gram matrix, specified as the comma-separated pair consisting of 'KernelFunction'. The Gram matrix of a set of n vectors $\{x_1,..,x_n;\ x_j \in R^p\}$ is an n-by-n matrix with element (j,k) defined as $G(x_j,x_k) = <\phi(x_j),\phi(x_k)>$ an inner product of the transformed predictors using the kernel function ϕ. For nonlinear SVM, the algorithm forms a Gram matrix using the predictor matrix columns. The dual formalization replaces the inner product of the predictors with corresponding elements of the resulting Gram matrix (called the "kernel trick"). Subsequently, nonlinear SVM operates in the transformed predictor space to find a separating hyperplane. The kernel functions available for this method are 'linear', 'gaussian' or 'rbf', and 'polynomial'; and

- Standardize – This parameter is a flag to standardize the predictor data, specified as the comma-separated pair consisting of 'Standardize' and true (1) or false (0). If you set 'Standardize',true, the software centers and scales each column of the predictor data (X) by the weighted column mean and standard deviation, respectively. MATLAB® does not standardize the data contained in the dummy variable columns generated for categorical predictors. The software trains the classifier using the standardized predictor matrix, but stores the unstandardized data in the classifier property X.

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Multiclass Support Vector Machine (SVM)

"templateSVM" function kernel functions (Matlab® site, 2016l) used in ClassificationECOC class

Value	Description	Formula
'gaussian' or 'rbf	Gaussian or Radial Basis Function (RBF) kernel, default for one-class learning	$G(x_1, x_2) = \exp(-\|x_1 - x_2\|^2)$
'linear'	Linear kernel, default for two-class learning	$G(x_1, x_2) = x_1'x_2$
'polynomial'	Polynomial kernel. Use 'PolynomialOrder',p to specify a polynomial kernel of order p.	$G(x_1, x_2) = (1 + x_1'x_2)^p$

3. Methodology, Results and Discussion

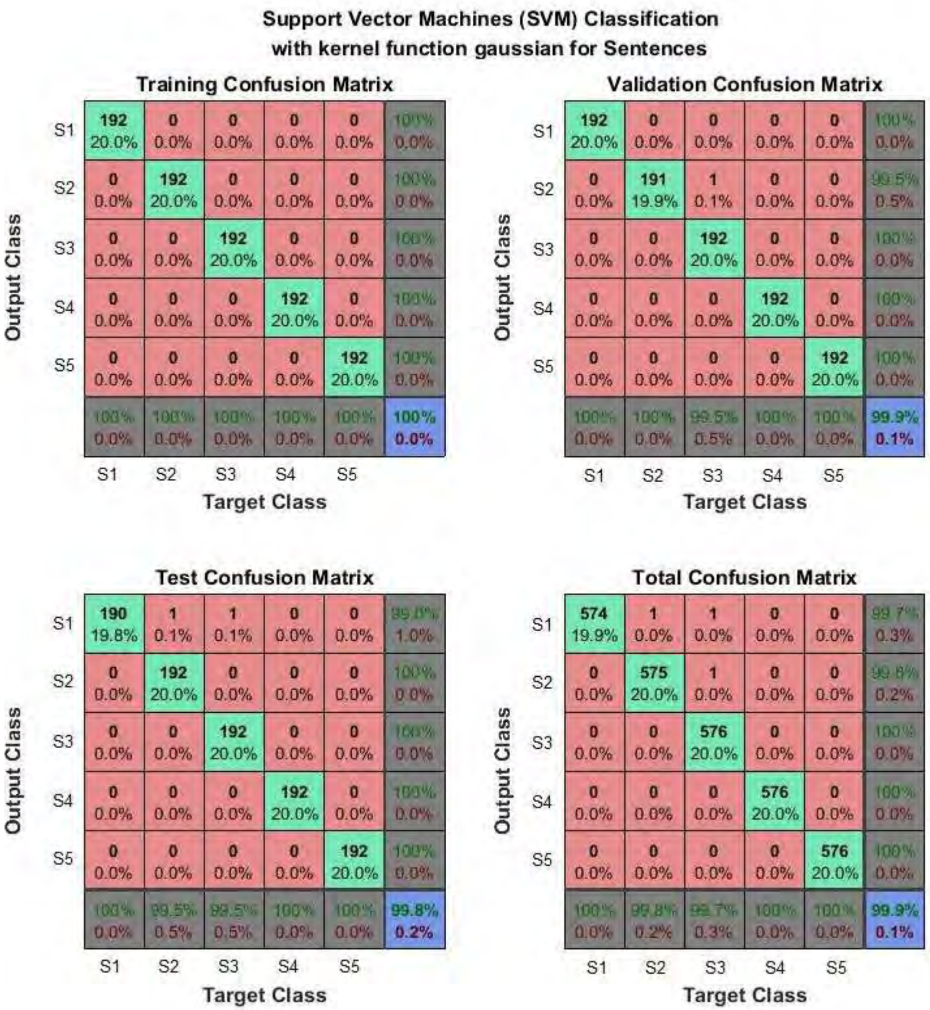
Apply discrimination (Supervised Classification)

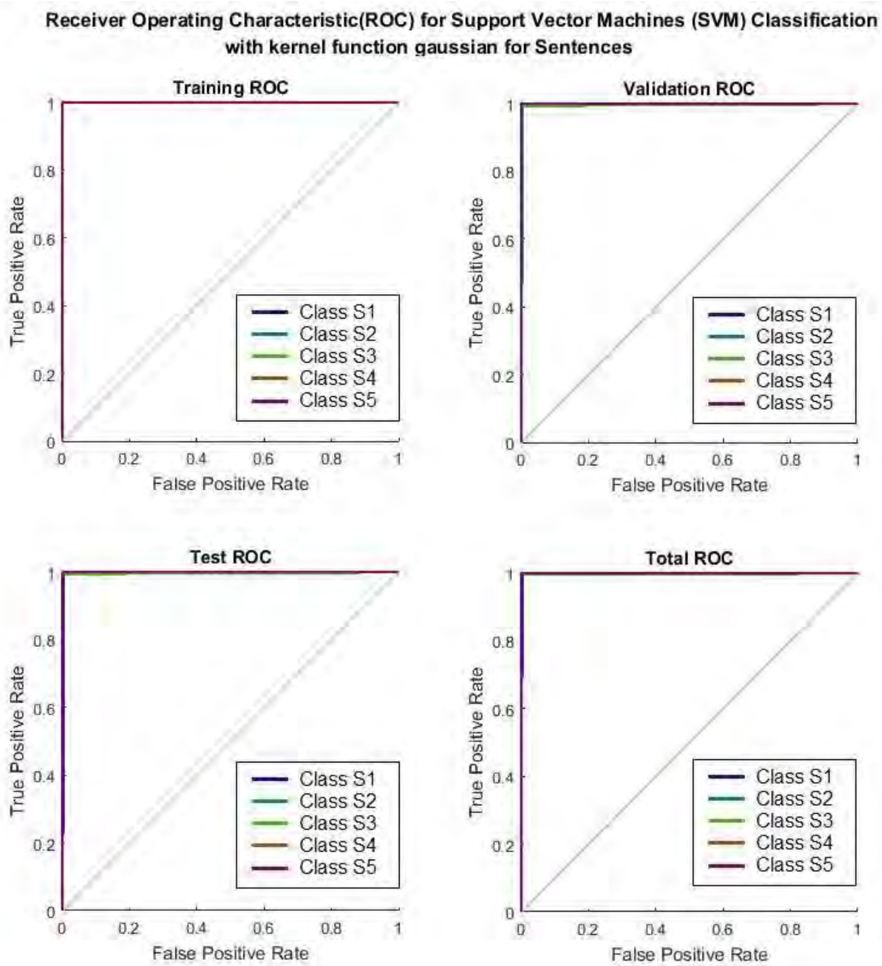
- Multiclass Support Vector Machine (SVM)

It were done many tests changing the kernel distribution functions and parameters for both tasks and the best results were for following configuration: “BoxConstraint” was 0.01; “KernelFunction” was “Gaussian”; and “Standardize” was “off”’, for both tasks.

3. Methodology, Results and Discussion

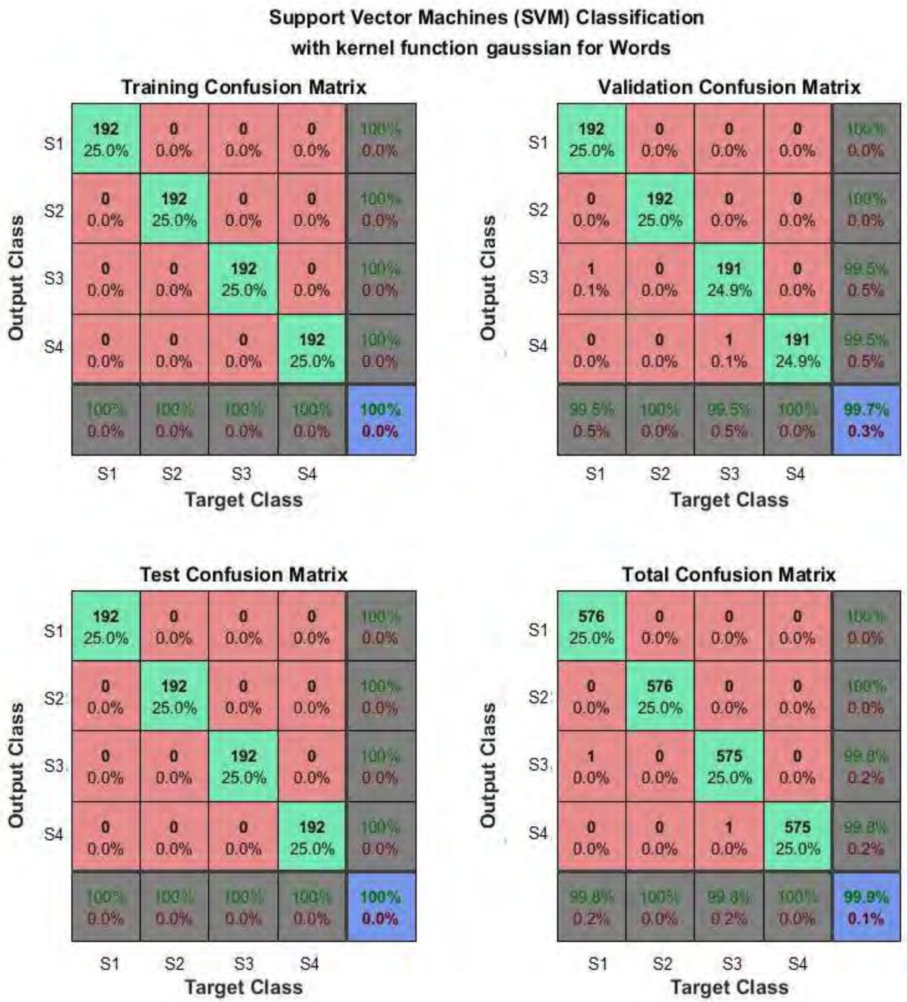
Multiclass SVM Sentences (best result)

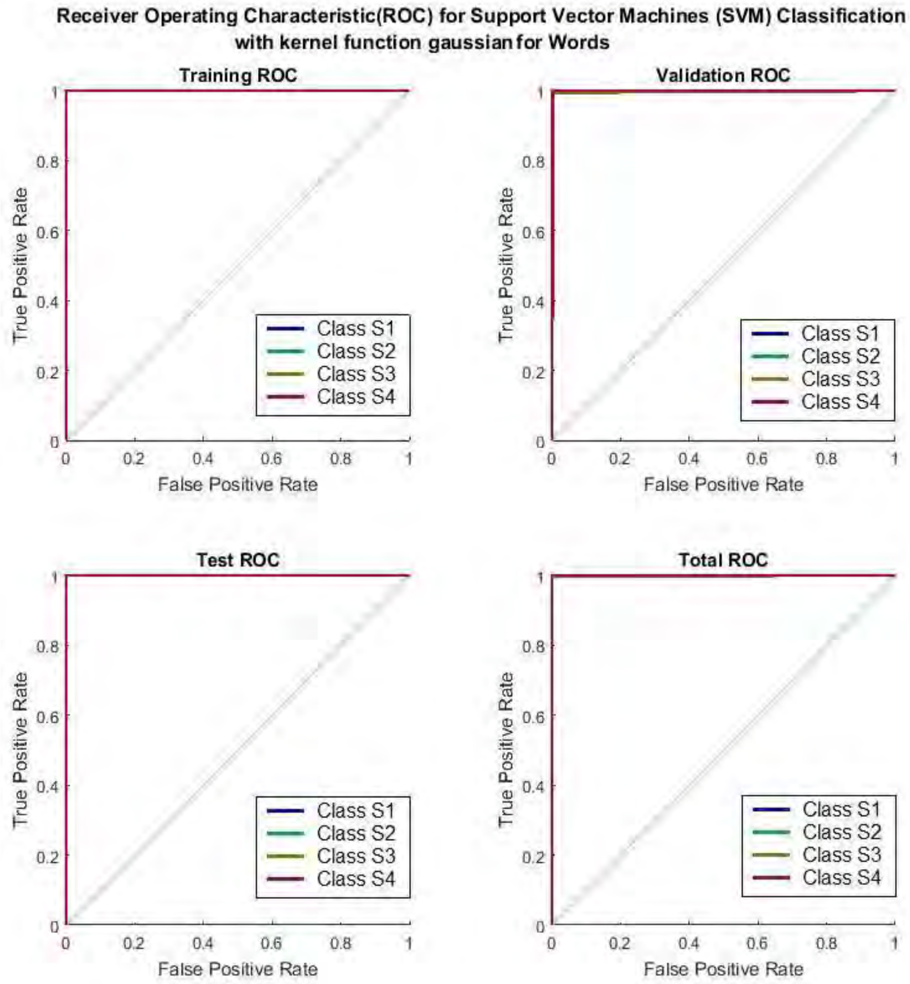




3. Methodology, Results and Discussion

Multiclass SVM Words (best result)



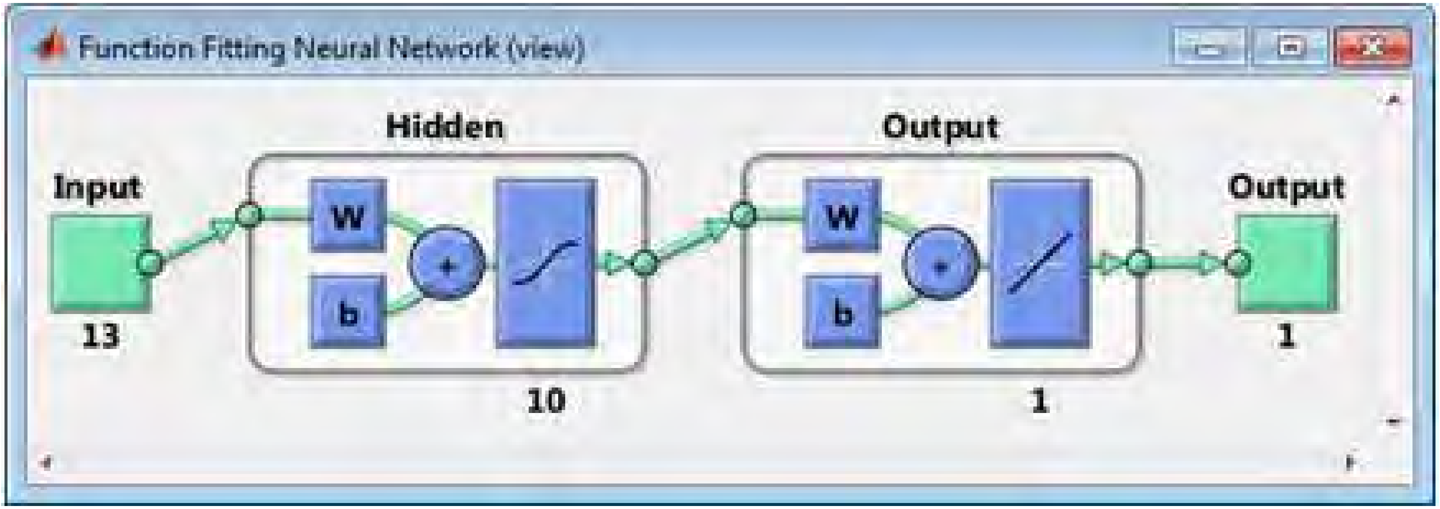


3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Neural Network

In this study, it was used the Matlab®'s Neural Network Toolbox™



A two-layer feedforward network with sigmoid hidden neurons and linear output neurons. This type of network can fit multidimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer (Matlab® site, 2016m)

Apply discrimination (Supervised Classification)

- Neural Network

The mainly parameters (MATLAB® site, 2016n) used in this study were:
a) Number of hidden layers ("hiddenLayerSize") – This property defines the number of hidden neurons of the neural network;

b) Neural Network Input-Output Processing Functions ("net.input.processFcns" and "net.output.processFcns") - This property defines the pre-processing and pos-processing functions for the neural network.

Neural Network pre-processing and pos-processing functions (MATLAB® site, 2016o).

Function	Algorithm
mapminmax	Normalize inputs/targets to fall in the range [−1, 1]
mapstd	Normalize inputs/targets to have zero mean and unity variance
processpca	Extract principal components from the input vector
fixunknowns	Process unknown inputs
removeconstantrows	Remove inputs/targets that are constant

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Neural Network

c) Setup Division of Data for Training, Validation, Testing (“net.divideFcn “) - This property defines the data division function to be used when the network is trained using a supervised algorithm, such as backpropagation.

Function	Algorithm
dividerand	Divide the data randomly (default)
divideblock	Divide the data into contiguous blocks
divideint	Divide the data using an interleaved selection
divideind	Divide the data by index

Neural Network Setup Division of Data for Training, Validation, Testing functions (MATLAB® site, 2016p).

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Neural Network

d) Divide Mode (“net.divideMode”) - This property defines the target data dimensions which to divide up when the data division function is called. Its default value is 'sample' for static networks and 'time' for dynamic networks. It may also be set to 'sampletime' to divide targets by both sample and timestep, 'all' to divide up targets by every scalar value, or 'none' to not divide up data at all (in which case all data is used for training, none for validation or testing).

e)Set up Division of Data for Training, Validation, Testing (“net.divideParam.trainRatio”, “net.divideParam.valRatio”, and “net.divideParam.testRatio”) - This property defines the size proportion in relation of the total amount of data for the Training, Validation and Test essays. As already mentioned, for all classifiers it was used 1/3 for all essays.

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Neural Network

f) Multilayer Neural Network Training Function (“net.trainFcn”) - This property defines the function that will be used to train the neural network. The options available are:“trainlm” - Levenberg-Marquardt backpropagation;
“trainbr” - Bayesian Regulation backpropagation;
“trainbfg”- BFGS quasi-Newton backpropagation;
“traincgb”- Conjugate gradient backpropagation with Powell-Beale restarts;
“traincgf”- Conjugate gradient backpropagation with Fletcher-Reeves updates;
“traincgp” - Conjugate gradient backpropagation with Polak-Ribiere updates;
“traingd” - Gradient descent backpropagation;
“traingda” - Gradient descent with adaptive lr backpropagation;
“traingdm” - Gradient descent with momentum;
“traingdx” - Gradient descent w/momentum & adaptive lr backpropagation;
“trainoss” - One step secant backpropagation;
“trainrp” - RPROP (resilient backpropagation) backpropagation;
“trainscg” - Scaled conjugate gradient backpropagation.
“trainb” - Batch training with weight & bias learning rules;
“trainc” - Cyclical order weight/bias training;
“trainr” - Random order weight/bias training;
“trains” - Sequential order weight/bias training;
“trainbu” - Unsupervised batch training with weight & bias learning rules; and
“trainru” - Unsupervised random order weight/bias training;

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

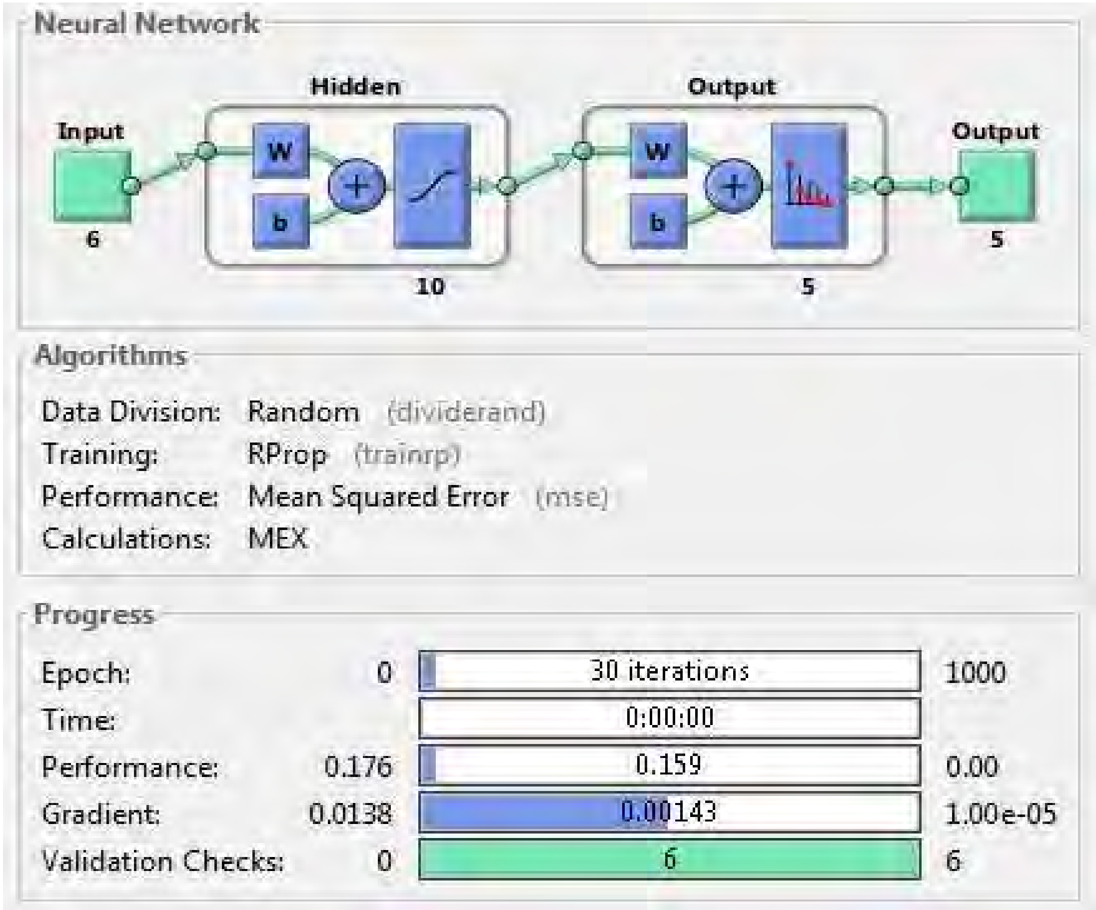
- Neural Network

g) Neural Network Performance Function (“net.performFcn”) - This property calculates a network performance given targets and outputs, with optional performance weights and other parameters. The options available are:

“crossentropy” - cross entropy function;
“mae” - Mean absolute error performance function;
“mse” - Mean squared normalized error performance function;
“sse” - Sum squared error performance function; and
“sae” - Sum absolute error performance function.

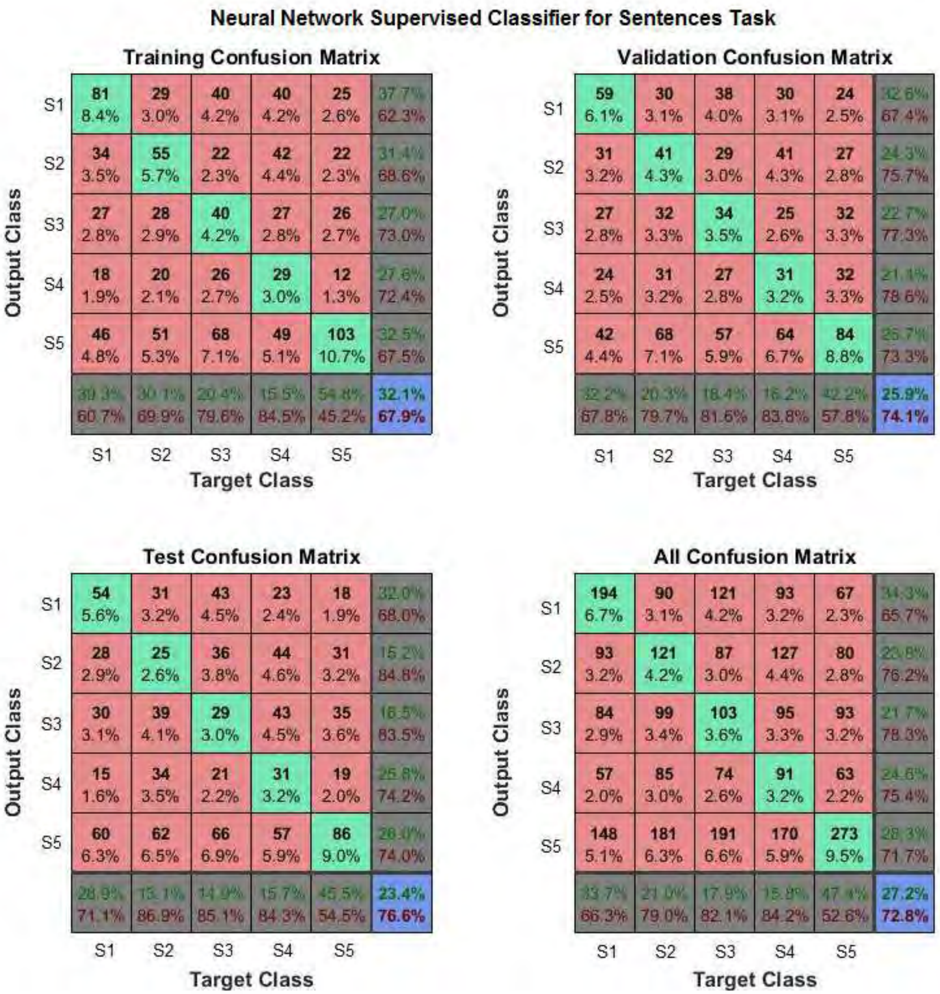
3. Methodology, Results and Discussion

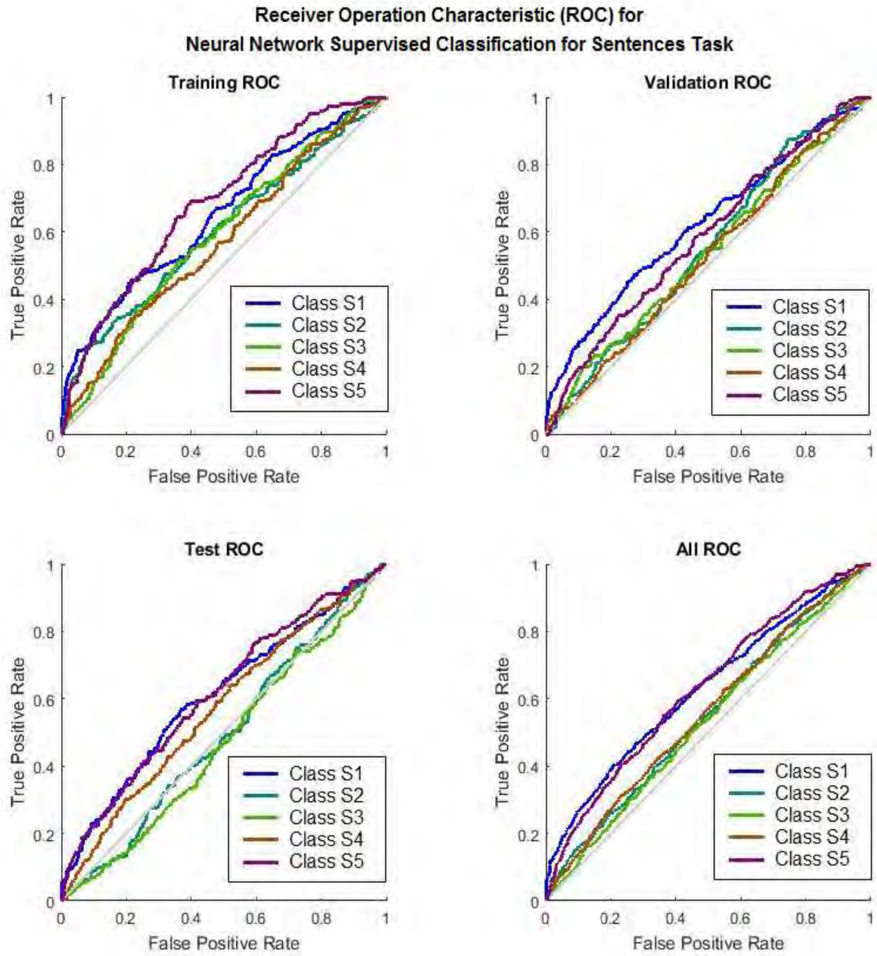
Neural Network for Sentences Task (best result)



3. Methodology, Results and Discussion

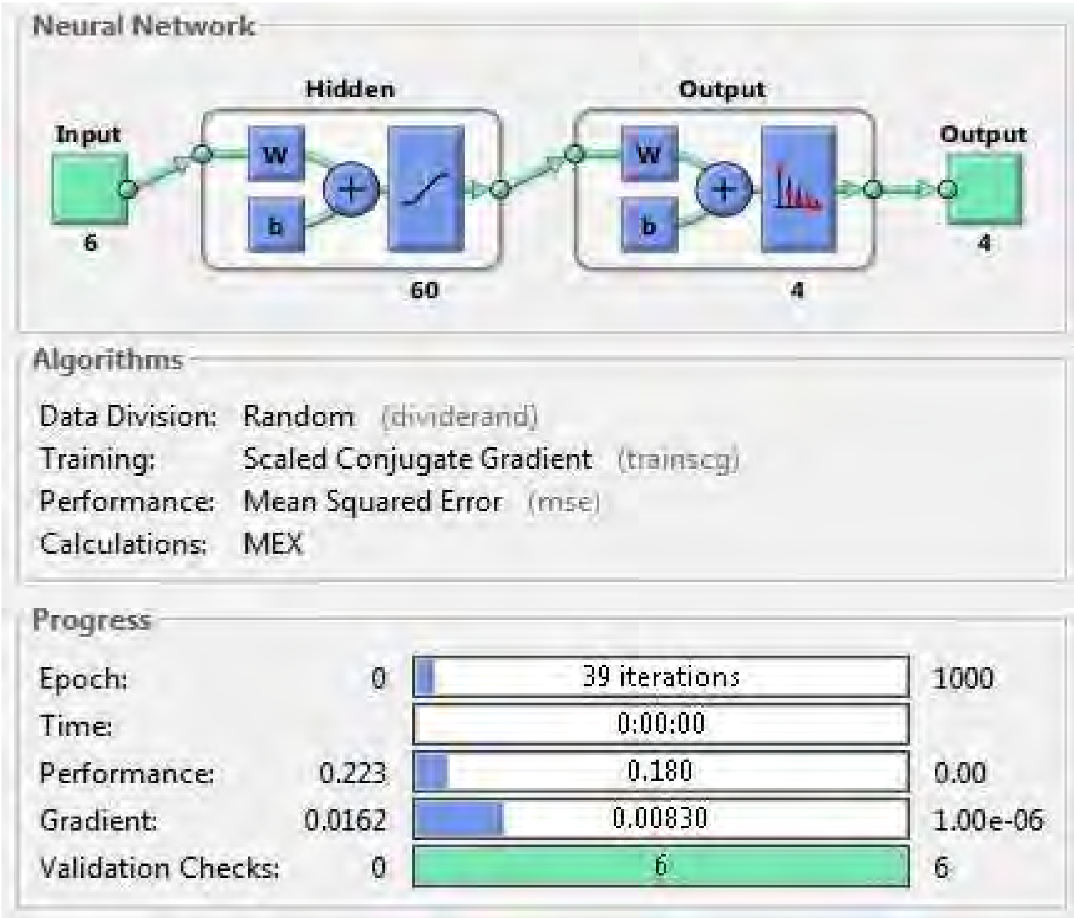
Neural Network for Sentences Task (best result)





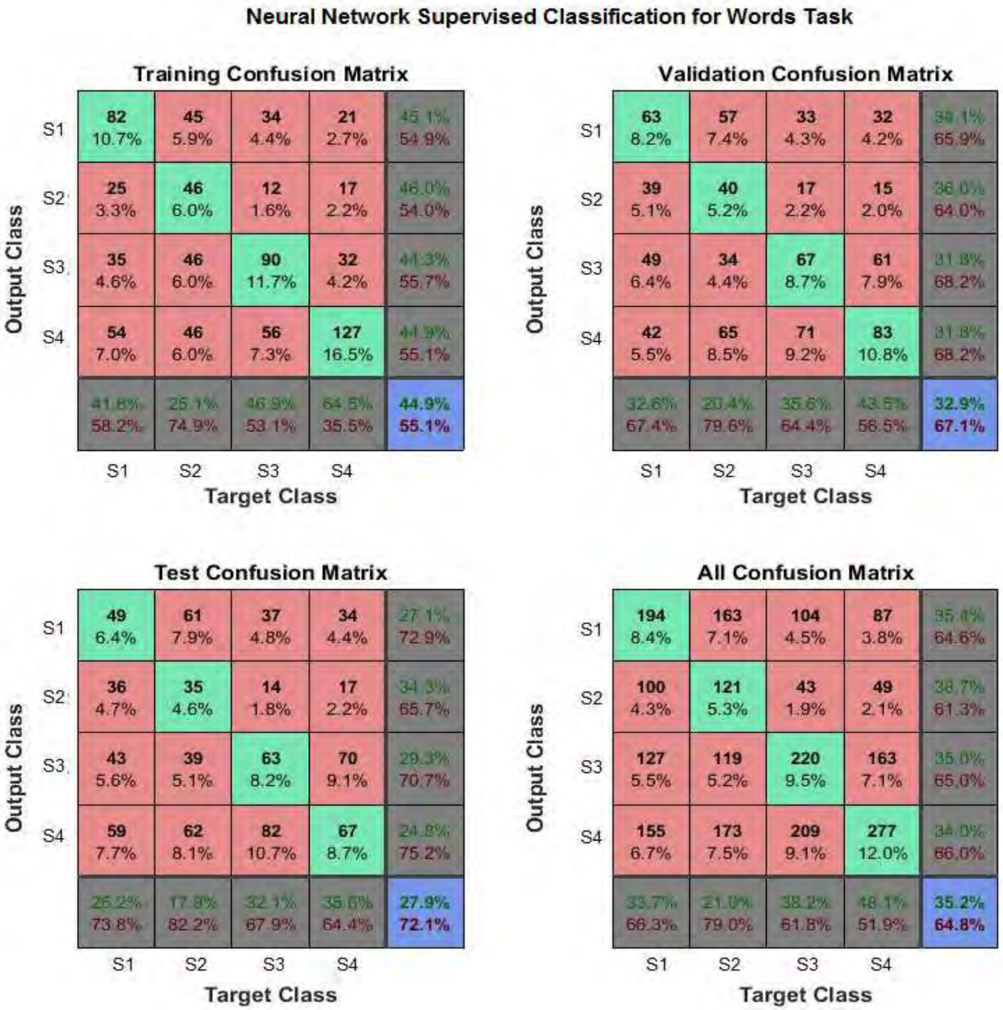
3. Methodology, Results and Discussion

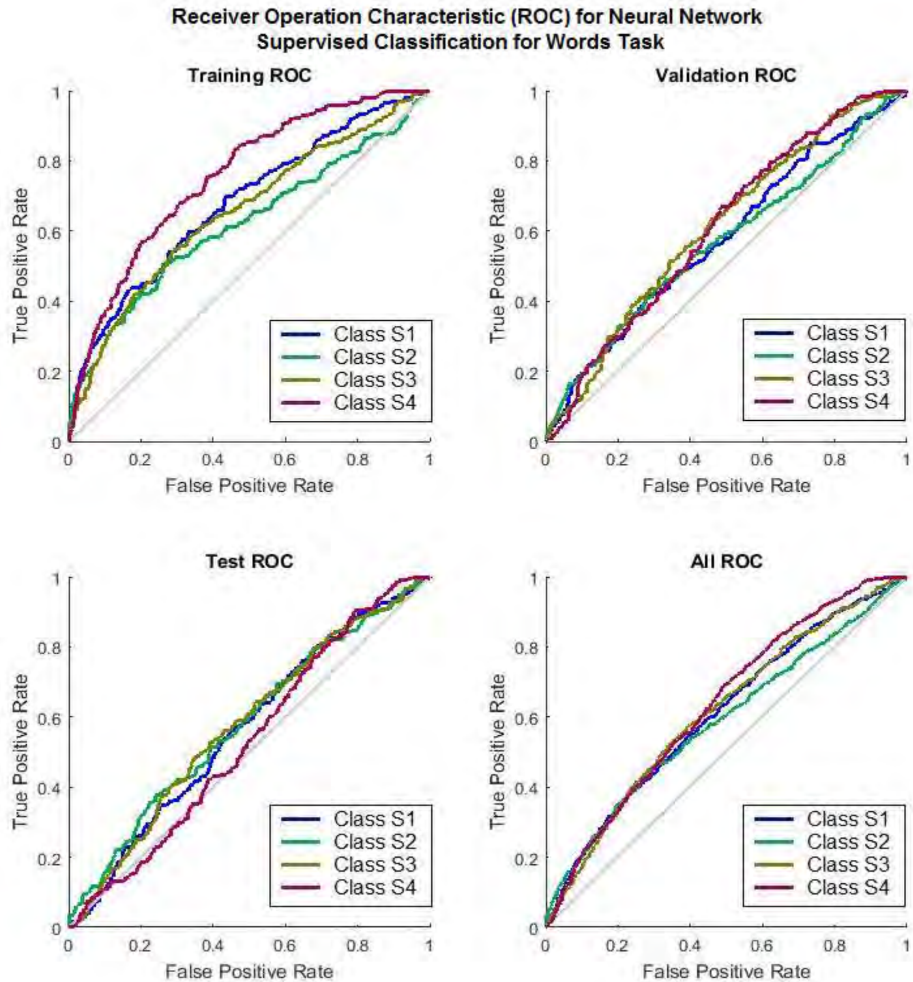
Neural Network for Words Task (best result)



3. Methodology, Results and Discussion

Neural Network for Words Task (best result)





3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Random Forest

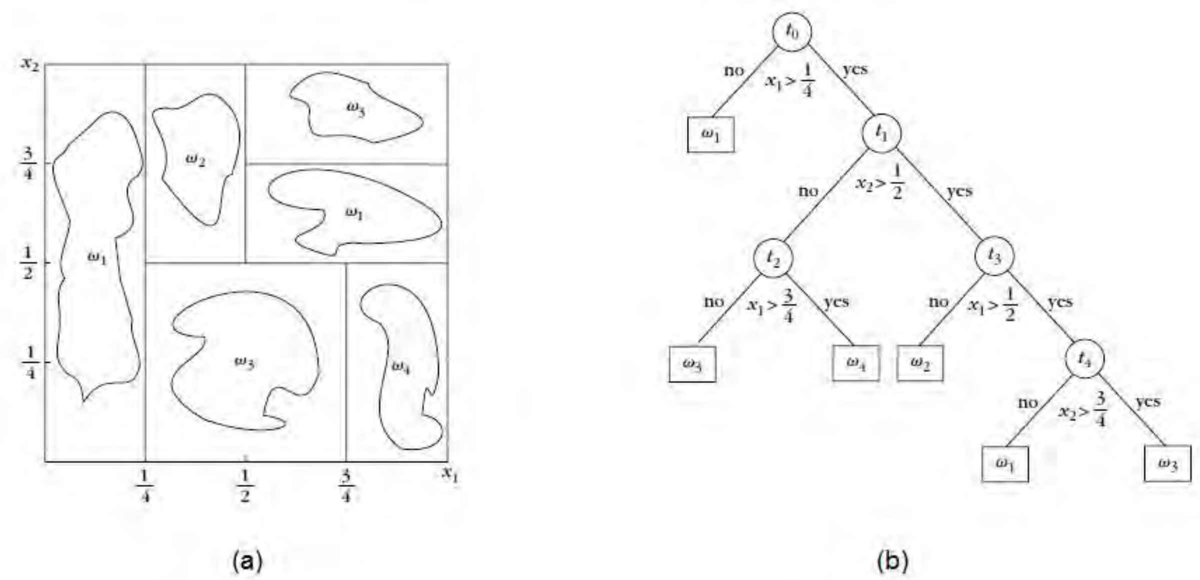
An example of this kind of approach is the classification and regression tree (CART) model that uses an expansion into indicator functions of multidimensional rectangles. In this study, the CART classifier method used is the ensemble method of Random Forest.

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Random Forest

Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.



a) Example case; and their b) decision tree (THEODORIDIS, 2009)

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

- Random Forest

In Matlab®, to create an ensemble for classification or regression is used the “fitensemble” function (Matlab® site, 2016q).

To train an ensemble using “fitensemble”, the syntax used (Matlab® site, 2016r) is:
Model = fitensemble(X, Y, method, NLearns, learners, type)

Where:
- X is the matrix of data. Each row contains one observation, and each column contains one predictor variable.
- Y is the vector of responses, with the same number of observations as the rows in X.
- “method” is a character vector, naming the type of ensemble.
- “NLearns” is the number of ensemble learning cycles, specified as a positive integer.
“- learners” is either a character vector, naming a weak learner, a weak learner template, or a cell array of such templates. Weak learners to use in the ensemble, specified as a weak-learner name, weak-learner template object, or cell array of weak-learner template objects.
- “type”: is the supervised learning type. In this study, the option is 'classification'.

3. Methodology, Results and Discussion

Apply discrimination (Supervised Classification)

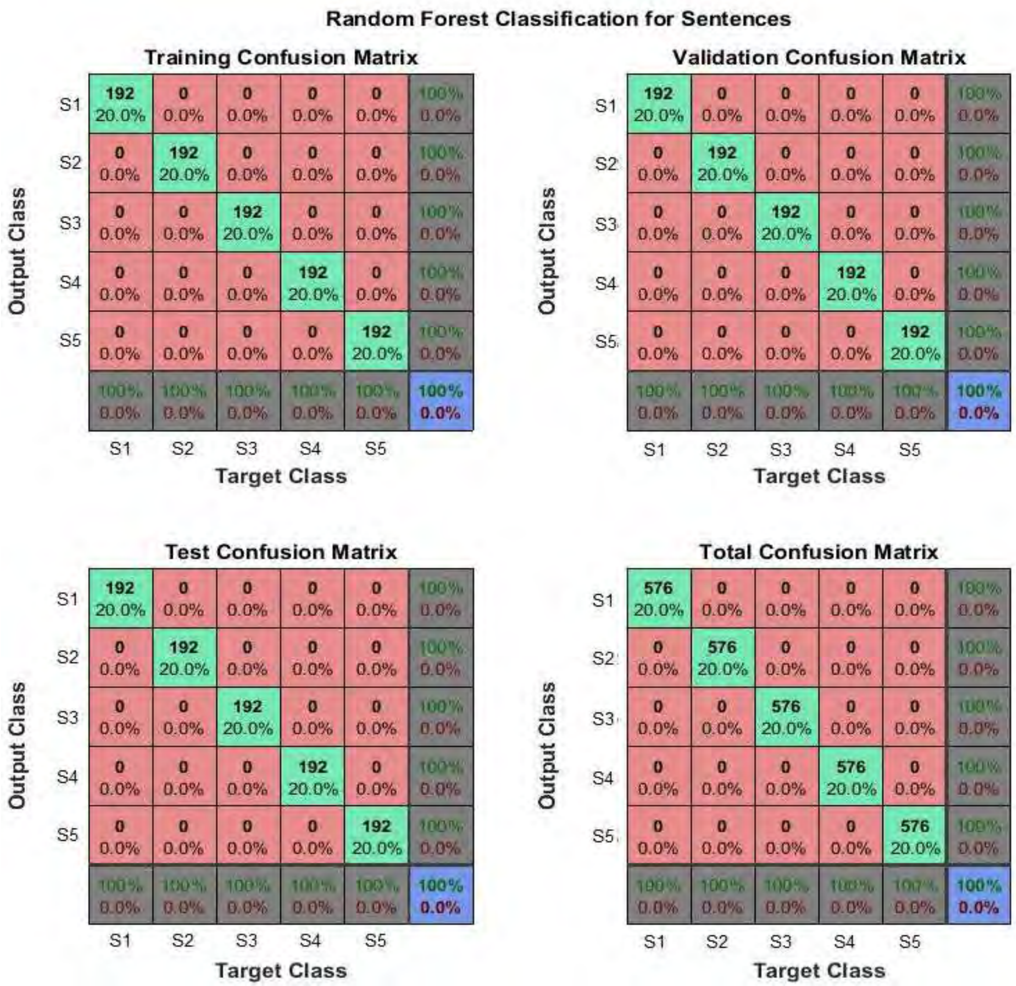
- Random Forest

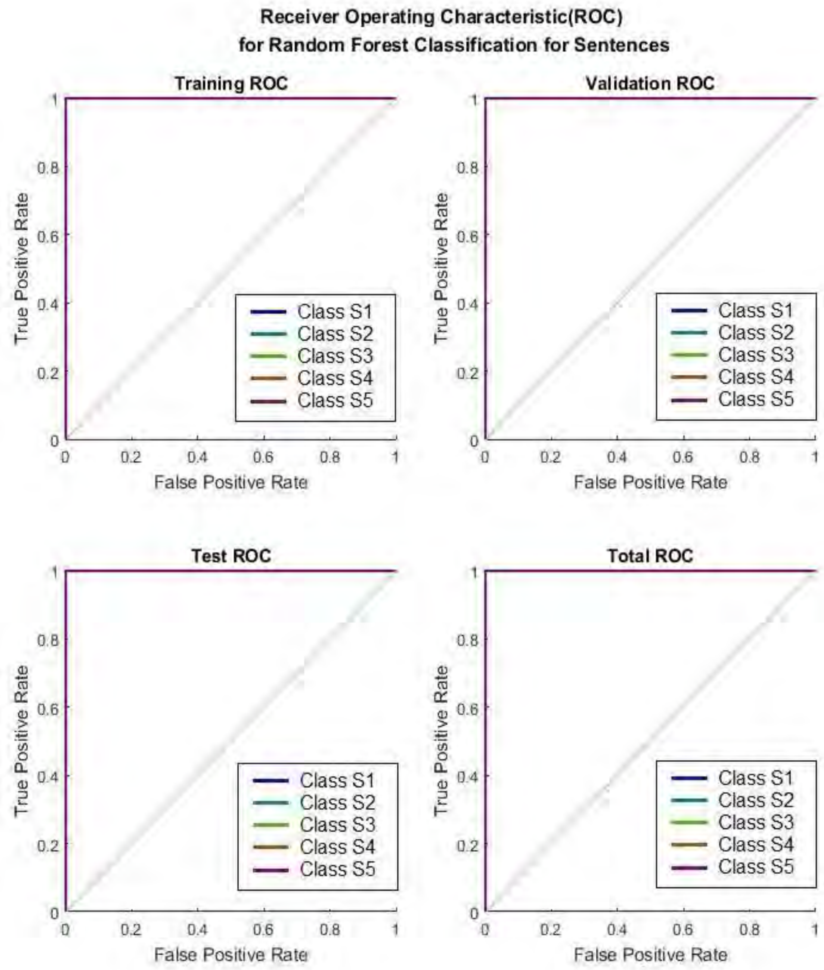
The best results were achieved using the following Matlab® function "fitensemble" parameters, for both tasks:

- 'method' (Ensemble-aggregation method): "bag";
- 'NLearn' (Number of ensemble learning cycles): 100;
- 'Learners' (Weak learners to use in ensemble): "Tree"; and
- 'Type' (Supervised learning type): 'classification'.

3. Methodology, Results and Discussion

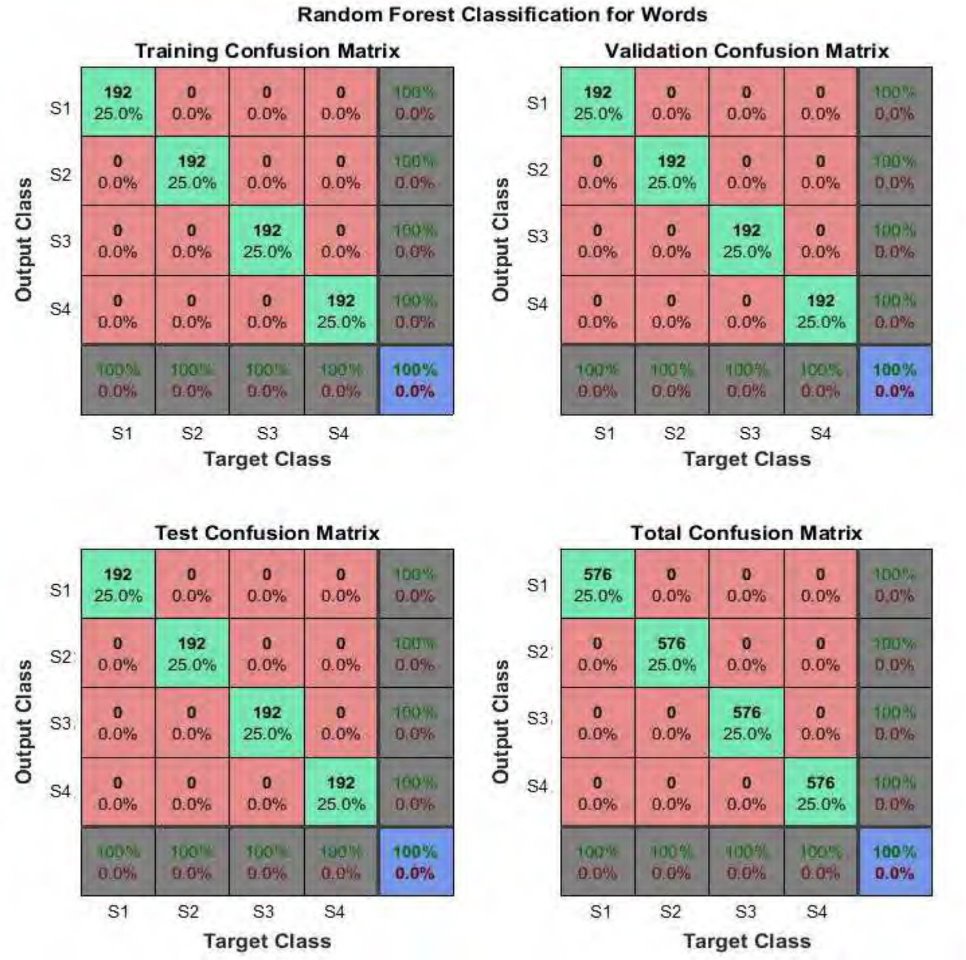
Random Forest for Sentences Task (best result)

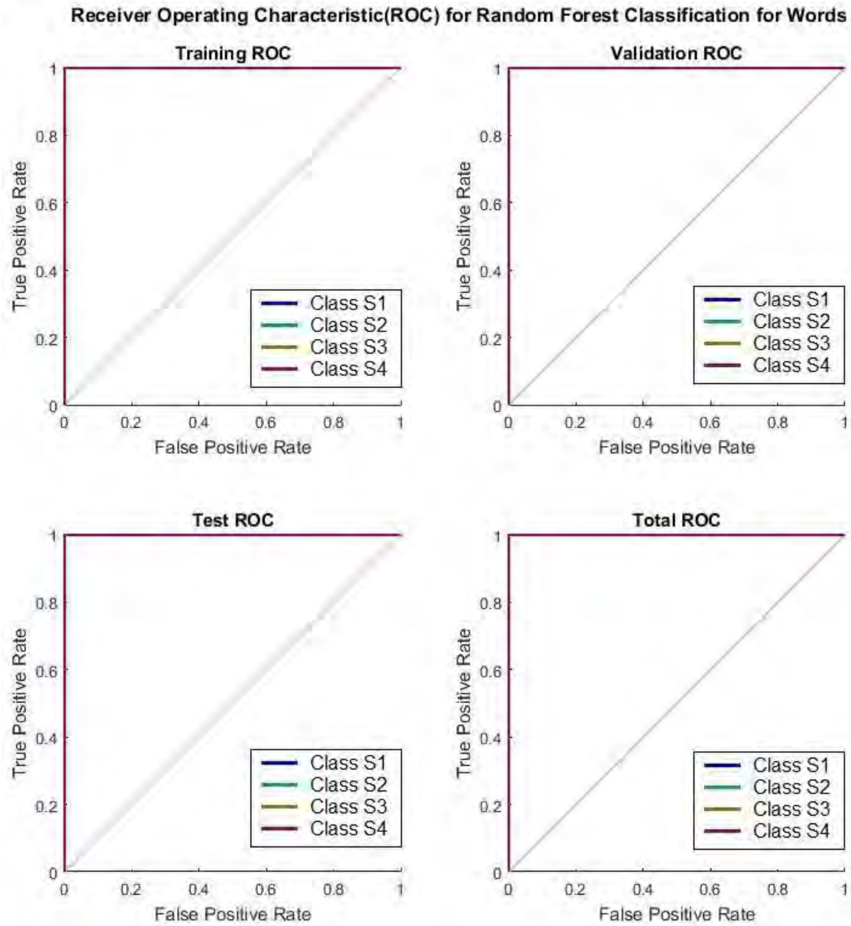




3. Methodology, Results and Discussion

Random Forest for Words Task (best result)





3. Methodology, Results and Discussion

Additional Tests

The test campaign considered all complete datasets for Sentences and Words Task. At the end of the proposed methodology, in order to deepen the investigation of the features used in classification, specifically concerning the subjects, it was performed classification tests without split, with the datasets for all supervised classifiers and for Random Forest method, for both tasks, but doing the classification campaign, without retrainning, running again the algorithms with only the data for each subject.

Since the behavior of the brain between individuals may be quite different (although the profile of the people who have passed the experiments are similar), the objective is to check if the results for individuals can be different in relation with the datasets complete.

3. Methodology, Results and Discussion

Classifiers Results for Individuals - Sentences Task

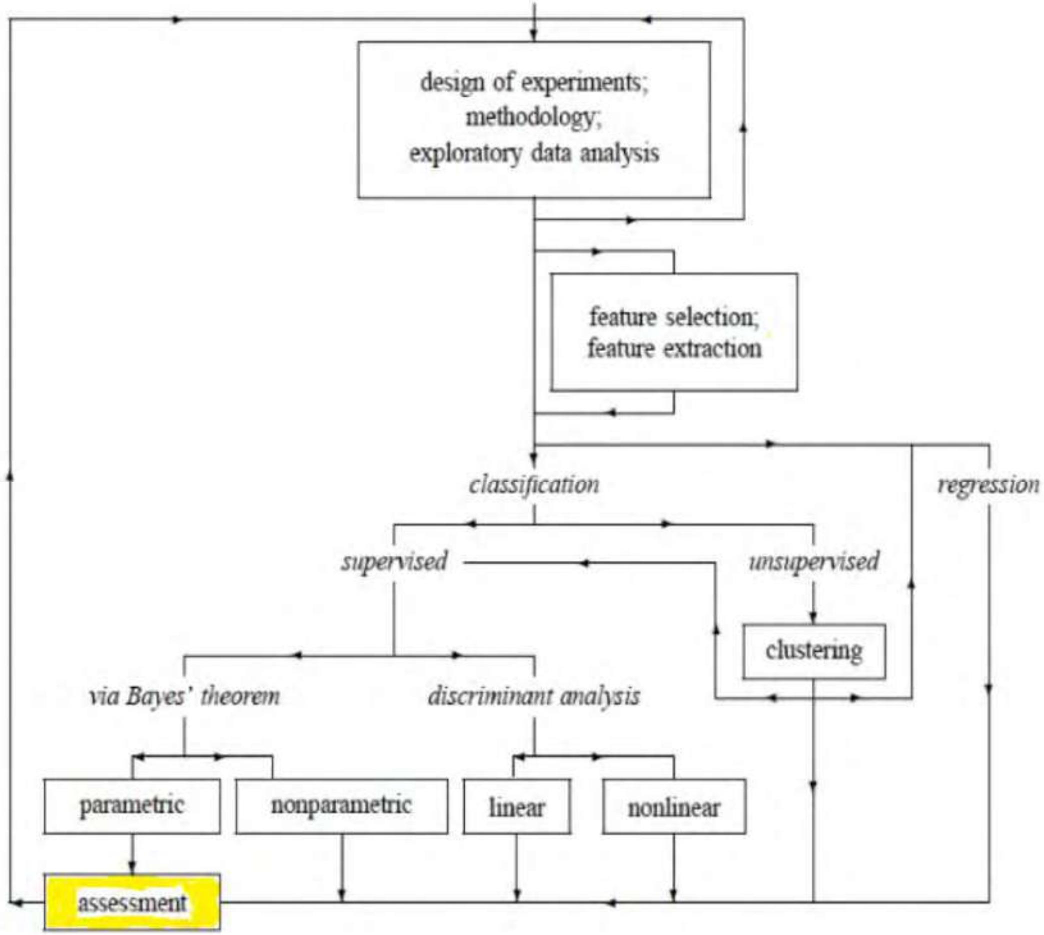
Applying Discrmination (Supervised Classifiers)						Regression method	
NAÏVE BAYES MVMN		SVM		Neural Network		Random Forest	
Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %
2	100.00%	2	100.00%	2	87,80%	2	100.00%
3	98.89%	3	99.44%	3	23,90%	3	100.00%
4	100.00%	4	99.44%	4	32,20%	4	100.00%
5	100.00%	5	100.00%	5	52,80%	5	100.00%
6	100.00%	6	100.00%	6	34,40%	6	100.00%
7	100.00%	7	100.00%	7	20,00%	7	100.00%
9	99.44%	9	100.00%	9	44,40%	9	100.00%
10	100.00%	10	100.00%	10	23,30%	10	100.00%
13	100.00%	13	100.00%	13	33,90%	13	100.00%
15	100.00%	15	100.00%	15	19,40%	15	99.44%
16	100.00%	16	100.00%	16	45,00%	16	100.00%
17	99.44%	17	100.00%	17	25,60%	17	100.00%
18	100.00%	18	100.00%	18	43,30%	18	100.00%
19	100.00%	19	100.00%	19	61,70%	19	100.00%
20	100.00%	20	100.00%	20	36,10%	20	100.00%
21	100.00%	21	100.00%	21	29,40%	21	100.00%
Classifier accuracy with complete dataset %	97.26%	Classifier accuracy with complete dataset %	99.90%	Classifier accuracy with complete dataset %	27,20%	Classifier accuracy with complete dataset %	100.00%

3. Methodology, Results and Discussion

Classifiers Results for Individuals - WordsTask

Applying Discrmination (Supervised Classifiers)						Regression method	
NAÏVE BAYES MVMN		SVM		Neural Network		Random Forest	
Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %
2	100.00%	2	100.00%	2	26,40%	2	100.00%
3	100.00%	3	99.31%	3	34,70%	3	100.00%
4	100.00%	4	99.31%	4	37,50%	4	100.00%
5	100.00%	5	100.00%	5	25,70%	5	100.00%
6	99.31%	6	100.00%	6	34,70%	6	100.00%
7	100.00%	7	100.00%	7	30,60%	7	100.00%
9	100.00%	9	100.00%	9	50,00%	9	100.00%
10	100.00%	10	100.00%	10	28,50%	10	100.00%
13	100.00%	13	99.31%	13	36,10%	13	100.00%
15	100.00%	15	100.00%	15	25,00%	15	100.00%
16	100.00%	16	100.00%	16	27,80%	16	100.00%
17	100.00%	17	100.00%	17	34,70%	17	100.00%
18	100.00%	18	100.00%	18	52,10%	18	100.00%
19	100.00%	19	100.00%	19	32,60%	19	100.00%
20	100.00%	20	100.00%	20	41,00%	20	100.00%
21	100.00%	21	100.00%	21	21,50%	21	100.00%
Classifier accuracy with complete dataset %	97,60%	Classifier accuracy with complete dataset %	99,90%	Classifier accuracy with complete dataset %	35,20%	Classifier accuracy with complete dataset %	100.00%

4. Conclusions and Final Considerations



4. Conclusions and Final Considerations

Classifier	Method	Total Accuracy for Sentences Task	Observation concerning the Parameters used	Total Accuracy for Words Task	Observation concerning the Parameters
Unsupervised	Hierarchical Clustering	21,63 %	"pdist" metric "cityblock" with a "linkage" method "average" "pdist" metric "cityblock" with a "linkage" method "centroid"	28,21%	"pdist" metric "spearman" with a "linkage" method "single"
	K-means	52,92 %	2 clusters with k-means metric "cityblock"	48,44 %	2 clusters with k-means metric "cityblock"
		19,44 %	5 clusters with k-means metric "cityblock"	23,26 %	4 clusters with k-means metric "cityblock"
		53,33 %	2 clusters with k-means metric "sqEuclidean"	32,68 %	3 clusters with k-means metric "sqEuclidean"
		18,13 %	5 clusters with k-means metric "sqEuclidean"	25,00 %	4 clusters with k-means metric "sqEuclidean"
	Gaussian Mixture Models	19,24 %	Features used: Mean Amplitude Between two fixed latencies and Peak Amplitude	24,91 %	Features used: Mean Amplitude Between two fixed latencies and Peak Amplitude
		21,18 %	Features used: Mean Amplitude Between two fixed latencies and Peak Latency	24,78 %	Features used: Mean Amplitude Between two fixed latencies and Peak Latency
		19,24 %	Features used: Peak Amplitude and Peak Latency	24,39 %	Features used: Peak Amplitude and Peak Latency

4. Conclusions and Final Considerations

Classifier	Method	Total Accuracy for Sentences Task	Observation concerning the Parameters used	Total Accuracy for Words Task	Observation concerning the Parameters
Supervised	Naïve Bayes	33,7 %	Distribuiton function "kernel"	39,2 %	Distribuiton function "kernel"
		97,3 %	Distribuiton function "MVMN"	97,6 %	Distribuiton function "MVMN"
	Multiclass Support Vector Machine	99,9 %	"BoxConstraint":0.01; "KernelFunction":"Gaussian"; and "Standardize":"off"'	99,9 %	"BoxConstraint":0.01; "KernelFunction":"Gaussian"; and "Standardize":"off"'

	Neural Network	27,2 %	a) Number of hidden layers ("hiddenLayerSize"): 10
			b) Neural Network Input Processing Function ("net.input.processFcns"): 'removeconstantrows','mapstd'
			c) Neural Network Output Processing Function("net.output.processFcns"): 'removeconstantrows','mapstd'
			d) Setup Division of Data for Training, Validation, Testing ("net.divideFcn"): 'dividerand'
			e) Train Ratio ("net.divideParam.trainRatio") = 1/3;
			f) Validation Ratio ("net.divideParam.valRatio") = 1/3;
			g) Test Ratio ("net.divideParam.testRatio") = 1/3;
			h) Divide Mode ("net.divideMode"): 'sample'
			i) Multilayer Neural Network Training Function ("net.trainFcn"):'trainrp'
			j) Neural Network Performance Function ("net.performFcn"):'mse'
				35,2 %	a) Number of hidden layers ("hiddenLayerSize"): 60
					b) Neural Network Input Processing Function ("net.input.processFcns"): 'removeconstantrows','mapstd'
					c) Neural Network Output Processing Function("net.output.processFcns"): 'removeconstantrows','mapstd'
					d) Setup Division of Data for Training, Validation, Testing ("net.divideFcn"): 'dividerand'
					e) Train Ratio ("net.divideParam.trainRatio") = 1/3;
					f) Validation Ratio ("net.divideParam.valRatio") = 1/3;
					g) Test Ratio ("net.divideParam.testRatio") = 1/3;
					h) Divide Mode ("net.divideMode"): 'sample'
					i) Multilayer Neural Network Training Function ("net.trainFcn"):'trainscg'
					j) Neural Network Performance Function ("net.performFcn"):'mse'

4. Conclusions and Final Considerations

Classifier	Method	Total Accuracy for Sentences Task	Observation concerning the Parameters used	Total Accuracy for Words Task	Observation concerning the Parameters
Regression	Random Forest	100,0 %	a) 'method' (Ensemble-aggregation method): “bag”; b) 'NLearn' (Number of ensemble learning cycles): 100; c) 'Learners' (Weak learners to use in ensemble): “Tree”; and d) 'Type' (Supervised learning type): 'classification'.	100,0 %	a) 'method' (Ensemble-aggregation method): “bag”; b) 'NLearn' (Number of ensemble learning cycles): 100; c) 'Learners' (Weak learners to use in ensemble): “Tree”; and d) 'Type' (Supervised learning type): 'classification'.

4. Conclusions and Final Considerations

The objective of this work is to check if applying the pattern recognition methodology proposed by Webb (2002) in the ERP results from the Soto (2014) data experiment, is possible to obtain good classification paradigms was considered as achieved.

The software tools EEGLAB®, ERPLAB® and Matlab® perform properly the extraction and treatment of the focused EEG data and the pattern recognition algorithms proposed. As demonstrated in this thesis simulations and results, the "clustering and unsupervised classification" is not appropriate for the task, on the other hand, the Webb (2002) proposed methodology allow us to obtain a good results to support the goal of this work with excellent results for supervised classification and regression method.

The regression method Random Forest was the best supervised classifier method for these data sets which a total accuracy of 100%. Another good results are achieved with the discrimination supervised classifiers SVM and Naïve Bayes, with total accuracies higher than 96%. These results also indicated that for these ERP datasets, for both Sentences and Words Tasks, non-linear approaches were more suitable to classify the data from Soto (2014) experiment configuration. This result is valid both for each subject and for group of subjects.

4. Conclusions and Final Considerations

Even with these good results, is suitable continue the studies in relation to the analysis of the classifiers proposed with the separation of the other features, especially the ROI and the ERP time range features. Maybe, they cannot only allow a more specific way to classify the data, but also could indicate which ERP features can be more influent in the classification process.

Other methods of classification and, especially, for the clustering and unsupervised classification shall be considered in order to promote the advance in this dataset study in not only pattern recognition, but also their use as a possible method for neuro linguistics and medicine areas.

5. Bibliography

BREIMAN, Leo. **Random Forests**. Machine Learning. October 2001, Volume 45, Issue 1, pp 5–32. doi:10.1023/A:1010933404324.

CONG, Fengyu, RISTANIEMI, Tapani, LYYTINEN, Heikki, **Advanced Signal Processi on Brain Event-Related Potentials: Filtering ERPs in Time, Frequency and Spa Domains Sequentially and Simultaneously**, Singapore, Ed. World scientific Publishir 2015, ISBN 978-981-4623-08-7.

EEGLAB® site. **EEGLAB® Tutorial**. Available on: <http://sccn.ucsd.edu/wiki/Getting_Started>, accessed in April, 15th,2016.

ERPLAB® site. **ERPInfo-ERPLAB® Toolbox**. Available on:<http://erpinfo.org/ERPLAB®>, consulted on April, 15th,2016.

GESUALDI, Aline da Rocha; FRANÇA, Aniela Improta. **Event-related brain potentials (ERP): an overview**. Revista Linguística / Revista do Programa de Pós-Graduação em Linguística da Universidade Federal do Rio de Janeiro. Volume 7, número 2, dezembro de 2011. ISSN 1808-835X 1. [http://www.letras.ufrj.br/poslinguistica/revistalinguistica]

HANDY, T. C. **Event Related Potentials: A Methods Handbook**. Cambridge, MA: Bradford/MIT Press., 2005.

HASTIE, T., TIBSHIRANI, R., and FRIEDMAN, J. **The Elements of Statistical Learning.** Second Edition. NY: Springer, 2008.

KAMEL, Nidal, MALIK, Aamir Saeed, **EEG/ERP analysis: methods and apllications**, Boc Raton/FL, USA, Ed. CRC Press, 2015. ISBN 978-1-4822-2469-6.

LOUPPE, Gilles. **Understanding Random Forests: From Theory to Practice.** Doctoral thesis. Université de Liège. 2014.

LUCK, Steven J., **An Introduction to the Event-Related Potential Technique**, Massachusetts Institute of Technology MIT Press books, 2nd Ed., 2014, ISBN 978-0-262-52585-5

MATLAB® site. **Introduction to Cluster Analysis**. Available on: <https://www.mathworks.com/help/stats/hierarchical-clustering.html l>, consulted on April, 15th, 2016a.

MATLAB® site. **pdist - Pairwise distance between pairs of objects**. Available on: <https://www.mathworks.com/help/stats/pdist.html>, consulted on April, 15th, 2016b.

MATLAB® site. **linkage - Agglomerative hierarchical cluster tree.** Available on: <https://www.mathworks.com/help/stats/linkage.html>, consulted on April, 15th, 2016c.

MATLAB® site. **kMeans Clustering**. Available on: <https://www.mathworks.com/help/stats/kmeansclustering.html>, consulted on April, 15th, 2016d.

MATLAB® site. kmeans - **k-means clustering.** Available on: <https://www.mathworks.com/help/stats/kmeans.html>, consulted on April, 15th, 2016e.

MATLAB® site. Clustering Using Gaussian Mixture Models. Available on: <https://www.mathworks.com/help/stats/clusteringusinggaussianmixturemodels.html>, consulted on April, 15th, 2016f.

MATLAB® site. **Cluster Data from Mixture of Gaussian Distributions**. Available on: <https://www.mathworks.com/help/stats/clusterdatafrommixtureofgaussiandistributions.html>, consulted on April, 15th, 2016g.

MATLAB® site. **Naive Bayes Classification**. Available on: <https://www.mathworks.com/help/stats/naive-bayes-classification.html>, consulted on April, 15th, 2016h.

MATLAB® site. **Train multiclass naive Bayes model**. Available on: <https://www.mathworks.com/help/stats/fitcnb.html>, consulted on April, 15th, 2016i.

MATLAB® site. **Support Vector Machines for Binary Classification**. Available on: <https://www.mathworks.com/help/stats/supportvectormachinesforbinaryclassification.html>, consulted on April, 15th, 2016j.

MATLAB® site. **ClassificationECOC class - Multiclass model for support vector machines or other classifiers**. Available on: <https://www.mathworks.com/help/stats/classificationecocclass.html>, consulted on April, 15th, 2016k.

MATLAB® site. **templateSVM - Support vector machine template**. Available on: <https://www.mathworks.com/help/stats/templatesvm.html>, consulted on April, 15th, 2016l.

MATLAB® site. **Neural Network Toolbox**. Available on: <https://www.mathworks.com/products/neuralnetwork.html>, consulted on April, 15th, 2016m.

MATLAB® site. **Neural Network Object Properties**. Available on: <https://www.mathworks.com/help/nnet/ug/neural-network-object-properties.html>, consulted on April, 15th, 2016n.

MATLAB® site. **Choose Neural Network Input-Output Processing Functions**. Available on: <https://www.mathworks.com/help/nnet/ug/choose-neural-network-input-output-processing-functions.html>, consulted on April, 15th, 2016o.

MATLAB® site. **Divide Data for Optimal Neural Network Training**. Available on: <https://www.mathworks.com/help/nnet/ug/divide-data-for-optimal-neural-network-training.html>, consulted on April, 15th, 2016p.

MATLAB® site. **Ensemble Methods**. Available on: <https://www.mathworks.com/help/stats/ensemblemethods.html>, consulted on April, 15th, 2016q.

MATLAB® site. **Fit ensemble of learners for classification and regression**. Available on: <https://www.mathworks.com/help/stats/fitensemble.html>, consulted on April, 15th, 2016r.

ROKACH, Lior, and ODED, Maimon. **Clustering methods. Data mining and knowledge discovery handbook**. Springer US, 2005. 321-352.

SOTO, Marije, **ERP and fMRI Evidence of Compositional Differences between Linguistic Computations for Words and Sentences**. Marije Soto - Rio de Janeiro:UFRJ/Faculdade de Letras, 2014.

THEODORIDIS, Segios, KOUTROMBAS, Konstantinos, **Pattern Recogniton**, 4th Edition, Academic Press, Elsevier Inc, .2009, ISBN: 978-1-59749-272-0

WEBB, Andrew R., **Statistical Pattern Recognition**, Second Edition. John Wiley & Sons, Ltd. 2012. ISBNs: 0-470-84513-9 (HB); 0-470-84514-7 (PB)

WIKIPEDIA, **Components of ERP**. Available on: <https://en.wikipedia.org/wiki/Event-related_potential#/media/File:ComponentsofERP.svg>, consulted on April, 15th,2016.