International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 37-44
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Learning Experiences Using Neural Networks and Support Vector Machine (SVM)

Soumya ARACH, Pr. Halima BOUDEN,
Laboratory of Computer Science, Operational Research and Applied Statistics.
University Abdelmalek Essaadi.
Tétouan, Morocco.
soumya.arach@gmail.com , bouden.halima@gmail.com

ABSTRACT

This article is part of the global data mining framework, it addresses the theme of learning and classification, to identify the classes to which objects belong from using some descriptive parameters. They are particularly suited to the problem of automated decision-making. In this article we tried to implement three learning techniques, the Support Vector Machine (SVM), the Neural Networks and the Decision Trees.
This application study aims to compare the results of these three techniques in terms of respecting the performance of the classification used for the contained objects in the data set ''IRIS'' based on the confusion matrix generated by the software weka, which is the tool used to carry out these learning experiences.

KEYWORDS

Data Mining, Machine learning, classification, Support Vector Machine (SVM), Decision trees, Neural Networks.

I. INTRODUCTION:

Classification methods aim to identify the classes to which objects belong of the basis of some descriptive parameters. They apply to many human activities and are particularly suited to the problem of automated decision-making. The classification procedure will be extracted automatically from a set of examples. An example is the description of a case with the corresponding classification. A learning system must, then, from this set of examples, extract a classification procedure, it is a question of extracting a general rule from the observed data.
The procedure generated must correctly classify examples of sample and have good predictive power to correctly classify new descriptions.

The methods used for classification are many, include: the method of Support vector machine (SVM), Neural Networks, decision trees, etc. We present in the rest of this article a study of three techniques SVM, Neural networks and decision trees. These methods have proven their effectiveness in many application areas such as image processing, text categorization and medical diagnostics.

II. MACHINE LEARNING

Machine learning refers to the development, analysis and implementation of methods that allow a machine to evolve through a process of learning, and so perform tasks that are difficult or impossible to fill by more conventional algorithmic means. [4]
Its goal is to automatically extract and exploit this information in a data set.
The learning algorithms can be categorized according to the type of learning they employ: Supervised learning, unsupervised learning and reinforcement.

Unsupervised learning is a type of machine learning algorithm used to draw conclusions from input data of compounds datasets without categorizing responses.
The unsupervised learning method is the most common data partitioning, which is used to perform an exploratory analysis of data to find hidden patterns or clusters in the data. The clusters are designed by means of a similarity measure defined by metrics such as Euclidean distance or probabilistic distance. [5]

Supervised learning: needless to mention here the well-known regression techniques. The most typical method of data mining is certainly that decision trees: to predict a response Y, either numerical or qualitative, it initially looks for the best score of all data (usually two subsets) after a score performed on predictors and iterating in each of the subsets: the exponential growth of the tree is controlled by cost-complexity type of stop criteria as well as the validation data use that eliminate irrelevant branches. This technique results in very readable decision rules, hence its success, and prioritizes explanatory factors. In contrast in terms of readability, data mining software often offer highly nonlinear methods such as neural networks, support vector machines (SVM), decision trees, etc ... that we are going to apply in this third party to know which of these three methods give the best classification for a set of data. [2] [9]

1. SUPPORT VECTOR MACHINES (SVM)

Support Vector Machines (SVM), also called wide margin separators are supervised learning techniques designed to solve classification problems. Support Vector Machines the concepts relating to the theory of statistical learning and the theory of boundaries of Vapnik and Chervonenkis [16]. The intuitive justification of this method is: if the learning sample is linear separately, it seems natural to separate the elements of the two classes of so that they are as far as possible from the chosen frontier. These famous machines were invented in 1992 by Boser and al, but their denomination by SVM appeared only in 1995 with Cortes and al. [17]

2. NEURAL NETWORKS

Historically the inspiration for neural networks originated however, of the desire to create sophisticated, even intelligent, artificial systems, able to execute operations similar to those performed by the human brain routinely, and to try to improve understanding of the brain.
Most neural networks have a certain capacity for learning, means that they learn from examples. The network can then to be able to generalize that is to say to produce correct results on new cases that had not been presented to him during the during the learning process... [18]

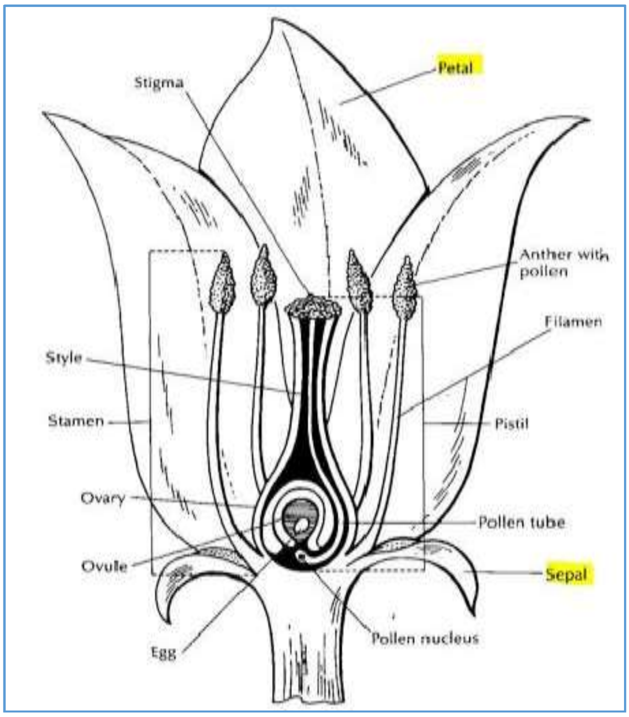
3. DECISION TREES

A decision tree is a diagram representing the possible outcomes of a series of interconnected choices. It allows a person or organization to evaluate different actions based on their cost, likelihood and benefits. It can be used to feed an informal discussion or to generate an algorithm that determines the best choice mathematically. A decision tree usually begins with a node from which several possible outcomes result. Each of these results leads to other nodes, from which emanate other possibilities. The pattern thus obtained is reminiscent of the shape of a tree.

III. APPLICATION STUDY:

The work involves testing different learning systems namely neural networks (PMC), Support Vector Machine (SVM)(SMO) and decision trees (J48) on some appropriate databases and examine how does the performance (rate error, confusion matrix, …)

1. DESCRIPTION OF THE DATABASE



IRIS contains the famous series Fisher iris data. The dataset includes measurements of 150 samples of flowers of all three species of

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 37-44
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

flowers. Setosa Iris, Iris Virginica, and Iris versicolor [8]

Four characteristics (assigned) were measured for each sample:

- The length of the flower sepals
- The width of sepal flowers
- The length of the flower petal
- The width of flower petal

All 150 samples from the data of the iris Fisher are stored in a single table called measures:
The four columns correspond to the four types of measures: the length of the sepals, width of sepals, petals length and width of the petals, respectively.

- The first **50** rows contain data for Iris Setosa
- The **50** second lines contain data for Iris Virginica
- The **50** third lines contain data for Iris versicolor

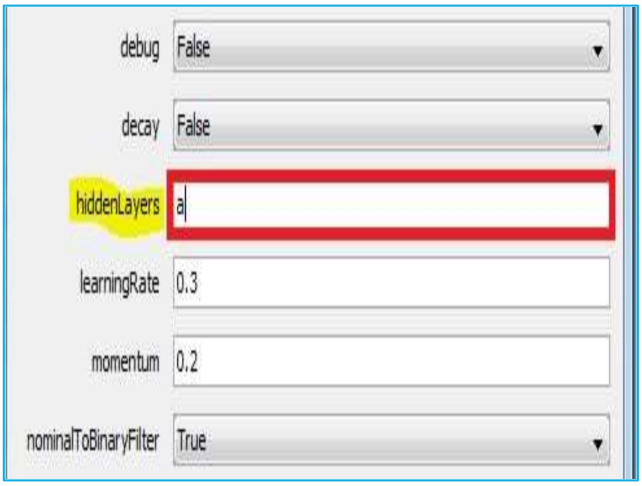
To test these learning systems namely neural networks (PMC), **SVM (SMO)** and decision trees **(J48**), we used the **WEKA** software.
The data file format is the format Weka '.arff.
Once given the game is open we will be providing the following information:

– the number of data and the number of attributes per data,
– the name of each attribute,
– What is the class attribute,
– the number of observed values by attribute,
– the number of missing data for each attribute
– information distribution of attribute values,
– the distribution of each attribute,

2. IMPLEMENTATION OF ALGORITHMS:

2.1 Neural Networks

On the WEKA software [7] we ran the algorithm Neural Networks by changing each time the number of neurons in the hidden layer in Weka changing the Hidden layer attribute that describes the number and size of hidden layers



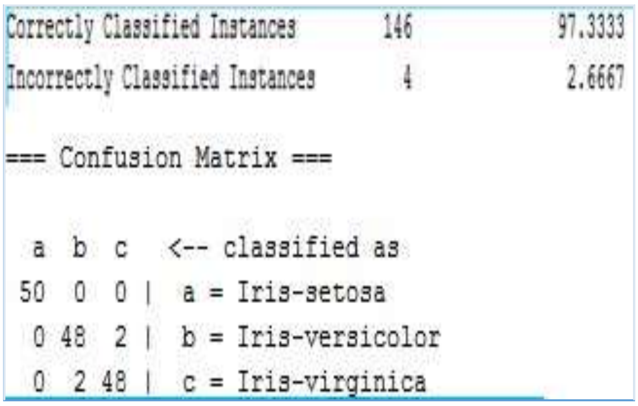
-Or special values defining a single hidden layer:

a: (number of attributes + number of classes) / 2 t: number of attributes + number of classes
o: number of classes
i: number of attributes

When we are running the neural network algorithm with the first value:

<u>a = (number of attributes + number of classes) /</u> 2)

we obtained the following results:



```
Correctly Classified Instances        146        97.3333
Incorrectly Classified Instances        4         2.6667

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

From this table, we see that 97**, 33%** of the sample were classified correctly. The confusion matrix below, indicates that the errors concerned the "iris-versicolor 'class for which **48** examples are correctly classified **50**, and **48** examples for" Iris- vrginica "that are correctly classified **50** examples.

<u>The second value (t: number of attributes + number of classes)</u>

```
Correctly Classified Instances        144        96    %
Incorrectly Classified Instances        6         4    %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

The results are distinguished **96%** were correctly classified, while **4%** were misclassified. The confusion matrix shows that the class **"iris-setosa"** was ranked well in addition to the common errors are the class level **"iris-versicolor"** for which **48** examples **50** are correctly classified, and **48** examples for **"Iris-vrginica"** that are correctly classified **50** examples.

<u>The third value (o: number of classes)</u>

```
Correctly Classified Instances        146        97.3333 %
Incorrectly Classified Instances        4         2.6667 %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

According to this table, it is seen that **97, 33%** of the examples were correctly classified. The matrix of confusion, indicates that the errors related to the class "iris-versicolor" of which 48 examples out of **50** are correctly classified, and **48** examples for "Irises-vrginica" which are correctly classified on **50** examples.

<u>The fourth value (i: number of attributes)</u>

```
Correctly Classified Instances        147        98    %
Incorrectly Classified Instances        3         2    %

=== Confusion Matrix ===
.
  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  1 49 |  c = Iris-virginica
```

The matrix of confusion, indicates that the errors related to the class "iris-versicolor" for which 48 examples out of 50 are correctly classified, and 49 examples for "Irises-vrginica" which are correctly classified on 50 examples. Thus for the database Iris the best result is got if the number of neurons of the hidden layer is equal to the number of attribute with an error rate of 2% and 3 badly classified examples. (The value "I" is the best).

2.1.a Application of Boosting

Who aims at the improvement of the procedures of decision by overweighting the badly classified units, and by reiterating the process.

We applied the BOOSTING in each value (has, **I, O** and **T**)

a: (number of attributes + number of classes)/2

```
Correctly Classified Instances        144        96    %
Incorrectly Classified Instances        6         4    %

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  3 47 |  c = Iris-virginica
```

t: number of attributes + number of classes

```
Correctly Classified Instances        144        96    %
Incorrectly Classified Instances        6         4    %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 47  3 |  b = Iris-versicolor
 0  3 47 |  c = Iris-virginica
```

o: number of classes

```
Correctly Classified Instances        142        94.6667 %
Incorrectly Classified Instances        8         5.3333 %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 46  4 |  b = Iris-versicolor
 0  4 46 |  c = Iris-virginica
```

i: number of attributes

```
Correctly Classified Instances        147        98    %
Incorrectly Classified Instances        3         2    %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 48  2 |  b = Iris-versicolor
 0  1 49 |  c = Iris-virginica
```

Here the table summarizing the results obtained by modifying the Hiddenlayer parameter before and after application of the boosting to the method of neural network:

Database	**a**	**o**	**t**	**i**
% of authority correctly classified	97,33%	97,33%	96%	98%
% of authority correctly classified after	96%	96%	94,66%	98%

After having applied the boosting by using the same number of neurons in the hidden layer and the attributes, one obtains the best error rate by **2%** and **3** badly classified cases.

We notice in this example that the effect of the boosting, on the analysis of the base Iris with the neural networks, **deteriorates** the results for **a, o** and **t** in more, it does not have any effect on i.

2.2 Support Vector Machines

When we carry out the Separating algorithm with Support Vector Machines we got the following results:

```
Correctly Classified Instances          144            96    %
Incorrectly Classified Instances          6             4    %
Kappa statistic                           0.94
Mean absolute error                       0.2311
Root mean squared error                   0.288
Relative absolute error                  52      %
Root relative squared error              61.101  %
Coverage of cases (0.95 level)          100      %
Mean rel. region size (0.95 level)       66.6667 %
Total Number of Instances               150

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

- **96%** of the examples were classified correctly.

The matrix of confusion, indicates that the errors related to the class “iris- versicolor” of which **49** examples out of **50** are correctly classified, and **45** examples for “Irises-vrginica” which are correctly classified on **50** examples.

2.2.a Application of Boosting

```
Correctly Classified Instances        147        98    %
Incorrectly Classified Instances        3         2    %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

It is noticed that the error rate decreased by **4%** to **2%.** In this case the **boosting improved classification.**

2.3 Decision Trees

```
Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

The results show that **96%** of the examples were classified correctly.
The matrix of confusion in the bottom, indicates that the errors related to the class "iris-versicolor" of which 47 examples out of **50** are correctly classified, and **48** examples for "Irises-vrginica" which are correctly classified on **50** examples.

2.3.a Application of boosting

```
Correctly Classified Instances         140               93.3333 %
Incorrectly Classified Instances        10                6.6667 %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

The error rate after the application of boosting increased from 4% to 6.66% after applying boosting. In this case the boosting deteriorates the results.

The following table summarizes the results obtained from the iris base.

Cross-Validation						
	RN	**BO**	**SV**	**BO**	**AD**	**BO**
% of instances Correctly classified	98%	98%	96%	98%	96%	93.33%
% of instances incorrectly	2%	2%	4%	2%	4%	6.667%

From this table we see that the neural network is the most appropriate algorithm for this data since there is an error rate of **2%** and **3** examples badlly classified.

In addition there is another method for this analysis is based on the
"use training set"

The same previous steps are followed by **cross validation** using
"use training set".

We get the following table summarizes the results of the application of learning methods using this option:

Use training set						
	RN	BO	SVM	BO	AD	BO
% of instances Correctly classified	98.666	98.666	96.6667	98%	98%	100 %
% of instances incorrectly classified	1.3333	1.3333	1.3333	3.333	2%	0%

We note in the above table that the results of the use of the full data set are much better than the result of **cross-validation (10 folds).**

The J48 (decision tree) and neural networks are the two most appropriate algorithms for this data set.

EVOLUTION AND CONCLUSION

After performing several experiments with different data and three classifications algorithms (neural networks, SVM, decision trees J48), and evaluation learning through the cross-validation method (10) cutting the training set into ten parts and the method use training set using all of the examples for learning.

By examining the matrix of confusion and error rates, according to the two methods of assessment, we note that the best classifiers for both bases is **decision tree (J48)** and **neural networks.**

The Boosting application on SVM (SMO), decision tree (J48) and neural networks can give different effects:

- Improve the results of the algorithms.
- Worsening the results of the algorithms.
- Do not cause any influence on the outcome of the algorithms.

These effects depend on the size of the database, the Boosting application of the small sized data sets effectively improves the results, if not the Boosting has no influence or little influence on the results.
The experimental results seem to prove the following facts:
Decision trees work well if:

- The number of possible values for each attribute is low.
- Class is qualitative value.

Application of Boosting for decision tree (J48) is more efficient and effective than other algorithms **(SVM and neural networks).**
The calculation time for neural networks is generally higher than the calculation time for systems based on decision trees.

REFERENCES

[1] Probabilistic machine learning and artificial intelligence. Zoubin Ghahramani. Nature 521, 452–459(27 May 2015)

[2] Apprentissage artificiel. Antoine Cornuéjols et Laurent Miclet – Eyrolles Collection : Algorithmes – 2e édition – juillet 2011.

[3] Intelligence artificielle. Stuart Russel, Peter Norvig. Pearson Education. Collection Informatique (December 10, 2010)

[4] Machine Learning. Peter Flach. Cambridge University Press. (August 25, 2012)

[5] Computing machinery and intelligence. Alan M. Turing. Mind, 59, 433-460 (1950).

[6] Murphy, K. P. Machine Learning: A Probabilistic Perspective (MIT Press, 2012).

[7] Weka the University of Waikato. Machine Learning Group at the University of Waikato. http://www.cs.waikato.ac.nz/ml/weka/. 14/06/2017

[8] Antoine Cornuéjols et Claudia Marinica. Cours ISI-3 : Apprentissage Artificiel et fouille de données.https://www.lri.fr/~antoine/Courses/Master-ISI/Cours-ISI-3.html. 10/06/2017

[9] Classification supervisée et non supervisée des données de grande dimension. Charles BOUVEYRON1 & Stéphane GIRARD.Revue MODULAD, 2009

[10] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the Web: Survey and practical guide. Data Mining and Knowledge Discovery, 18:140–181, 2009.

[11] T. M. Mitchell. Machine Learning. McGraw-Hill, New York, NY, 1997.

[12] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning, 36:105–142, 1999

[13] J. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods — Support Vector Learning (Cambridge, MA) (B. Sch¨olkopf, C. J. C. Burges, and A. J. Smola,eds.), MIT Press, 1999, pp. 185–208.

[14] T. M. Mitchell, Machine learning, McGraw-Hill, NewYork, 1997

[15] Barber, D. (2012), Bayesian reasoning and machine learning, Cambridge University Press.

[16] V. Vapnik, “The nature of statistical learning theory, ” Springer-Verlag: New York, 1995.

[17] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273–297.

[18] The connection between regularization operators and support vector kernels," Neural Networks, vol. 11, pp. 637–649, 1998.