Hybrid classification system design using a decision learning approach and three layered structure - A Meta learning paradigm in Data Mining

Lamogha Ighoroje *Lamogha.ighoroje@hud.ac.uk*
University of Huddersfield
Huddersfield, United Kingdom

Joan Lu *j.lu@hud.ac.uk*
University of Huddersfield
Huddersfield, United Kingdom

Qiang Xu *Q.Xu2@*
University of Huddersfield
Huddersfield, United Kingdom

Abstract

A data classification system is designed consisting of three layers. The second layer is the main focus of this research paper. It describes a meta-learning (learning to learn) concept that uses certain characteristics of the dataset as well as some more **general knowledge about supervised and unsupervised machine learning algorithm**s (e.g. supervised learners tend to perform very well in the presence of a large pre-labelled training sets, etc.) to create some *hypothesis*. The main aim of this research is to harness general knowledge about a dataset and different machine learning methods to develop a set of meta-rules that when implemented will help to automate and speed up big data classification processes in data mining. An experiment is conducted to verify the hypotheses made using supervised and unsupervised knowledge flows in weka with some datasets taken from weka and UCI machine learning repositories. The performance result of the experiments is used to design a meta-learning algorithm in form of rules. The results from the experiments confirmed that general knowledge known about supervised and unsupervised learning is then harnessed successfully for making learning decisions.

Keywords: Meta Learning, Learning To Learn, Classification, Clustering, Supervised Learning, Unsupervised Learning, Machine Learning, Meta –Rules, data mining, Security Classification, big data classification.

1. Introduction

In designing classification systems, there is often a challenge with the search and selection of best performing machine learning algorithm(s) to use for a dataset in a short period of time. Often, one has to learn thoroughly about the data set structure & content, decide whether to use a supervised learning strategy or an unsupervised learning strategy and then investigate, select or design a classification or clustering algorithm that would work most accurately for that specific dataset. This can be quite a time consuming and tedious process. Additionally, a classification algorithm may not perform very well with a particular dataset as compared to if a clustering algorithm is used. Hence, the problem becomes "How can we automatically determine the best Machine learning (ML) method(s) to implement that can yield the best accuracy, given a heterogeneously large dataset and knowledge about *supervised* and *unsupervised* machine learning?"

As:

- A classifier trained using a particular labelled dataset may not be suitable for another dataset.
- Traditional methods cannot efficiently accommodate the large varieties of class types found in a dynamically growing dataset. This often leads to inaccurate classification results.
- Traditional methods are not suitable for present day multiple learning tasks [16].

One of the most basic ways for organizations to determine the relative importance of the data they possess is through data classification. An interview of three chief information security officers (CISOs), Microsoft, royal bank of Scotland and dell incorporations, by Microsoft trustworthy computing in [4] confirms the importance of data classification in today's information security scenery. Data classification allows organizations to determine associated data risks through the categorization of their stored data by sensitivity and business impact [14]. When data classification is properly implemented, it helps to ensure that confidential or sensitive data assets are managed and controlled better than less sensitive data assets, which leads to better resource prioritization. More on data classification can be found in [1, 2], [5], [9], [11], [12], [14] and in many other literatures. Ultimately, the goal of data classification is to control all enterprise data by discovering, protecting or destroying it based on its level of importance and potential impact [19]. The data many organizations have to deal with in recent years is referred to as big data; hence it is important to reason data classification in terms of big data. The characteristic of big data (*volume*, *variety*, & *velocity*) brings about new challenges and opportunities for classification algorithms [16]. The '*sensitivity*' of big data has been identified as a very vital security issue as regard to its use [17]. Hence, it will be beneficial if one can consider classifying big data based on sensitivity levels. Although there are a lot of data classification algorithms and tools that can be used to achieve big data classification, a major challenge still exists in deciding the best learning algorithm(s) to use for a given task, that can outperform others and at the same time has the ability to effectively address the major challenges of big data. These challenges are as a result of the characteristics of data being too large, having varieties of sources and at very high speeds. For example, some learning algorithms may not be very effective for handling heterogeneous datasets, large data streams, large datasets with changes for which they were not previously trained to handle in an automatic and effective manner.

Making the right decision about the best learning algorithm(s) to use in designing a classification system has over time become a time consuming, tedious and costly process. In machine learning, the decision about what learning method (*supervised learning/classifier* OR *unsupervised learning/clusterer*) has been incorporated into the meta-learning (*Learning to learn*) research. Meta-learning has proven to have a major correlation with classification tasks.

An interesting fact observed in the design of an effective classification system is that, there is a major distinct connection between the meta-learning paradigm and data mining classification. This connection is due to the fact that while designing a classification system, one must empirically & analytically study existing algorithms (tons of algorithms exists) and in some cases even make use of some base concepts or hypothesis. When designing the classification system, the process of deciding what machine learning approach (supervised and unsupervised) to be used in next after defining the goal. There are many trends and knowledge shown over the years about supervised and unsupervised machine learning, which can be formally harnessed in reducing the time spent in taking such decisions.

This research proposes a hybrid classification system architecture that comprises of three different layers. The second layer which is a decision learning level, automates the decision making process on what learning method to adopt at any point in time, given a heterogeneously large stream of data sets. This decision making process is in itself a Meta-learning (learning to learn) process. The first and third layers of the classification system are not the main focus of this research paper. Hence, this paper majorly contains discussions, experimental analyses and hypothesis about the first meta-learning level. The Weka (Waikato Environment for Knowledge Analysis) [10] tool is used in this research for the experimental study. It is a data-mining tool designed mainly for research purposes and widely accepted in the data mining community. It contains a lot of tools that allows for performing data mining tasks easily and can help assist in the development of new machine learning schemes.

An earlier formal abstraction of the algorithm selection problem is discussed in [13]. The author aims to answer the question: "what algorithm is best to use in a particular scenario?" by formalizing four criteria (the problem space **P**, the feature space **F**, the algorithm space **A** & the performance space **Y**) and five main steps as an analysis and possible solution for the algorithm selection problem. *Selection mapping* was also echoed from observations by the author as a single most important part of the algorithm selection problem solution.

Later on in [3], the term 'meta-learning' is coined. In the paper, the author discusses ways in which we can draw more general conclusions from the results of machine learning experiments, to give us a set of rules that unfolds situations in which certain algorithms significantly outdo others based on some needful measures. However similar some concepts are, the meta-learning hypothesis discussed in this research paper distinguishes from the study in [3] as it considers case studies involving both supervised and unsupervised learning and not only supervised learners. The set of Meta rules derived in this research paper is a result of empirical studies conducted to determine situations in which using a supervised learning algorithm might be more effective than using an unsupervised algorithm.

There are varying views of meta-learning in literatures, for example, in [18] the authors provide a survey of different meta-learning views with regards to machine learning. The authors also discuss their own viewpoint of meta-learning from the point of constructing self-adaptive leaners, which gathers its Meta knowledge by analyzing the whole instance and updates the knowledge base according to the characteristics of individual instances. They also point out an important fact, which states that despite the varying views on meta-learning, this constant question (how can knowledge about learning be exploited to improve the performance of learning algorithms?) remains unchanged. The process of learning to learn involves studying ways to improve learning by discovering, mining, and taking advantage of the *invariant transformations* across multiple domains. *Invariant transformations* gives a more general understanding of the nature of patterns across domains [18].

A unified framework used for analyzing various research developments that aims to tackle the algorithm selection problem as a general learning problem across different domains was shown in [15]. Some literatures refer to meta-learning algorithms as one in which learning improves in each iterative run of a base classifier. In some, it is referred to as the process of putting together a set of characteristics or meta-features specific to a particular domain with respect to the classifier's performance. For example, in [6], the authors use meta-learning to propose a novel dynamic ensemble selection framework, where five sets of meta-features capturing different properties of the base learner is proposed for classifier selection. Their classification selection rule is learned by a meta-classifier making use of the training data, which enables a set of rules to be induced by using a meta-learner to observe what conditions makes a learning algorithm perform better than others. This is limited as the meta-learner used for this analysis is related to only specific domain characteristics and not characteristics that can cut across domains.

However, this research paper presents a meta-learning concept that uses more general knowledge about supervised and unsupervised machine learning algorithms to create some hypothesis that is applied in an experiment. Based on the performance results of the experiments a set of decision rules are drawn to enable the decision learning process, which will further aid in achieving a high performance and automatic classification of big data.

2. Proposed System Design

This section describes briefly the design goals of the overall classification system proposed and depicts the importance of the meta-learning (learning-to-learn) decision layer of the system.

Design Goals

1. A meta-learning rule-based design that defines a structure for automatically determining whether to invoke a supervised learning algorithm or an unsupervised learning algorithm.
2. An unsupervised algorithm that is not only self-evolving (determining the classes from scratch without any labeled instances), but one that allows for a re-grouping of the classes to avoid having a large dataset of classes.
3. Scalability in terms of the system handling an increasing amount of heterogeneous data and data categories.
4. Achieves classification at a desired speed.
5. Can handle the challenges of big data effectively through parallel processing optimization.
6. Flexibility and adaptability.

Proposed Model Architecture.

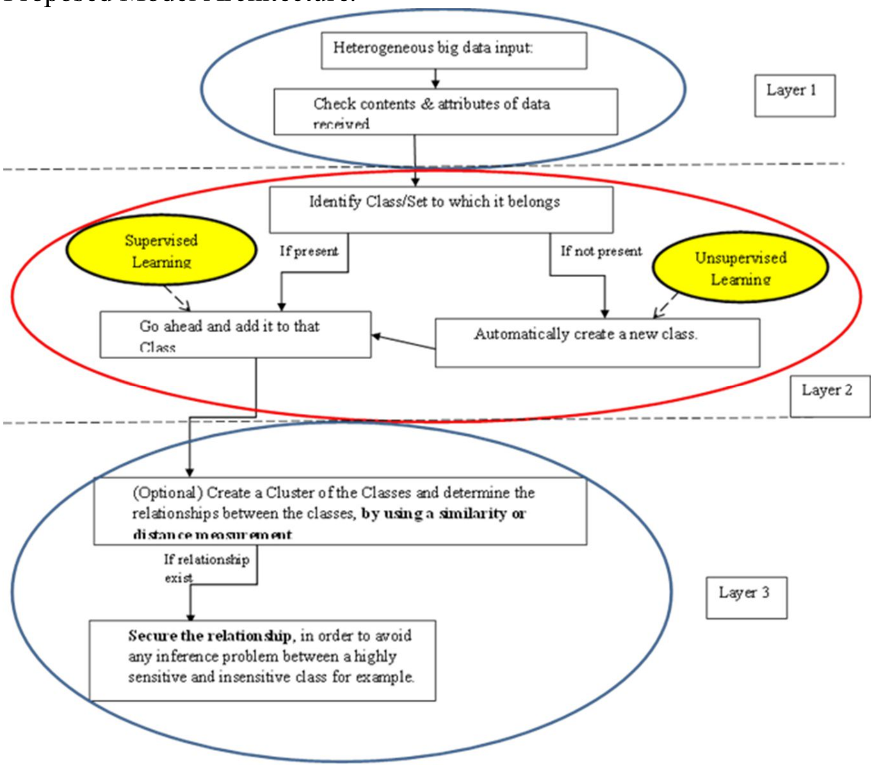

Fig. 1. A three tiered-layer architecture for the hybrid classification system.

Model Components

A. Layer 1 (Input / Pre-processing Layer): Since big data is a collection of heterogeneous data which makes it difficult to analyze [8], this layer ensures that an inflow of such a data set is pre-processed appropriately. The pre-processing phase will involve dividing the vast source of data into domain specific sources of knowledge, next a check through the contents and attributes of the data is done to determine if any knowledge or information about its content is present. Having this layer will assists in the process of preventing vagueness in the heterogeneous data. This layer provides layer 2 the reasoning about classifying data using either a supervised classifier or and unsupervised classifier.

B. Layer 2 (Strategic Learning Decision Layer): At this layer, the decision on which learning method to invoke is made. The main aim of this layer is automatic classification using the most effective learning method to achieve a high level of accuracy at a fast speed. The *hypothesis* used in this layer for making a decision is based on some general characteristics and knowledge about *supervised* and *unsupervised* machine learning. For example, characteristics such as the **existence of pre-existing labelled set for training** or not, the **size of the pre-labelled training set** (under the assumption that the size is relative to the number of instances in a particular dataset), **existence of a test set** which is a subset of the training set, etc.

Hypothesis 1: A supervised learner will be more appropriate than an unsupervised learner. Given a data set **D**, with an already existing large set of pre-labelled training data $\mathbf{D_{train}}$ and a test set $\mathbf{D_{test}}$ which is relatively smaller in size than $\mathbf{D_{train}}$, and based on general knowledge gained about supervised learners performing well in the presence of a larger pre-labelled $\mathbf{D_{train}}$,

Hypothesis 2: An unsupervised learner will more appropriate than a supervised learner. Given a data set **D**, without pre-labelled training data instances and the knowledge that unsupervised learners are best used when no pre-labelled training dataset exists, then it can be said that

C. **Layer 3 (Output / Optional Cluster formation Layer):** This layer is an optional layer whose aim is for scalability and to properly secure the relationships between class labels. For example, imagine a scenario in which the amount of resulting class labels becomes very large. The question now becomes: 'how can we effectively manage a large and increasing set of class labels?' At this layer, a good technique to effectively manage a large and self-evolving set of class labels is considered. This technique considers the formation of clusters/groups for the class labels by making use of a similarity or distance measurement. The resulting output from this layer will be a set of cluster labels (similar to the class labels, but for representing some knowledge about the clusters).

Model Characteristics

1. **Meta-learning / automatic learning architecture**: where supervised and unsupervised classification algorithms will be combined together and depending on certain characteristics knowledge of the data set under consideration, one of the algorithm is invoked automatically to give more accurate classes. This reduces significantly the time spent in deciding the best classification algorithm to use for a particular data set and the high cost of learning realistically accurate classifiers is overcome.
2. **Multi Class-label type classification**: a new unsupervised algorithm is also intended to be developed in this research, which can be used successfully in second layer of the classification system. The algorithm allows an instance of a data set to have multiple class labels based on sensitivity levels (e.g. *sensitive level* $\boldsymbol{l_1}$, $\boldsymbol{l_2}$....$\boldsymbol{l_n}$) assigned to each attribute per instance, rather than assigning one class label to the data instance as a whole (see illustration of this in Table 1 below).

Table 35. Hypothetical example case study of a multi-class label unsupervised algorithm.

#	**Bank ID**	**LName**	**FName**	**D.o.B**
1	10a	Flora	Catch	29.09.83
2	20s	Robin	Thomson	05.10.75
3	3b	Martha	Woods	04.7.60
Class	L1	L3	L2	L1

From Table 1, there are 4 attribute features and 3 instances of the dataset. Every bank ID and D.o.B. is given a sensitivity class label l_1, (where l_1 is assumed to be the most sensitive class), every instance of the Fname is given the label l_2 and the Lname is given a label l_3. From this, it is observed that each instance in the data set may have one or more class labels.

3. **Meta-Classification**: this simply means a process of classifying the classes.
4. **Multilevel type structure classification.**
5. **Auto-Class functionality**: the beneficial features of Auto-Class includes: 1) its ability to determine the number of classes automatically, 2) it permits the blend of discrete and real valued data, 3) it can handle missing values effectively.
6. **Classification Methods to be used:** *Probabilistic* and *Rule based* methods will be employed.
7. **Output**: the intended output per instance will be a numerical score that can be converted to a discrete label.

3. Materials

The hypothesis as stated in Section 2.2 which aims to allow an automation of the decision to either use a supervised learner or an unsupervised learner in the classification system proposed is experimented in Weka [10]. Using example datasets taken from Weka and also a data set [7] from the UCI machine-learning repository. Weka was the tool of choice to use in this experiment, because of its wide acceptance in the data mining community and its easy to use GUI interface.

The experiments were setup using the knowledge flow tool in Weka and was designed as shown in figure 2 and 3 below to determine the resulting performances of supervised and unsupervised algorithms present in weka and what factors or characteristics influenced their performance. A total of five supervised algorithms and two unsupervised algorithms were tested multiple times on the different datasets.

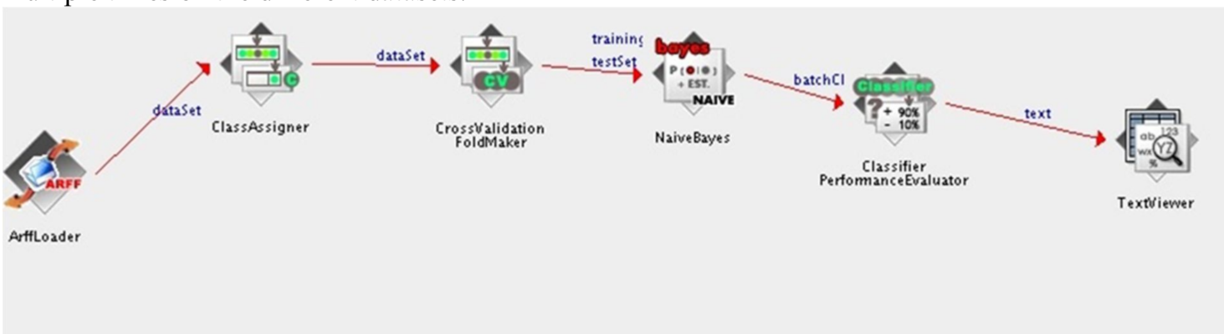

Fig. 2. Supervised Learning knowledge flow setup in weka.

In the experiment seen in Figure 3 above, the different datasets were loaded in by configuring the ‘ArffLoader’; the ‘ClassAssigner’ determines what the class label in the dataset is; a ‘Cross Validation FoldMaker’ and a ‘Train Test SplitMaker’ where used interchangeably to split the dataset into training and test sets; several supervised algorithms were used during different runs of the experiment instead of just a ‘NaiveBayes’ classifier alone.

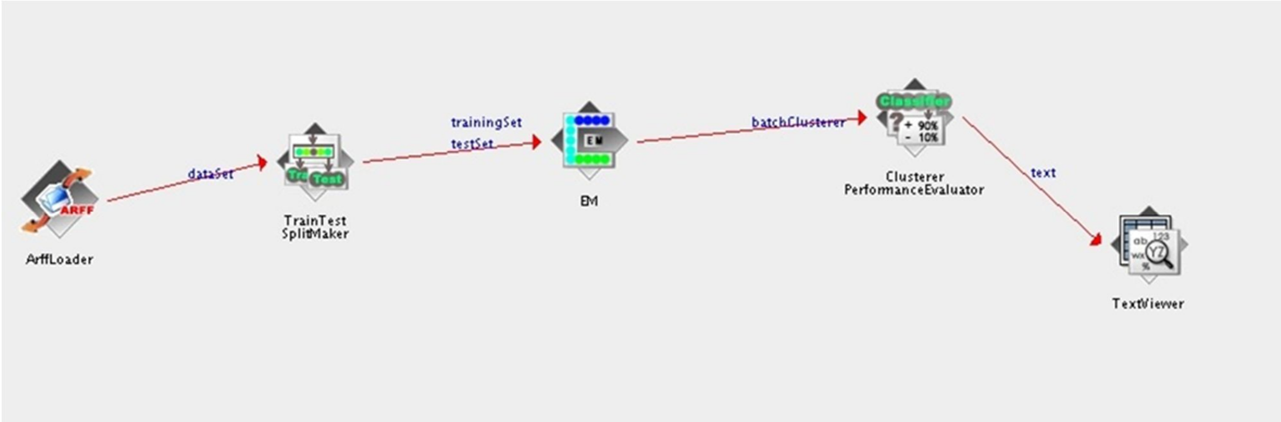

Fig. 3. Unsupervised Learning knowledge flow setup in weka.

For the unsupervised experiments, the 'Train Test SplitMaker' was varied with the 'Cross Validation FoldMaker' to split the dataset into both training and test sets.

4. Results and Discussions

The results obtained after each run of the supervised algorithms looks similar to what is observed in figure 4 while the results obtained from the different experimental runs of the unsupervised learner looks similar to what is seen in figure 6. It is important to note that during the experimental runs, for us to obtain varying sizes of training and test sets, the % parameter of the 'Train Test SplitMaker' and the number of folds in the 'Cross Validation FoldMaker' were adjusted severally.

```
=== Evaluation result ===

Scheme: NaiveBayes
Relation: contact-lenses


Correctly Classified Instances          20               83.3333 %
Incorrectly Classified Instances         4               16.6667 %
Kappa statistic                          0.6991
Mean absolute error                      0.2447
Root mean squared error                  0.3104
Relative absolute error                 65.3746 %
Root relative squared error             72.6025 %
Total Number of Instances               24

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.105      0.714      1         0.833      0.947    soft
                 0.5       0.05       0.667      0.5       0.571      0.913    hard
                 0.867     0.111      0.929      0.867     0.897      0.926    none
Weighted Avg.    0.833     0.1        0.84       0.833     0.829      0.928

=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  1  2  1 |  b = hard
  1  1 13 |  c = none
```

Fig. 4. Example of how the Supervised Learner result which displays several information about the performance of the classifier.

Using as an example illustration to explain the evaluation results better, is the result (Table 2 and Figure 5) obtained from a particular dataset called contact lenses, which contained 24 instances and 5 attributes. The influence of the training data size and test data size on the accuracy of a supervised algorithm (simple Naïve Bayes) is observed.

Table 2. Result from running a simple Naïve Bayes classifier on the contact-lenses data set.

Train instances	**Test instances**	**Correctness (%)**	**Kappa statistics**	**Mean Absolute err.**
24	24	95.8333	0.925	0.1783
24	10	100	1	0.1732
10	24	87.5	0.7895	0.245

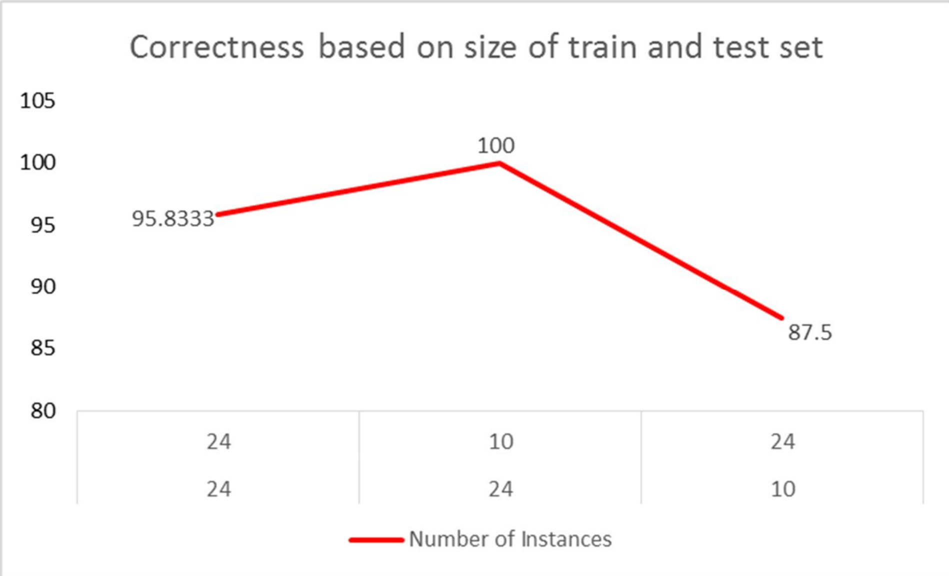

Fig. 5. Display of the influence the train and test datasets have on the classifier's accuracy (the test data size is represented on the horizontal top row, while the train data size is represented on the horizontal bottom row).

From Table 2 and Fig 5, it is observed that when the train dataset size is the same as the test dataset size the accuracy of the naïve Bayes classifier is better than if it was smaller than the test dataset size. Likewise, when the train dataset size was relatively larger than the test dataset size, the accuracy of the classifier improved to a 100% (in terms of the number of correctly classified instances).

```
=== Evaluation result for testinstances ===

Scheme: EMRelation: contact-lenses


EM
==

Number of clusters selected by cross validation: 3


                 Cluster
Attribute             0 1 2
                      (0.38)(0.38)(0.25)
==========================
age
  young               4 4 1
  pre-presbyopic      2 3 4
  presbyopic          3 2 2
  [total]             9 9 7
spectacle-prescrip
  myope               6 3 1
  hypermetrope        2 5 5
  [total]             8 8 6
astigmatism
  no                  2 6 2
  yes                 6 2 4
  [total]             8 8 6
tear-prod-rate
  reduced             6 1 4
  normal              2 7 2
  [total]             8 8 6
contact-lenses
  soft                1 6 1
  hard                2 2 1
  none                6 1 5
  [total]             9 9 7
Clustered Instances

0      6 ( 75%)
2      2 ( 25%)


Log likelihood: -4.57453
```

Fig. 6. Example of the results obtained after a run of the unsupervised learner flow.

From the various results obtained by carrying out the experiment of the knowledge described in supervised and unsupervised learning (i.e. in hypothesis 1 & 2 of section 3.1 above), it is observed that certain characteristics of the data set and knowledge about the machine learning methods definitely influence performance generally. Some conclusions derived from the experiments include:

- If a set of class labels exists already and can be specified for all training instances, then supervised learning is preferred.
- Supervised learning is preferable if a large amount of pre-labelled training set already exists in the presence of a small test sample under consideration,
- When the number of instances to be classified is small,
- Unsupervised learning is preferable if no pre-existing class label exists,
- Unsupervised learning is preferable if the training set is way smaller than the sample set to be tested, etc.

The conclusions derived from these experiments allows us to easily describe the decision learning (learning to learn) process of the classification system proposed as a set of Rules. Below is the Meta learning algorithm designed to this effect.

Meta-learning (Learning to learn) Algorithm:

Input: An inflow collection of both labeled (D_l) datasets and unlabeled (D_u) datasets from heterogeneous data sources OR a collection of fully unlabeled heterogeneous dataset (D). Also a set of IF → THEN rules defined from experimental knowledge obtained about supervised and unsupervised learning, that helps in the decision making process.

Output: A decision that invokes either a supervised classification algorithm or an unsupervised classification algorithm.

a. IF training labeled set exists then check the size of the labeled set.
b. IF size of the training set > than the test set, THEN invoke a supervised learning method.
c. IF no training set exists, THEN use an unsupervised algorithm.
d. IF the size of the training set < or = test set, use an unsupervised algorithm.

e. IF no labeled instances exist, use an unsupervised algorithm.
f. Output new decision by automatically invoking a learning algorithm that is the best fit for that dataset.

5. Conclusions and Future Work

This research paper shows the link between big data classification and the Meta learning paradigm, how knowledge obtained about a dataset and general knowledge about supervised and unsupervised learning can be used to design a set of meta-rules (decision rules) for the automatic selection of appropriate machine learning algorithms in a big data classification system. A hybrid data classification system is designed consisting of three layers. The second layer is the main focus of this research paper. It describes a meta-learning concept that uses certain characteristics of the dataset (such as the existence of pre-existing labelled set for training or not, the size of the training set, existence and size of a test set). Also two hypotheses were created based on general knowledge about supervised and unsupervised machine learning algorithms (e.g. supervised learners tend to perform very well in the presence of a large pre-labelled training set, etc.). An experiment is then conducted to verify these hypotheses, using supervised and unsupervised knowledge flows setup in weka, with some datasets taken from weka and UCI machine learning repositories. However, based on the performance results of the experiments, it can be said that a supervised algorithm is more appropriate to use than an unsupervised algorithm in the presence of large pre-labelled training set while an unsupervised algorithm is more appropriate to use in the absence of a large pre-labelled training set. A decision-learning algorithm in form of rules is also obtained from the result. The implementation of the meta-learning algorithm will help to automate and speed up the ML algorithm selection problem.

Achievements

- Design of a hybrid classification system, in which at any given point in time (depending on the dataset received) either a supervised machine learning algorithm or an unsupervised learning algorithm is invoked automatically.
- By using some general knowledge about Supervised and Unsupervised machine learning methods, some hypothesis were made.
- Experiments using Weka data mining tool and some datasets from weka and the UCI machine learning repository was performed to confirm the hypothesis and also help us in drawing some more conclusive rules that can assist in the automatic decision learning process of the classification system. The hypothesis were confirmed true.
- A meta-learning algorithm comprising of a set of rules was then formulated for the decision-learning layer of the classification system proposed.

Limitations: a limitation of this study is that the number of unsupervised algorithms available in weka (the tool used for experimental analysis) was not sufficient enough to fully uncover a wider variety of general knowledge about unsupervised learning, which can be used in decision learning process.

Future work

- A fully completed implementation of the Meta learning algorithm in the decision layer of the big data classification system.
- Design and testing of a fully self-evolving unsupervised classification algorithm that supports a re-classification of classes (if the number of classes become too large).
- Considering the challenges of big data, incorporate some big data processing methods to ensure the system can be used to classify big data efficiently.

Acknowledgements

Special thanks to Mr. and Mrs. Ighoroje, the supervisory team, administrative and financial staffs of the School of Computing and Engineering at University of Huddersfield for making it possible to undergo this research and present this research paper at the ISD 2016 conference.

References

1. Aggarwal, C. C. (2014). Data Classification: Algorithms and Applications. Hoboken, Chapman and Hall/CRC.
2. Aggarwal, C. C. (2014). "An Introduction to Data Classification." Data Classification: Algorithms and Applications: 1.
3. Aha, D. W. (1992). Generalizing from case studies: A case study. Proc. of the 9th International Conference on Machine Learning.
4. Computing, M. T. (2014). "CISO Perspectives on Data Classification." Trustworthy Computing. 2015, accessed 6 May 2016.
5. Concepts, O. D. M. (2008) 11g Release 1 (11.1). Oracle Corp 2007.
6. Cruz, R. M., et al. (2015). "META-DES: A dynamic ensemble selection framework using meta-learning." Pattern recognition 48(5): 1925-1935.
7. Davide, A. A., Ghio Luca, Oneto Xavier, Parra Jorge, L.Reyes-Ortiz (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Belgium, ESANN 2013: 24-26.
8. Doug, L. (2001). Data Management: Controlling Data Volume, Velocity, and Variety, Application Delivery Strategies", META Group (currently with Gartner).
9. Fabrico, L. Data Mining Classification (2014), https://courses.cs.washington.edu/courses/csep521/07wi/prj/leonardo_fabricio.pdf, Accessed May 15, 2015.
10. Hall, M., et al. (2009). "The WEKA data mining software: an update." ACM SIGKDD Explorations Newsletter 11(1): 10-18.
11. Japkowicz, N. and M. Shah (2011). Evaluating learning algorithms: a classification perspective, Cambridge University Press.
12. Reed, B. (2007). Data classification best practices. Network World. Southborough, Network World Inc. 24: 18.
13. Rice, J. R. (1975). "The algorithm selection problem." Computing Science technical reports, Department of Computer Science, Purdue University, and Report Number: 75-152.
14. Simorjay, F. (2014). "Data Classification for Cloud Readiness." Microsoft Trustworthy Computing.
15. Smith-Miles, K. A. (2009). "Cross-disciplinary perspectives on meta-learning for algorithm selection." ACM Computing Surveys (CSUR) 41(1): 6.
16. Suthaharan, S. (2014). "Big data classification: problems and challenges in network intrusion prediction with machine learning." ACM SIGMETRICS Performance Evaluation Review 41(4): 70-73.
17. Tankard, C. (2012). "Big data security." Network security 2012(7): 5-8.
18. Vilalta, R. and Y. Drissi (2002). "A perspective view and survey of meta-learning." Artificial Intelligence Review 18(2): 77-95.
19. Woody, A. (2013). Enterprise Security: A Data-Centric Approach to Securing the Enterprise, Packt Publishing Ltd.