Human Friendliness of Classifiers: A Review

HADDELA, Prasanna, HIRSCH, Laurence <http://orcid.org/0000-0002-3589-9816>, GAUDOIN, Jotham and BRUNSDON, Teresa

Human Friendliness of Classifiers: A Review

Prasanna Haddela[1,3*], Laurence Hirsch[1], Teresa Brunsdon[2] and Jotham Gaudoin[1]

Abstract During the past few decades Classifiers have become a heavily studied and well-researched area. Classifiers are often used in many modern applications as a core computing technique. However, it has been observed that many popular and highly accurate classifiers are lacking an important characteristic; that of human friendliness. This hinders the ability of end users to interpret and fine-tune the method of decision-making process as human friendliness allows for crucial decision making towards Applications. This paper presents, in term of classification (i) a taxonomy for human-friendliness (ii) comparisons with well-known classifiers as related to human friendliness, and (iii) discussion regarding recent developments and challenges in the field.

1 Introduction

Classification is an important machine learning technique that involves categorizing unseen instances into pre-labeled groups. This is also known as supervised learning. In the process of building classifiers, a "Training set" or historical data of a similar scenario is used. Apparently, much dedicated research efforts, have resulted in a few widely accepted classifiers and they are used in many different application domains [6]. But none of the classifiers can be termed perfect solutions due to the natural complexity of the problem.

Among the various types of classification applications, it is noted that a subset of classification applications essentially requires human friendliness. With this research, human friendliness is defined as the end user's ability to interpret and modify (fine-tune) the decision-making criteria. For example, consider a medical diagnostic system and a rule received for stroke risk prediction. Fig. 1, illustrates a rule received from a stroke prediction model [20]. As domain experts, doctors

if *hemiplegia* **then** *stroke risk* 59.0%
else if *cerebrovascular disorder* **then** *stroke risk* 44.7%
else if *hypovolaemia* and *chest pain* **then** *stroke risk* 14.6%
else if *transient ischaemic attack* **then** *stroke risk* 29.9%
else if *age_70* **then** *stroke risk* 4.5%
else *stroke risk* 9.0%

Fig. 1. Rule from stroke prediction system

may be highly motivated to use these types of systems due to the characteristic of human interpretability of the decision-making process. To the domain experts, how the system functions

Prasanna S. Haddela
e-mail: prasanna.s@sliit.lk

Laurence Hirsch
e-mail: l.hirsch@shu.ac.uk

Teresa Brunsdon
e-mail: teresa.brunsdon@warwick.ac.uk

Jotham Gaudoin
e-mail: j.gaudoin@shu.ac.uk

[1]Sheffield Hallam University, Sheffield S1 1WB, United Kingdom

[2]University of Warwick, Coventry, CV4 7AL, United Kingdom

[3]Sri Lanka Institute of Information Technology, Colombo, Sri Lanka

is highly transparent. This is vital before trusting it as a decision support tool in certain industries. Since there is no perfect classification method developed, the ability to customize the classifier is always of merit because it allows knowledgeable experts in the field to fine-tune the system.

Unfortunately, commonly used and widely accepted as highly accurate classifiers appear to lack human interpretability and the ability for experts to customize to suit requirements. For example, Support Vector Machine (SVM) and Artificial Neural Networks (ANN) are not human friendly. They obstruct monitoring and fine-tuning.

This paper reviews commonly used existing methods and techniques for classification, categorizes them and presents research challenges and ends with possible research directions.

The rest of the paper has been organized as follows; section 2 details the taxonomy for text classification. Section 3 shows some outlined applications indicating where they need human-friendly classifiers. Sections 4 and 5 illustrate the type of classifiers and variety of rule-based classifiers respectively. Finally, section 6 compares the classifiers in term of human friendliness.

2 Taxonomy for Classification

In the past, accuracy and efficiency dominated the development of classifiers. However, it is observed that some of the applications have an equally important characteristic for domain experts; which is human friendliness of the classifier. Human friendliness is defined as the ability of end users to interpret and modify the decision-making process of classifiers. The taxonomy shown in Fig 2 was developed based on the level of human friendliness of classifiers.

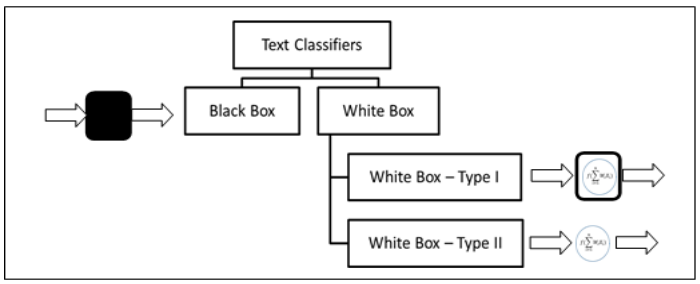

Fig. 2. Taxonomy for Classification

This taxonomy branches classifiers as black box and white box. Further, the white box classifiers have Type I and Type II. This taxonomy is used to organize the classifiers and to organize paper content.

2.1 Black-box Type Classifiers

Black box type classification models are capable of classifying text, but human interpretability and modifiability of the model are absent. Accordingly, end users of the classifier are blind about the way the classification has been carried out and also find it difficult to fine tune the model. Support Vector Machine (SVM) and Artificial Neural Network (ANN) based models are good examples for black box type classifiers.

2.2 White-box Type Classifiers

The main feature of White box type classification models is interpretability. Also, it has a higher level of human friendliness due to: transparency, explainability and sometimes modifiability.

Classifiers in this category can further divide into two groups: Type I, Classifiers are human interpretable but not easy to fine-tune or modify and Type II, Classifiers are human interpretable and also free to modify as needed by domain experts.

For example, rules derived from a decision tree are human interpretable to end users, but it has a limitation of how to incorporate end user's knowledge or feedback and reconstruct tree to produce optimal decision tree. Therefore, this decision tree is categorized as White-box – Type I classifiers. Classifiers belonging to Type II are search query-based classifiers or simple decision rules which represent a particular category and have the power of categorizing text into free labeled groups. These are capable of human interpretability as well as modifiability. Therefore, White box - Type II classifiers have achieved the highest level of human friendliness.

3 Human Friendliness of Medical Applications

White Box type classifiers are heavily used in applications requiring a high level of human friendliness as well as accuracy. In many cases, end users are curious about how it works and what if they do some changes before taking an action. This happens specially when users have sound knowledge about the specific classification problem.

There are many medical applications where the White box classifiers are heavily utilized. Most of the experts in the medical domain would like to know what the prediction model of the systems is and also to use their knowledge for fine tuning the system. Medical Scoring Systems are one of them and are used to recommend personalized medicine. They are designed to be human interpretable and also aim to fine-tune for maximum accuracy. Therefore, White box - Type II classifiers are more suitable for these applications. CHADS2 score system for stroke risk [9], Thrombolysis in Myocardial Infarction (TIMI) [1], Apache II score for infant mortality in the ICU [17], the CURB-65 score for predicting mortality in community-acquired pneumonia [21], system introduced in [20] for stroke prediction, in [18] presented an application where the thoracic surgery rules have been induced by classifying thoracic surgery into different classes are some of the applications for this category used in the health sector.

4 Type of Classifiers Vs Human Friendliness

Researchers have invented many classifiers employing different concepts and techniques for various types of datasets. The following classifiers from among them are popularly used.

SVM Classifiers: SVM classifiers are using linear or non-linear functions to partition the high dimensional data space for different classes or categories. In this case, the main challenge is to identify optimal boundaries between categories. The founder, Joachim states that [15], it is not necessary to human involved in parameter tuning as there is a theoretically motivated parameter tuning set up in place. This automatic parameter tuning hides internal behavior and eventually this means that SVM gives us a minimal level of human friendliness.

Neural Network Classifiers: Artificial Neural Networks (ANN) are popular classification techniques in many software solutions. These models are inspired by biological neural networks of animal brains. In ANNs, connectivity between input layer and output layer creates through the hidden layers. Due to the complex nature of connectivity, it makes ANN more complicated to understand. The activation function decides the output of each node for the set of inputs and which makes ANN a non-linear classifier. Both ANN and Deep NN methods are opaque or black box type due to the complexity and there have been many research attempts to make them are more human-friendly [5][2][27].

Naïve Bayes (NB) is a well-studied classifier in machine leaning research. The method used in NB classifier is that the joint probabilities of features and categories are used to compute the probabilities of categories of a given instance. For this, it makes the assumption of feature independence. That is the conditional probability of a feature given category is assumed to be independent from the conditional probabilities of other features given in that category. Due to the low human friendliness, some research attempts have tried to make it more interpretable [23] [28].

Nearest Neighbor Classifiers: The k-Nearest Neighbor (kNN) classification method finds closes (neighboring) objects within the training set and assign class labels. For a new object, the distance of k neighboring objects is measured and the label of majority class assigned. Euclidean distance or the Cosine value are commonly used distance or similarity measures [16]. In [29] the authors develop a system to convert kNN decisions in to set of rules in that way improving human friendliness which is lacking in such classifiers.

Rocchio method: The Rocchio method is used for inducing linear, profile style classifiers. It relies on an adaptation to classification of the well-known Rocchio's formula for relevance feedback in the vector space model, and it is perhaps the only classification method rooted in the information retrieval tradition [8] rather than in the machine learning. This adaptation was first proposed by Hull [14]. Some linear classifiers consist of an explicit profile (or prototypical document) of the category. This has obvious advantages in terms of interpretability as such a profile is more readily understandable by a human.

Decision Tree-based Classifiers: Decision Tree (DT) is a hierarchical view of the training set. Class labels are mapped to leaf node in the tree. Parent nodes split instances for child nodes based on impurity measures. Information gain and Gini index are commonly used in decision tree induction algorithms. When all instances belong to a child node, it stops splitting and makes it a leaf node otherwise nodes are split recursively. Decision trees are more transparent and make it easier to build rules for classification. Therefore, DT has a higher level of human friendliness.

Rule-based Classifiers: Rule-based classifiers are determined the features or term patterns which are most likely to be related to the different classes. Decision criteria consist of DNF (Disjunctive Normal Form) rules which denote the presence or absence of terms patterns in the testing object while the clause head denotes the category. These rules are used for the purposes of classification. Rule-based classifiers have the highest level of human friendliness.

Genetic Algorithm-based classifiers: Genetic Algorithm-based (GA) methods follow iteratively progressing approach to develop a population towards achieving the desired end. Such developments are often inspired by biological mechanisms of evolution. The genetic operators; selection, crossover (recombination) and mutation are applied to the individuals to breed the next generations. Fitness functions are used in order to measure the strength of an individual [22], [19]. When the fitness is higher, there is a high probability for the next generation of individuals to be selected, to take part in creating the next generation. Thus, the genetic material of strong individuals will survive throughout the evolutionary process until an optimal or near optimal solution is found. GAs are often used to support other algorithms, for example by parameter tuning. GAs have also been used to generate search query based text classifiers with very high human friendliness[11].

5 Human Friendliness: Approaches and Challenges

In the past, there have been many research attempts to improve the accuracy of classifiers, but in most cases, human friendliness of classifiers was neglected. However, in the recent past, a new trend of building methods to enable human friendliness for Black box type classifiers, is evident [20], [5],[23]. Yet, White box type classifiers which are human friendly by nature appear not to

have drawn much attention. This section reviews White box type classifiers and other related approaches.

5.1 White Box Classifiers: Type I versus Type II

Decision tree-based classifiers are very frequently used with projects. These classifiers often use decision rules based on information theory to branch the parent node and link with the child nodes. This method is transparent and interpretable. But decision trees are not flexible for end users to modify. Therefore, such classifiers are categorized under Type 1.

The following two examples illustrate a popular rule-based system and highly compact search query-based text classifiers. In CONSTRUE [10] to classify documents in the 'wheat' category of the Reuters dataset an example rule of the type used is illustrated below.

if ((wheat & farm) **or**
(wheat & commodity) **or**
(bushels & export) **or**
(wheat & tonnes) **or**
(wheat & winter & ¬ soft))
then
WHEAT **else** ¬ WHEAT

The search queries evolved from GA-SFQ [11] system for popular Reuters dataset acquisitions category;

(buy 10) (company 11) (bid 13) (offer 15)

In GA-SFQ, terms and their proximity to the start of a document are taken into consideration when constructing these type of search query-based classifiers. In [35] various search query types were evaluated for classifier effectiveness but with a clear objective of interpretability.

These search queries and rules are highly interpretable and also mean that terms of the classifiers can be amended by end users easily and applied again. These classifiers belong to White box- Type II and has highest level of human friendliness compared to Black box type classifiers and White box - Type I classifiers.

5.2 Rule-based Classifiers: Direct versus Indirect

Rule-based classifiers belong to White box type II and there are broadly two types: Direct method and indirect method. Direct methods extract rules from datasets directly while indirect methods extract rules from other classification models. This is an indirect way of achieving a higher level of human friendliness for black box type and white box - type I classifiers. These popular direct rule-based classifiers are discussed in section 5.3 and indirect method in section 5.4.

5.3 Rule-based direct methods

The IREP rule learning algorithm [8] is one of the base algorithms and it has been improved later for better results. The RIPPERk [4] is one of the successors of IREP rule learning algorithm and authors have compared its results with C4.5rules. As per their experiments, RIPPERk is very comparative with respect to error rates but much more efficient on large datasets. Rule sets are very friendly. A certain type of prior knowledge can also be communicated to the rule learning system. In [32], the authors have presented a system which is capable of automatically categorizing web documents in order to enable effective retrieval of web information. Based on

the rule learning algorithm RIPPER, they have proposed an efficient method for hierarchical document categorization. In [34] describes TRIPPER – it is a rule induction algorithm and it is an extended version of RIPPER. TRIPPER uses background knowledge in the form of taxonomies over values of features used to describe data. Their experiments show that the rules generated by TRIPPER are generally more accurate and more concise compared to RIPPER.

The Olex [30], is a method of creating rule-based classifiers automatically. It developed using an optimization algorithm. Both positive and negative terms generated by optimization algorithm are used in constructing classification rules. The paper [25] presents extended version of Olex. It is a genetic algorithm, called Olex-GA, for the induction of rule-based text classifiers of the form "classify document d under category c if $t_1 \in d$ or ... or $t_n \in d$ and not ($t_{n+1} \in d$ or ... or $t_{n+m} \in d$) holds", where each t_i is a term. Olex-GA relies on an efficient several individual rule representations per category and uses the F-measure as the fitness function. Results of improved version have been presented in [31]. The Olex system also provides classifiers that are accurate, compact, and comprehensible.

The classifier developed in [3] has used Genetic Programming to evolve classifying agents where each agent evolves a parse-tree representation of a user's particular information need. An agent undergoes a continual training process, therefore, feedback from the user enables the system to learn to the user's long-term information requirements.

The Boolean information retrieval system proposed in [33] has been developed using the Genetic Programming techniques. Using randomly selected terms of relevant documents create Boolean queries. These queries become elements of next population that used for breeding to produce new elements. Boolean queries developed for each category are used as classifiers and they are human interpretable.

In [12] a Genetic Algorithm (GA) is described which is capable of producing accurate compact and human interpretable text classifiers. Document collections are indexed using Apache Lucene and a GA is used to construct Lucene search queries. Evolved search queries are binary classifiers. The fitness function helps producing effective classifiers for a particular category when evaluated against a set of training documents. This system has extended in the paper [11] and they found that a small set of disjunctive Lucene SpanFirst queries meet both accuracy and classifier readability effectively. QuIET in [26] is also automatically generates a set of span queries from a set of annotated documents and uses the query set to categorize unlabeled texts but is not a GA based algorithm.

Classifiers IREP, RIPPERk, TRIPPER, Olex, Olex-GA and other systems outlined above are following direct methods for extracting rule sets.

5.4 Rule-based indirect methods

The paper published in [7] describes an algorithm which generates non-overlapping classification rules from linear support vector machines. This algorithm designed as a constrained-based optimization problem and it extracts classification rules iteratively. For this computationally inexpensive algorithm, authors have discussed number of properties of the algorithm and optimization criteria. These rules can easily understandable to humans unlike support vector machine.

The paper published in [13] presents a method of deriving an accurate rule set using association rule mining. Commonly used rule-based classifiers are preferred small rule sets to large rule sets. But small rule sets are sensitive to the missing values in unseen test data. This paper presents a classifier that is less sensitive to the missing values in unseen test data.

The paper published in [18] describes a human interpretable medical scoring system. They are producing decision lists in which includes a series of if...then... statements. Those if…then… statements groups a high-dimensional feature space into a series of simple, interpretable decision

statements. The authors have introduced a generative model called Bayesian Rule Lists that yields a posterior distribution over possible decision lists. This is an alternative to the CHADS2 score, actively used in clinical practice for estimating the risk of stroke in patients.

The research work in [29] outlines a new hybrid approach for text classification. It has combined a kNN with a rule-based system. kNN classifier has used in building the base model for a given labeled dataset. Rule-base expert system has used to improve the accuracy carefully handing false positives and false negatives. This system can easily fine-tune by humans adding or removing classification rules.

The authors in paper in [5] are trying to break the barrier of human interpretability of Deep Neural Networks (DNN). Concentrating on the video captioning task, authors first extract a set of semantically meaningful topics from the human descriptions that cover a wide range of visual concepts and integrate them into the model with a less interpretable. Then they proposed a prediction difference maximization algorithm to interpret the learned features of each neuron.

6 Conclusion

Classification is a well-matured research field often utilizing a range of applications as a core computing technique. For a certain type of applications, human friendliness is vital. Classifiers belonging to the White box–type II category appear to have the highest level of human-friendliness while the White box –type I seems to have good human interpretability. A number of research attempts to make black box type classifiers more human friendly were found. Further, it is noted that human friendliness of the classifier is critical for certain types of applications.

References

[1] Antman, E.M. et al.: The TIMI risk score for unstable angina/non–ST elevation MI: a method for prognostication and therapeutic decision making. Jama. 284, 7, 835–842 (2000).

[2] Arras, L. et al.: " What is relevant in a text document?": An interpretable machine learning approach. PloS one. 12, 8, e0181142 (2017).

[3] Clack, C. et al.: Autonomous document classification for business. In: Proceedings of the first international conference on Autonomous agents. pp. 201–208 ACM (1997).

[4] Cohen, W.W.: Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning. pp. 115–123 (1995).

[5] Dong, Y. et al.: Improving Interpretability of Deep Neural Networks with Semantic Information. arXiv preprint arXiv:1703.04096. (2017).

[6] Espejo, P.G. et al.: A Survey on the Application of Genetic Programming to Classification. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 40, 2, 121–144 (2010).

[7] Fung, G. et al.: Rule extraction from linear support vector machines. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. pp. 32–40 ACM (2005).

[8] Fürnkranz, J., Widmer, G.: Incremental reduced error pruning. In: Proceedings of the 11th International Conference on Machine Learning (ML-94). pp. 70–77 Morgan Kaufmann (1994).

[9] Gage, B.F. et al.: Validation of clinical classification schemes for predicting stroke: results from the National Registry of Atrial Fibrillation. Jama. 285, 22, 2864–2870 (2001).

[10] Hayes, P.J. et al.: Tcs: a shell for content-based text categorization. In: Artificial Intelligence Applications, 1990., Sixth Conference on. pp. 320–326 IEEE (1990).

[11] Hirsch, L.: Evolved Apache Lucene SpanFirst queries are good text classifiers, (2010).

[12] Hirsch, L. et al.: Evolving Lucene search queries for text classification, (2007).

[13] Hu, H., Li, J.: Using association rules to make rule-based classifiers robust. In: Proceedings of the 16th Australasian database conference-Volume 39. pp. 47–54 Australian Computer Society, Inc. (2005).

[14] Hull, D.A. et al.: Method combination for document filtering. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 279–287 ACM (1996).
[15] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98. 137–142 (1998).
[16] Khan, A. et al.: A Review of Machine Learning Algorithms for Text- Documents Classification. Journal of Advances in Information Technology. 1, 1, 4 (2010).
[17] Knaus, W.A. et al.: APACHE II: a severity of disease classification system. Critical Care Medicine. 13, 10, 818–829 (1985).
[18] Koklu, M. et al.: Applications of Rule Based Classification Techniques for Thoracic Surgery. In: Managing Intellectual Capital and Innovation for Sustainable and Inclusive Society: Managing Intellectual Capital and Innovation; Proceedings of the MakeLearn and TIIM Joint International Conference 2015. pp. 1991–1998 ToKnowPress (2015).
[19] Koza, J.R.: Genetic programming : on the programming of computers by means of natural selection. MIT Press (1992).
[20] Letham, B. et al.: Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics. 9, 3, 1350–1371 (2015).
[21] Lim, W.S. et al.: Defining community acquired pneumonia severity on presentation to hospital: an international derivation and validation study. Thorax. 58, 5, 377–382 (2003).
[22] Mitchell, M.: An introduction to genetic algorithms. MIT Press (1996).
[23] Mori, T.: Superposed Naive Bayes for Accurate and Interpretable Prediction. In: Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on. pp. 1228–1233 IEEE (2015).
[24] Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: Advances in neural information processing systems. pp. 841–848 (2002).
[25] Pietramala, A. et al.: A genetic algorithm for text classification rule induction. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 188–203 Springer (2008).
[26] Polychronopoulos, V. et al.: QuIET: A text classification technique using automatically generated span queries. In: Semantic Computing (ICSC), 2014 IEEE International Conference on. pp. 52–59 IEEE (2014).
[27] Ribeiro, M.T. et al.: Why should i trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144 ACM (2016).
[28] Ridgeway, G. et al.: Interpretable Boosted Naïve Bayes Classification. In: KDD. pp. 101–104 (1998).
[29] Román, J.V. et al.: Hybrid approach combining machine learning and a rule-based expert system for text categorization. Presented at the (2011).
[30] Rullo, P. et al.: Learning rules with negation for text categorization. In: Proceedings of the 2007 ACM symposium on Applied computing. pp. 409–416 ACM (2007).
[31] Rullo, P. et al.: Olex: effective rule learning for text categorization. IEEE Transactions on Knowledge and Data Engineering. 21, 8, 1118–1132 (2009).
[32] Sasaki, M., Kita, K.: Rule-based text categorization using hierarchical categories. In: Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on. pp. 2827–2830 IEEE (1998).
[33] Smith, M.P., Smith, M.: The use of genetic programming to build Boolean queries for text retrieval through relevance feedback. Journal of Information Science. 23, 6, 423–431 (1997).
[34] Vasile, F. et al.: TRIPPER: Rule learning using taxonomies. Advances in Knowledge Discovery and Data Mining. 55–59 (2006).
[35] L. Hirsch and T. Brunsdon, “A comparison of Lucene search queries evolved as text classifiers,” Applied Artificial Intelligence, vol. 32, no. 7, pp. 768-784., 2018.