Transformation of Discriminative Single-Task Classification into Generative Multi-Task Classification in Machine Learning Context

Han Liu, Mihaela Cocea, Alaa Mohasseb and Mohamed Bader
School of Computing, University of Portsmouth
Buckingham Building, Lion Terrace, Portsmouth, United Kingdom PO1 3HE
Email: han.liu@port.ac.uk, mihaela.cocea@port.ac.uk, alaa.mohasseb@port.ac.uk, mohamed.bader@port.ac.uk

***Abstract*—Classification is one of the most popular tasks of machine learning, which has been involved in broad applications in practice, such as decision making, sentiment analysis and pattern recognition. It involves the assignment of a class/label to an instance and is based on the assumption that each instance can only belong to one class. This assumption does not hold, especially for indexing problems (when an item, such as a movie, can belong to more than one category) or for complex items that reflect more than one aspect, e.g. a product review outlining advantages and disadvantages may be at the same time positive and negative. To address this problem, multi-label classification has been increasingly used in recent years, by transforming the data to allow an instance to have more than one label; the nature of learning, however, is the same as traditional learning, i.e. learning to discriminate one class from other classes and the output of a classifier is still single (although the output may contain a set of labels). In this paper we propose a fundamentally different type of classification in which the membership of an instance to all classes(/labels) is judged by a multiple-input-multiple-output classifier through generative multi-task learning. An experimental study is conducted on five UCI data sets to show empirically that an instance can belong to more than one class, by using the theory of fuzzy logic and checking the extent to which an instance belongs to each single class, i.e. the fuzzy membership degree. The paper positions new research directions on multi-task classification in the context of both supervised learning and semi-supervised learning.**

***Keywords*—Data Mining; Machine Learning; Binary Classification; Multi-class Classification; Fuzzy Logic**

I. Introduction

Machine learning has become an increasingly popular approach of artificial intelligence, due to the vast and rapid increase in the size of data. In the form of learning strategies, machine learning can be specialized into two types: supervised learning and unsupervised learning. Supervised learning means learning with a teacher, i.e. data used in the training stage is labeled. In practice, supervised learning can be involved in classification and regression tasks. The main difference between classification and regression is that the output attribute must be discrete for the former type of tasks whereas it must be continuous for the latter type of tasks. Unsupervised learning means learning without a teacher, i.e. data used in the training stage is unlabeled. In practice, unsupervised learning can be involved in association and clustering tasks. Association is aimed at identifying the relationships between different attributes in a quantitative or qualitative way. Clustering is aimed at grouping of different objects on the basis of their similarity.

Classification is one of the most popular tasks of machine learning in practice and has been involved in broad application areas, such as decision making [1], [2], sentiment analysis [3], [4] and ontology engineering [4]. In general, classification can be specialized into binary classification and multi-class classification. The former type of classification means to classify a data instance into one of two given categories whereas the latter type of classification means to classify an instance into one of multiple categories. The rest of this paper focuses on multi-class classification tasks.

In the past years, classification problems have been dealt with in a mutually exclusive way, which means that different classes are assumed to be mutually exclusive and thus each instance can belong to one class only. However, this assumption is not always appropriate in practice, especially when considering the commonly known example that a student can belong to multiple categories. In particular, a student can belong to international students according to their nationality, to full-time student according to the study mode, or to undergraduate student according to the degree level. The above example indicates that nationality, study mode and degree level are three totally independent aspects and thus the three class labels (international student, full-time student and undergraduate student) do not involve mutual exclusion.

To address this issue, the area of multi-label classification has emerged, which allows an instance to be given more than one label. The way to achieve multi-label classification still does not fundamentally change the nature of learning of classifiers, i.e. multi-label classification is still aimed at discriminating one class from other classes, although each class may consists of multiple labels. In addition, both multi-class (single-label) classification and multi-label classification belong to single-task learning. This paper proposes a fundamentally different approach by turning discriminative single-task classification into generative multi-task classification. In particular, the class attribute is transformed into several binary attributes, each of which is corresponding to one of the class labels and is judged on the membership of an instance to it in a generative way. We call this approach generative to stress the opposition to the discriminative approach.

The rest of this paper is organized as follows: Section II provides an overview of multi-class classification in the machine learning context, and identifies the limitations of the existing classification approaches; Section III proposes to adopt a fuzzy classification approach to investigate empirically the extent to which each instance actually belongs to each single class by checking the fuzzy membership degree; in Section IV an experimental study is conducted on five UCI data sets to show the results that support the argumentation that an instance can naturally belong to more than one class; Section V analyses the impact of the proposed new type of classification on real applications; Section VI summarises the contributions of this paper and suggests further directions for this research.

II. Overview of Multi-class Classification

In the context of machine learning, multi-class classification involves classifying an instance into one of three or more categories. This has been involved in broad application areas. In medical applications, multi-class classification can involve judging the type of contact lenses for patients, i.e. hard lenses, soft lenses or no lenses [5]. In pattern recognition, multi-class classification can involve letter recognition [6], handwritten digit recognition [7], Human Hand Movements recognition [8] and emotions recognition [9], [10]. In social media analysis, multi-class classification frequently involves in identifying the attitude of a person, i.e. sentiment classification [3]. Also, movie classification [11] is a special type of item ratings on social media and other platforms. In other social media like Twitter, a tweet could be classified to one of multiple classes, e.g. sports, health and food. Newspaper articles could also be classified to one of multiple categories, e.g. politics and medical care. Also, multi-class classification could be used in book classification, i.e. a book could be put into one of genres, e.g. Fiction, Horror and English.

As mentioned in Section I, multi-class classification tasks are generally undertaken by assuming that different classes are mutually exclusive, which forces each instance to belong to one class only. In practice, this assumption can result in the following issues.

Firstly, the above assumption is not always appropriate in practical applications. For example, a movie on war can belong to both military and history, due to the fact that this type of movies tells a real story that involves soldiers and that happened in the past. In the context of library management, the same book may be used by students from different departments and thus can belong to different subjects. In the context of shopping, the same product may have multiple functions that provide users with different applications. From this point of view, the same product can also be put into different categories. On the other hand, there is also the real case that the classes are mutually exclusive but some instances are complex leading to the difficulty in classifying each of such instances to one category only. For example, in letter recognition, handwritten 'a' and 'd' are very similar and sometimes hard to distinguish; in fact, it may be beneficial

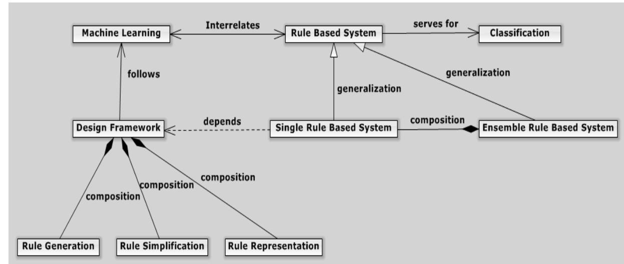

Fig. 1. UML Class Diagram

to identify that an individual tends to write these letters in a very similar way for the purpose of preventing signature fraud.

Secondly, a real data set is dynamically grown in practice. In this context, if a classifier is trained using a data set with a certain number of predefined classes, then it is likely to result in inaccurate classifications when the data set is updated [19], [20]. In fact, it is very likely to occur in practice that a data set is initially assigned a number of class labels by domain experts on the basis of their incomplete knowledge, but the data set is later provided with extra labels following the gain of new knowledge. In this case, the learning of a classifier must be redone once the labels of the data set are updated. In addition, the classifier learned from the initially labelled data set is not useful any more. In the context of software engineering, the above case generally indicates that the classifier is poor in maintainability, re-usability, extendability and flexibility and thus not acceptable in real applications [21].

Thirdly, in the context of object oriented software engineering, different classes can have interrelations such as association, generalization and aggregation [21]. Association means that different classes relate to each other through passing messages between them. For example, Fig. 1 illustrates the association between the two classes (Machine Learning and Rule Based Systems), which indicates that these two subjects are academically linked to each other. Generalization means that classes have hierarchical relationships, i.e. one class can be specialized into a number of subclasses. For example, Fig. 1 illustrates that the class 'Rule Based System' has two subclasses, namely 'Single Rule Based System' and 'Ensemble Rule Based System'. Aggregation means that one class consists of a number of subclasses, i.e. each of the subclasses is a part of its superclass. For example, Fig. 1 illustrates that the class 'Design Framework' consists of three subclasses, namely 'Rule Generation', 'Rule Simplification' and 'Rule Representation'. If the classes defined on a data set involve the relationship of generalization or aggregation, then it would be inappropriate to assume that different classes are mutually exclusive in a multi-class classification problem. In fact, if an instance belongs to a class, then this instance would automatically belong to the superclass of this class.

In recent years, the above issues were addressed in practice by transforming a multi-class classification problem into a multi-label classification problem. The term multi-label generally means that an instance is assigned multiple labels jointly or separately, which is the main difference to the term single-

TABLE I
EXAMPLE OF PT3

Instance ID	Class
1	A
2	B
3	$A \wedge B$
4	$A \wedge B$

TABLE II
EXAMPLE OF PT4 ON LABEL A

Instance ID	Class
1	A
2	$\neg A$
3	A
4	A

TABLE III
EXAMPLE OF PT4 ON LABEL B

Instance ID	Class
1	$\neg B$
2	B
3	B
4	B

TABLE IV
EXAMPLE OF PT5

Instance ID	Class
1	A
2	B
3	A
3	B
4	A
4	B

label meaning that an instance is assigned a single label only. Details can be found in [12], [13], [14], [15], while here we briefly outline the approaches used and their disadvantages.

There are three typical ways of dealing with multi-label classification problems referred to as PT3, PT4 and PT5 respectively, as reviewed in [12]. PT3 is designed to enable that a class consists of multiple labels as illustrated in Table I. For example, two classes A and B can make up three labels: A, B and $A \wedge B$. PT4 is designed to do the labelling on the same data set separately regarding each of the predefined labels as illustrated in Tables II and III. In addition, PT5 is aimed at uncertainty handling. In other words, it is not certain to which class label an instance should belong, so the instance is assigned all the possible labels and is treated as several different instances that have the same inputs but different class labels assigned. An illustrative example is given in Table IV: both instances (3 and 4) appear twice with two different labels (A and B) respectively, which would be treated as four different instances (two assigned A and the other two assigned B) in the process of learning.

However, all of these approaches are still fundamentally aimed at dealing with classification problems on a mutually exclusive basis. In spite of some practical differences between single-label classification and multi-label classification in terms of data labelling, they are still fundamentally the same by learning a multiple-input-single-output classifier through discriminating one class from other classes, i.e. the aim is still to learn a classifier that provides a unique output, in terms of the class to which an instance belongs.

The practical differences mentioned above can be interpreted in the context of granular computing [16]. In particular, the main difference between single-label classification and multi-label classification by PT3 is that the former simply treats each label as a single class, whereas the latter manages to merge multiple labels to make a new class, such as $A \wedge B$, where A and B are two labels, i.e. PT3 involves the use of a granular computing concept referred to as organization by means of composing several parts into a whole [17]. However, like single-label classification, PT3 is still aimed at learning a classifier on a mutually exclusive basis, leading to the output of a unique class to which an instance belongs, although the unique class may consist of multiple labels.

PT4 involves another concept of granular computing referred to as granulation, by means of decomposing a whole into several parts [17]. In other words, PT4 decomposes a complex learning problem into several separate sub-problems, each of which is aimed at binary classification towards judging whether an instance belongs to a particular label. However, each of the sub-problems is resolved in a separate way, which is the same as how a single-label classification problem is dealt with.

PT5 involves the use of rough set theory which is another part of granular computing. In general, rough set theory allows an instance to belong to a label subject to specific conditions, due to the incomplete information available from a data set [16]. In the context of multi-label classification by PT5, different labels may cover common patterns that an instance matches, but it is still unable to judge further if the instance also matches the pattern that is uniquely covered by only one of the labels, due to the insufficient information. In this case, PT5 is designed to assign all the possible labels to the instance with uncertainty, resulting in several instances that have the same inputs but different labels. However, the actual learning of a classifier is still on a mutually exclusive basis towards having several distinct parts of pattern, each of which is uniquely covered by a particular label. Due to the presence of overlapping instances, different parts of the pattern are likely to have intersection, leading to multiple possible labels with different certainty degrees that can be provided as an output set. In order to manage the uncertainty handling, it is designed to allow the output of a set of labels with higher certainty degrees. The output of multiple labels would result in practical issues in classifying test instances and measuring the accuracy. In fact, as shown in Table IV, each test instance only has a single output attribute (the class attribute) of the string type and thus can only be assigned one label. If the output is a set of labels, then it would not be feasible to assign a test instance multiple labels nor to test the accuracy effectively against multiple labels.

In practice, all three approaches have limitation in relation to correlation/independence between labels, as well as computational complexity [15]. In particular, PT3 only works under the assumption that different labels are correlated. In other words, if two labels only coincidentally happen to cover common instances, it does not make much sense to merge them to make a new class, especially when considering the case that different labels may be totally independent of each other and merge of them may result in high coupling in the context of software engineering [21]. Coupling generally means the degree of interdependence between different parts. Also, PT3 may result in the massive classes problem [18]. In fact, a finite set of labels can make up $2^n - 1$ classes, where n is the number of labels.

PT4 only works under the assumption that different labels are independent of each other. If these labels are actually correlated, PT4 may result in reduction of cohesion in the context of software engineering [21], i.e. the degree to which the parts of a whole link together is lower, and thus failing to identify the correlations between different classes. PT5 may result in a massive size of sample, especially when the data set already encounters the massive classes problem due to a large number of labels. In fact, for labels that are independent of each other, the learning to judge the membership or non-membership of an instance to each of these labels needs to be done separately. If the labels are correlated, they need to be defined as several output variables in order to enable identifying their correlations in the training stage and judging the membership or non-membership of an instance to the labels on a correlative basis.

Overall, PT3, PT4 and PT5 all aim to involve learning of a multiple-input-single-output classifier from a single data set, i.e. single-task learning, towards providing a unique output to classify an instance. None of the above ways actually concerns the relationship between different classes/labels, regarding the natural case that an instance belongs to multiple classes/labels. Also, the multi-labelling of instances is done artificially by experts without empirically checking the necessity. The next section presents how fuzzy rule based classification approaches can be used to investigate and show empirically that an instance can belong to multiple classes on the basis of natural patterns learned from data, although the training data is single-labelled by experts. Furthermore, the experimental results presented in Section IV show the necessity to propose a new type of classification, which involves an adaptable way of data labeling and learning of classifiers.

III. Fuzzy Rule Based Classification

This section presents in detail the concepts of fuzzy logic and illustrates the procedure of fuzzy rule based classification approaches. This section also justifies theoretically why and how fuzzy classification approaches can be used to check that an instance can naturally belong to multiple classes with a high level or even full certainty.

A. Key Features

Fuzzy logic is an extension of deterministic logic, i.e. the truth value is ranged from 0 to 1 rather than a binary value. The theory of fuzzy logic is mainly aimed at turning a black and white problem into a grey problem [22]. In the context of set theory, deterministic logic is employed by crisp sets regarding the membership of an element to a set, which means that each element in a crisp set fully belongs to the set with no uncertainty. In contrast, fuzzy logic is employed by fuzzy sets, which indicates that each element in a fuzzy set may just partially belong to the set, i.e. the element belongs to the set to a certain degree referred to as fuzzy membership degree. In practice, the degree of a fuzzy membership can be measured by using a particular fuzzy membership functions such as trapezoid, triangular and Gaussian membership functions [23].

Fuzzy logic involves some logical operations that are slightly different from the operations used in deterministic logic such as conjunction, disjunction and negation. In terms of conjunction, the min function is used to get the smallest value among the values of the given fuzzy variables. For example, if a, b and c are three fuzzy variables with the fuzzy truth values of 0.4, 0.6 and 0.8 respectively, then $a \wedge b \wedge c = min(a, b, c) = 0.4$. For the same example, disjunction involves using the max function instead of the min function, i.e. $a \vee b \vee c = max(a, b, c) = 0.8$. In terms of negation, for the above example, $\neg a = 1 - a = 0.6$. More details on fuzzy operations can be found in [23].

Fuzzy logic is popularly used in rule based systems for dealing with uncertainty [24]. In general, there are three popular types of fuzzy rule based systems namely Mamdani, Sugeno and Tsukamoto [23]. The first two types of fuzzy rule based systems apply to regression problems, as the output from such a system is a real value, and the third type generally applies to classification problems, as the output is a discrete value. As this paper focuses on classification, an illustrative example of a Tsukamoto system is provided below.

The Tsukamoto system has two input variables x_1 and x_2 and one output variable y. The variable x_1 has two linguistic terms 'Good' and 'Bad', x_2 has two linguistic terms 'High' and 'Low' and y has two linguistic terms 'Positive' and 'Negative'. The fuzzy membership functions for the above linguistic terms are defined as follows:

Good: 0.25/1, 0.5/2, 0.75/3, 0.5/4, 0.25/5
Bad: 0.75/1, 0.5/2, 0.25/3, 0.5/4, 0.75/5
High: 0.3/1, 0.4/2, 0.6/3, 0.7/4, 0.5/5
Low: 0.7/1, 0.6/2, 0.4/3, 0.3/4, 0.5/5
Positive: equals to the rule firing strength
Negative: equals to the rule firing strength

There are four rules as follows:
Rule 1: if x_1 is Good and x_2 is High then y= Positive;
Rule 2: if x_1 is Good and x_2 is Low then y= Positive;
Rule 3: if x_1 is Bad and x_2 is High then y= Negative;
Rule 4: if x_1 is Bad and x_2 is Low then y= Negative;

In practice, each rule is derived of its firing strength following the given input values, e.g. if both x_1 and x_2 are assigned the value of 3, then the firing strength of rule 1 is 0.6 as the fuzzy truth values for 'Good' and 'High' are 0.75 and 0.6, respectively. In this case, Rule 1 provides the output value 'Positive' with the fuzzy truth value of 0.6 towards predicting an unseen instance. Each of these four rules work in the same way and the final output value is determined by taking the output value provided by the rule with the highest firing strength. In particular, the firing strengths of Rule 2, Rule 3 and Rule 4 are 0.4, 0.25 and 0.25 respectively. Finally, the output value provided by the fuzzy rule based system is 'Positive' as Rule 1 is of the highest firing strength (0.6), i.e. the fuzzy truth value for 'Positive' is 0.6= $max(0.6, 0.4)$ and the value for 'Negative' is 0.25= $max(0.25, 0.25)$.

B. *Justification*

Fuzzy rule based approaches are used in this paper to show empirically that an instance can belong to multiple classes, due to the nature that fuzzy approaches consider each class to be assigned to an instance with a membership degree, i.e. the extent to which an instance belongs to each single class. In this context, the final classification is made by assigning an unseen instance the class with the highest fuzzy membership degree.

Since fuzzy approaches can show explicitly the fuzzy membership degree of an instance to each single class, we can observe if an instance has a fuzzy membership degree equal to or close to 1 for two or more classes. If the above phenomenon is frequently discovered, then it can strongly support the argumentation that an instance can belong to multiple classes. In addition, even if an instance happens to have a fuzzy membership degree higher than 0.5 for at least two classes, it can still be considered that the instance weakly belongs to both of the two classes.

On the other hand, fuzzy approaches can be used to investigate if classes are mutually exclusive or not. In particular, if classes are mutually exclusive, then the sum of the fuzzy membership degrees for these classes should typically be equal to 1. This could have two different phenomena. One would show that the fuzzy membership degree of an instance is 1 to only one class and 0 to all the other classes. This phenomenon indicates that the instance fully belongs to one class only. The other one would show that an instance belongs to more than one class but the sum of the fuzzy membership degrees for these classes is equal to 1. This phenomenon indicates that the classes are mutually exclusive, but the instance is complex and belongs to different classes to different degrees. However, if the sum of the fuzzy membership degrees for these classes is greater than 1, then the classes are not mutually exclusive.

The next section shows experimentally how the fuzzy membership degree of an instance to each single class can be checked and in what way it can be judged that an instance belongs to two or more classes and if the classes are mutually exclusive or not.

TABLE V
DATA CHARACTERISTICS

Dataset	Attribute Types	Attributes	Instances	Classes
anneal	discrete, continuous	38	798	6
autos	discrete, continuous	26	205	7
heart-c	discrete, continuous	76	920	5
heart-h	discrete, continuous	76	920	5
zoo	discrete, continuous	18	101	7

TABLE VI
RESULTS ON ANNEAL DATASET

ID	class	1	2	3	4	5	U	output
20	3	0	0	1	0	1	0	3
52	3	0	1	1	0	0	0	3
66	5	0	0.99	0	0	0.99	0	2
76	2	0	0.6	1	0	0	0	3
183	3	0	1	1	0	0	0	3
197	3	0	0	1	0	0	0	3
218	2	0	0.8	1	0	0	0	3
296	2	0	1	1	0	0	0	3
329	3	0	1	1	0	0	0	3
380	3	0	0	1	0	1	0	3
457	2	0	0.8	1	0	0	0	3
559	3	0	0.96	1	0	0	0	3
588	2	0	1	0.75	0	0	0	2
606	3	0	1	1	0	0	0	3
670	5	0	0	1	0	1	0	3
681	5	0	1	1	0	1	0	3
682	2	0	1	1	0	0	0	3
696	3	0	1	1	0	1	0	3
700	5	0	0	1	0	1	0	3
721	3	0	0.96	1	0	0	0	3
744	3	0	1	1	0	1	0	3
777	2	0	1	1	0	0	0	3
848	3	0.84	1	0.1	0	0	0	2
857	3	0	1	1	0	0	0	3
870	U	0	1	0.6	0	0	1	U

IV. EXPERIMENTAL STUDY

This experimental study is conducted on five data sets retrieved from the UCI repository [25]. The characteristics of these data sets are presented in Table V. In particular, all these chosen data sets are single labelled. The aim is to show empirically that instances that belong to different classes may have high fuzzy similarity to each other and thus classes can be overlapping by having common instances. In addition, the fuzzy rule induction approach implemented on the KNIME platform is adopted to undertake the experiments [26].

Table VI shows that 25 test instances (selected as representative examples from 200) are judged to belong to multiple classes in accordance with the fuzzy membership degrees measured. In particular, three instances (681, 696 and 744) are judged to belong to three classes and the rest of the instances are judged to belong to two classes. Moreover, Table VI shows 11 instances incorrectly classified according to traditional machine learning principles. However, looking at the columns 3 to 8, it can be noted that the the above 11 instances may not be considered as incorrectly classified. For example, it can be seen for instance 66 that the actual class

TABLE VII
RESULTS ON AUTOS DATASET

ID	class	-1	-2	-3	0	1	2	3	output
17	0	0.77	0	0	0.73	0.20	0	0	-1
130	0	0	0	0	0.5	0	0	0.84	3
141	0	0	0	0	0.75	0	0.7	0	0
150	1	0	0	0	0.74	0.82	0	0	1
172	2	0	0	0	0.83	0	0.68	0	0
181	-1	0.6	0	0	0.67	0	1	0.5	2
197	-1	1	1	0	0	0	0	0	-1

TABLE VIII
RESULTS ON HEART-C DATASET

ID	num	50	50_1	50_2	50_3	50_4	output
11	50	1	1	0	0	0	50
28	50	1	1	0	0	0	50
36	50_1	1	1	0	0	0	50
82	50	1	0.83	0	0	0	50
102	50	0.8	1	0	0	0	50_1
113	50_1	0.81	1	0	0	0	50_1
138	50_1	0.96	1	0	0	0	50_1
194	50_1	1	1	0	0	0	50
271	50_1	0.98	1	0	0	0	50_1
291	50_1	1	1	0	0	0	50

label is '5' and the predicted label is '2' but the instance has the membership degree of 0.99 to both classes (columns 4 and 7). Another example is instance 681 – it is assigned '3' as the predicted class label and the actual label is '5', but according to the fuzzy membership the instance fully belongs to three classes ('2', '3' and '5'). Overall, most of the instances from this data set are assigned '3' as their predicted class label, but none of these cases can really be considered as incorrect classifications when looking at column 5 regarding the fuzzy membership degrees for class '3'. In addition, this table shows to some extent the correlation between classes '2' and '3'.

Table VII shows that 7 instances (selected as representative examples from 200) are judged belonging to multiple classes in accordance with the fuzzy membership degrees measured. In particular, there are 4 instances incorrectly classified according to traditional machine learning principles – see columns 2 and 10. However, these instances may not be considered to have been incorrectly classified when looking at their fuzzy membership degrees to these classes. For example, it can be seen from instance 17 that the predicted label is '-1' and the actual label is '0' but the instance actually belongs to both of the two classes, as the instance has the membership degree of 0.77 to class '-1' and the degree of 0.73 to the class '0'. In addition, this table does not show any obvious correlations between different classes, which may indicate that these classes are independent of each other. However, the opposite phenomenon can be seen in Tables VIII and IX. Thus, the results on the two data sets, 'Heart-c' and 'Heart-h',show correlation between the classes 50 and 50_1.

Table X shows that 26 instances (selected as representative examples from 30) are judged belonging to two or more

TABLE IX
RESULTS ON HEART-H DATASET

ID	num	50	50_1	50_2	50_3	50_4	output
17	50	0.9	1	0	0	0	50_1
68	50	0.6	1	0	0	0	50_1
86	50	1	0.9	0	0	0	50
92	50	1	1	0	0	0	50
93	50	1	1	0	0	0	50
96	50	0.74	1	0	0	0	50_1
107	50	0.7	1	0	0	0	50_1
124	50	1	1	0	0	0	50
127	50	1	1	0	0	0	50
160	50	1	1	0	0	0	50
169	50	1	1	0	0	0	50
214	50_1	1	1	0	0	0	50
228	50_1	1	0.6	0	0	0	50
231	50_1	1	1	0	0	0	50
293	50_1	1	1	0	0	0	50

TABLE X
RESULTS ON ZOO DATASET

ID	type	1	2	3	4	5	6	7	output
1	6	1	0	0	0	1	1	0	6
2	3	0	0	1	0	1	1	0	6
3	6	1	0	0	0	1	1	0	6
6	6	1	0	0	0	1	1	0	6
9	6	1	0	0	0	1	1	0	6
10	6	1	0	0	0	1	1	0	6
18	3	0	0	1	0	1	1	0	6
21	2	0	1	0	0	0	1	0	6
24	4	0	0	0	1	1	0	0	5
31	6	1	0	0	0	1	1	0	6
36	6	1	0	0	0	1	1	0	6
37	2	0	1	0	0	0	1	0	6
41	2	0	1	0	0	0	1	0	6
42	4	0	0	0	1	1	0	0	5
46	5	0	0	0	1	1	0	0	5
47	6	1	0	0	0	1	1	0	6
67	6	1	0	0	0	1	1	0	6
68	6	1	0	0	0	1	1	0	6
71	2	0	1	0	0	0	1	0	6
73	3	0	0	1	0	1	1	0	6
82	3	0	0	1	0	1	1	0	6
83	2	0	1	0	0	0	1	0	6
84	6	1	0	0	0	1	1	0	6
92	3	0	0	1	0	1	1	0	6
97	4	0	0	0	1	1	0	0	5
100	2	0	1	0	0	0	1	0	6

classes in accordance with the fuzzy membership degrees measured. In this table, columns 3-9 represent the IDs of the 7 class labels predefined for the 'Zoo' data set. In particular, '1' represents amphibian, '2' represents bird, '3' represents fish, '4' represents insect, '5' represents invertebrate, '6' represents mammal, and '7' represents reptile. Table X shows 15 instances that are incorrectly classified according to traditional machine learning principles. However, when looking at the fuzzy membership degrees of these instances to different classes, it can be noted that different classes have strong correlations. For example, 11 instances (1, 3, 6, 9, 10, 31, 36, 47, 67, 68 and 84) all belong to three classes, namely amphibian (‘1’), invertebrate (‘5’) and mammal (‘6’). Also, 5 instances (2, 18, 73, 82 and 92) belong to three classes, namely fish (‘3’), invertebrate (‘5’) and mammal (‘6’). In the former case, all 11 instances are considered to have been correctly classified by traditional classification. In contrast, the latter case shows that although all mentioned 5 instances have been incorrectly classified by traditional machine learning, these instances actually belong to both the predicted class label (mammal – ‘6’) and the actual class label (fish – ‘3’), which indicates that none of these 5 instances should be considered to have been incorrectly classified.

A similar phenomenon can also be seen for other instances, i.e 21, 37, 41, 71, 83 and 100. Overall, most of the 26 instances are assigned mammal (‘6’) as their predicted class label, but none of these instances is really incorrectly classified when looking at column 8. In addition, this table shows four different correlations between different classes, i.e. amphibian, invertebrate and mammal; bird and mammal; fish, invertebrate and mammal; and, insect and invertebrate.

V. Discussion

The results shown in Section IV indicate that an instance can belong to multiple classes and that different classes may not be mutually exclusive and can even have correlations among each other. According to the findings obtained from the results, this section analyses the impact of the proposed generative multi-task classification on real applications.

As identified in Section II, traditional multi-class classification, which is dealt with by considering different classes to be mutually exclusive, may result in poor extendability of classifiers. The same issue also arises with multi-label classification. In contrast, the proposed generative multi-task classification addressed this issue by judging the membership of each instance to all classes. Thus, if a new class is added a new learning task can deal with it without affecting the previous learning tasks on the other classes. Consequently, in this context, building a classifier is defined as a multiple learning task, in which each of the single learning tasks involves learning to judge the membership of an instance to a particular class, and these single learning tasks are generative on an independent or correlative basis. In this case, if a new class is added to the data set, then the classifier, which is built on the basis of the original data set, can easily be extended by having another new single learning task on the updated data set. In other words, in the context of multi-class/multi-label classification, each classifier has only a single output, which is one or a subset of the predicted class labels. In contrast, in the context of generative multi-task classification, each classifier can have multiple outputs, each of which is corresponding to a particular class label. In practice, it is critical that a classifier can be easily extended in accordance with the dynamic update of a data set in terms of class labels.

On the other hand, as mentioned in Section II, it is not always appropriate to assume that different classes are mutually exclusive in a classification task. For example, a book can belong to different subject areas. In fact, the nature of a classification problem is on prediction of the value of a discrete attribute. As introduced in [27], a discrete attribute can be specialized into different types, such as nominal, ordinal, string and categorical. For rating problems, the class attribute is of ordinal type. In this case, all the labels make up a whole enumeration so these labels need to be mutually exclusive. Also, classification tasks can be undertaken in practice for the purpose of decision making, which means to make a decision on the selection of one of the class labels. In this case, different classes also need to be mutually exclusive. When a classification task is undertaken for the purpose of categorization of items, it is very likely to occur that different classes have common instances, i.e. an instance can belong to multiple classes. In this case, it is not appropriate to consider that different classes are mutually exclusive. On the basis of the above statement, in the context of traditional multi-class classification, some problems cannot be solved properly in practice. Although multi-label classification was proposed to allow multi-labeling of instances, the nature of classifier learning is still the same by means of learning to discriminate one class from other classes. However, in the context of generative multi-task classification, these problems can be effectively solved by involving each class in a generative single learning task as part of multi-task learning. Also, the outcome of multi-task learning can be used as the basis for secondary learning towards identification of the relationships between different classes, such as generalization and aggregation.

In practice, the proposed generative multi-task classification can be achieved through both supervised and semi-supervised learning. For supervised learning, data labelling needs to be done by transforming the class attribute into several binary attributes, each of which is corresponding to a class label. In this way, experts need to judge on each class whether an instance belongs to it by assigning a truth value (0 or 1). For semi-supervised learning, data sets which have been previously used in multi-class classification tasks, can be used again by transforming the class attribute into several binary attributes. In this way, the transformed data set would have all the binary attributes assigned truth values of 0 or 1. If an instance has its one of the binary attributes assigned 0, this would mean that the instance has not been labelled on the corresponding class. Otherwise, the instance would have been labelled on the corresponding class. On the basis of the transformed data set, fuzzy rule learning approaches can be used to measure the fuzzy membership degrees of each instance to each of the predefined class labels. Through cross validation, each of the instances would be used in turn as a test instance to be measured on the extent to which the instance belongs to each single class. Finally, all these instances would have been assigned fuzzy truth values regarding the fuzzy membership degrees to each of the classes, which can be easily discretised to binary truth values.

VI. Conclusions and future work

In this paper we proposed a fundamentally different approach to address the issue of multi-output for classification tasks. Previous approaches worked with the assumption that different classes need to be mutually exclusive in multi-class or multi-label classification tasks, due to discriminative learning of classifiers. In particular, this paper has proposed to transform a discriminative single-task classification problem into a generative multi-task classification problem. In other words, the class attribute, which is typically involved in a multi-class or multi-label classification task, needs to be transformed into several binary attributes, each of which is corresponding to one of the predefined class labels and could be independent or correlated to the other labels.

The proposed generative multi-task classification is fundamentally different from both single-label (multi-class) classification and multi-label classification discussed in Section II. The proposed type of classification is aimed at judging the membership or non-membership of an instance to each of the predefined class labels, through learning of a multiple-input-multiple-output classifier. In this way of learning, it is possible that an instance does not belong to any of the classes/labels, whereas the other two types of classification are aimed at learning of a multiple-input-single-output classifier that provides a unique output towards classifying an instance. This would be useful in identifying outliers or unusual behaviour which are difficult problems due to class imbalance issues (i.e. fewer examples to learn from).

The above proposal has been investigated empirically by using fuzzy rule based classification approaches on five UCI data sets. The experimental results show that different classes may not be mutually exclusive and thus an instance can belong to multiple classes, especially when the classification task is for the purpose of categorization. Also, this paper has also demonstrated a novel application of fuzzy classification approaches towards identifying the membership of an instance to each of the predefined classes. Consequently, fuzzy approaches are well suited for this new type of classification.

The proposed new type of classification, which is referred to as generative multi-task classification, can be achieved through both supervised learning and semi-supervised learning, as mentioned in Section V; we will explore the use of multi-task classification in future work for both of these contexts. In addition, we will use this new type of classification for the identification of the relationships between different classes such as generalization and aggregation, especially when fuzzy approaches are adopted.

ACKNOWLEDGMENT

The authors acknowledge support for the research reported in this paper through the Research Development Fund at the University of Portsmouth.

REFERENCES

[1] S. Das, S. Kar and T. Pal, ”Robust decision making using intuitionistic fuzzy numbers”, Granular Computing, vol. 1, pp. 1-14, 2016.
[2] Z. Xu and H. Wang, ”Managing multi-granularity linguistic information in qualitative group decision making: an overview”, Granular Computing, vol. 1, no. 1, pp. 21-35, 2016.
[3] B. Liu, ”Sentiment Analysis and Opinion Mining”. Morgan and Claypool Publishers, San Rafael, 2012
[4] W. Perycz and S.M. Chen, ”Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence”. Springer, Heidelberg, 2016.
[5] J. Cendrowska, ”PRISM: An algorithm for inducing modular rules”, International Journal of Man-Machine Studies, vol. 27, pp. 349-370, 1987.
[6] P. W. Frey and D. J. Slate, ”Letter recognition using Holland-style adaptive classifiers”, Machine Learning, vol. 6, no. 2, pp. 161-182, 1991.
[7] M. Buscema, ”MetaNet: The theory of independent judges”, Substance Use and Misuse, vol. 33, no. 2, pp. 439-461, 1998.
[8] J. Wei, Q. Meng and A. Badii, ”Classification of human hand movements using surface EMG for myoelectric control”, Advances in Computational Intelligence Systems: Contributions Presented at the 16th UK Workshop on Computational Intelligence, 79 September, 2016, Lancaster, UK, Advances in Intelligent Systems and Computing, vol. 513, pp. 331-339, Springer, 2016.
[9] H. Binali, C. Wu and V. Potdar, ”Computational approaches for emotion detection in text”, In: 4th IEEE International Conference on Digital Ecosystems and Technologies, 13-16 April 2010, Dubai, United Arab Emirates, pp. 172-177.
[10] Z. Teng, F. Ren and S. Kuroiwa, ”Emotion recognition from text based on the rough set theory and the support vector machines”, In: International Conference on Natural Language Processing and Knowledge Engineering, 30 August- 1 September 2007, Beijing, China, pp. 36-41.
[11] K. Asad, T. Ahmed and M. Rahman, ”Movie popularity classification based on inherent movie attributes using C4.5, PART and correlation coefficient”, In: International Conference on Informatics, Electronics and Vision, 18-19 May 2012, Dhaka, Bangladesh, pp. 747-752.
[12] G. Tsoumakas and I. Katakis, ”Multi-label classification: An overview”, International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 1-13, 2007.
[13] M. R. Boutell, J. Luo, X. Shen and C. M. Brown, ”Learning multi-label scene classification”, Pattern Recognition, vol. 37, pp. 1757-1771, 2004.
[14] G. Tsoumakas, I. Katakis and I. Vlahavas, ”Mining multi-label data”, In: O. Maimon and L. Rokach (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667-685, Springer, 2010.
[15] M. Zhang and H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819-1837, 2014.
[16] Y. Yao, ”Perspectives of granular computing”, In: Proceedings of 2005 IEEE International Conference on Granular Computing, 25-27 July 2005, Beijing, China, pp. 85-90.
[17] H. Hu and Z. Shi, ”Machine learning as granular computing”, In: International Conference on Granular Computing, 17-19 August 2009, Nanchang, China, pp. 229-234.
[18] X. Wang and Y. He, ”Learning from uncertainty for big data: future analytical challenges and strategies”, IEEE Systems, Man and Cybernetics Magazine, vol. 2, no. 2, pp. 26-32, 2016.
[19] S. Suthaharan, ”Big data classification: problems and challenges in network intrusion prediction with machine learning”, ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 70-73, 2014.
[20] L. Wang and C. A. Alexander, ”Machine learning in big data”, International Journal of Mathematical, Engineering and Management Sciences, vol. 1, no. 2, pp. 52-61, 2016.
[21] T. C. Lethbridge and R. Laganiere, ”Object Oriented Software Engineering: Practical Software Development using UML and Java (2nd)”, 2005.
[22] L. Zadeh. ”Fuzzy logic: A personal perspective”. Fuzzy Sets and Systems, vol. 281, pp. 4-20, 2015.
[23] T. Ross. ”Fuzzy Logic with Engineering Applications”. Wiley, West Sussex, 2010.
[24] L. X. Wang and J. M. Mendel. ”Generating fuzzy rules by learning from examples”. IEEE Transactions on Systems, Man and Cybernetics, vol. 22, no. 6, pp. 1414-1427, 1992.
[25] M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.
[26] M. R. Berthold, B. Wiswedel and T. R. Gabriel, ”Fuzzy logic in KNIME: modules for approximate reasoning”, International Journal of Computational Intelligence Systems, vol. 6, no. 1, pp. 34-45, 2013.
[27] P. N. Tang, M. Steinbach and V. Kumar, ”Introduction to Data Mining”. New Jersey: Pearson Education, 2006.