Induction of "In-between" Classes: Learning Vague Concepts

George Potamias[*1,2], and Vassilis Moustakis[1,3]
[*1] Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH),
P.O. Box 1385, 711 10 Heraklion, Crete, Greece,
Phone: +30-81-391693, Fax: +30-81-391601
email: potamias@ics.forth.gr
[2] Department of Computer Science, University of Crete,
P.O. Box 1470, 714 09 Heraklion, Crete, Greece
[3] Department of Production and Management Engineering, Technical University of Crete
University Campus, Kounoupidiana, 73100 Chania, Greece
Phone: +30-821-37323, Fax: +30-821-69410
email: moustaki@dpem.tuc.gr

ABSTRACT: This paper presents a synergistic iterative process, SIR, for resolving between classes assigned to cases. The "vagueness" of concepts, represented by multi-class assignment to cases, is a common phenomenon in the context of concept learning from examples (CLFE) paradigm. The causes could be attributed to the specifics of the application domain, to the poor initial representation or, to the learning heuristics themselves. The methodology presented in this paper take advantage of multi-class assignment in order to improve learning results. Our methodology implements a two-step iterative process: (1) an inductive algorithm runs on the training set of cases, and (2) application of a specially devised set of heuristics aiming to invent new classes, and resolve the conflict presented by multi-class assignment. Thus, "vague" concepts, laying "in-between" the underlying concepts, are learned. Experiments on real-world domains from medicine and finance are presented, and the utility of the SIR process in decision-making tasks is discussed.

KEYWORDS: machine learning, concept learning from examples, vague concepts, medicine, venture capital assessment

INTRODUCTION

The *vagueness* of concepts represented by *multi-class* assignment to cases is a common phenomenon in the context of concept learning from examples (CLFE) paradigm. Most of the CLFE algorithms, like ID3, Quinlan (1986), or CN2, Clark (1989), employ a 'preference' ordering heuristic over the classes (actually the rules for each class) to resolve multi-class assignment. In the case of the C4.5rules system, Quinlan (1993), the heuristic is based on the minimum description length (MDL) principle concluding into a prioritization over the classes and their respective rules. Thus, rules are applied in order and when one of them is fired, its class is assigned to the case. CN2 follows a simple voting strategy: when multiple rules apply then, the class with the highest coverage over the cases is assigned to the case.

The implicit assumption underlying the aforementioned multi-class resolving approaches is about data conclusiveness. But, multi-class case coverage may be caused either by application context dependent factors or by CLFE learning heuristics or, both. The different classes may be tolerated despite the fact that mutual exclusiveness between classes holds. Take for example medical diagnostic domains where, most than often, different diagnoses (diseases) share common properties (symptoms), and initial treatment assignment ranges over multiple alternatives. Treatment assignment to patients is based not only to strictly medical theory but, to external factors as well e.g., the social status of patients, the availability of specific drugs for accompanying the therapeutic treatment and so on. These factors are difficult to capture and record especially when the data set has been compiled in the past with no regard to supporting learning processes. Consequently we are confronted with the problem of *data inclusiveness* and *vague concept* modeling, a task which has been addressed in machine learning literature, Spangler (1989). This reality is conducive to bounding rationality in expert decision-making, Simon (1978). Other authors, Mintzberk (1976), rely on this intrinsic (or hidden) reality to explain the *artistic* trait of expert decision-making. In a study from us, including more than 150

patient real world patient cases, we found that expert decision-making is not consistent and proposed a machine learning approach to rectify and/or model the inconsistency, Moustakis (1992).

The identification of the nature underlying multi-class assignment poses the need of distinguishing between *central* and *borderline* cases. A case may not be central or, *representative* of the concept in which it belongs to; rather it represents a borderline case, Michalski (1993). No matter the reason, multi-class case coverage interferes with learning before even we make an encounter with it in learning outcome; it reflects both to quality and accuracy of output. Alas it is the execution of the learning outcome on cases that makes this reflection even stronger in the sense that, conflicting class assignment can be identified and the corresponding cases may be re-evaluated for their representative power. Concluding the previous discussion we may say that similar concepts hide similar properties that make unique (single-class) case coverage very difficult or even impossible to achieve. Inductive learning systems, such as C4.5, aiming to overcome this problem conclude into *overspecialized* results which most of the times 'obscure' rather than 'reveal' the problem. Consequently, concept similarities, that is multi-class assignment, remain unexplored and difficult to model.

In the present study we elaborate on an *iterative* CLFE process, which cope with multi-class assignment. Objectives are twofold: (a) to present, and demonstrate a methodology for *inventing "in-between"* hidden classes that could explain and model multi-class assignment; and, (b) to identify representative and borderline cases. We support our approach by *coupling* the learning process with multi-class *resolve heuristics* reflecting respective domain dependent *background knowledge*. Our use of the meaning of invention draws from the work of Zytkow, Zytkow (1993). The in-between class invention underlying our methodology takes a more or less declarative interpretation, namely: "*what* other facts about the domain should be stated" as opposite to *discovery* which is mostly linked to a *procedural* interpretation: "*how* should declared facts be considered and evaluated". Declaring a class to be the outcome of different combinations of given classes presents an invention operation because the invented classes base their description on the given classes.

Next section presents our methodology and its implementation by a special *Synergistic Iterative Re-assignment* (SIR) process. Two different implementations of the SIR process, using the similarity-based learning framework and the CN2 algorithm are presented in the second section. We demonstrate our approach in the fourth section where, the different SIR implementations are evaluated on two real-world domains from finance and medicine. We conclude the paper in the last section by discussing the importance of our work for vague concept modeling and decision support, and by suggesting areas for future work.

METHODOLOGY: THE SIR PROCESS

To present our methodology in a formal way we adopt special notation and introduce definitions, which are presented in the lines that follow.

Definition 1. Let $E = \{1,2, \ldots, n\}$, $C = \{c_1, c_2, \ldots, c_k\}$ be the sets of cases and classes, with cardinality $|E| = n$, and $|C| = k$, respectively. The *combined* or, *in-between* classes of C are all the members (c_i, cj) of $CC = C \times C$, denoted with $\boldsymbol{c_{ij}}$; for $i=j$, (c_i, c_i) represents the original single-class c_i. Note that c_{ij}, and c_{ji} represent different combined classes. Furthermore, $\varnothing$ as a member of CC, represents the *null-class*, denoted with $c_{\varnothing}$. A *default rule* equalizes the null class with a class from CC. Thus, the set CC contains a total of k^2 combined classes.

Definition 2. Let E, n, k, and CC as defined in Definition 1. A *state*, $\boldsymbol{s(E)}$, is a n places ordered vector:

$$\boldsymbol{s(E)} = \langle cv_1, cv_2, \ldots, cv_n \rangle ,$$

where, $cv_i \in CC$, is the class (original or combined) assigned to case i, $1 \leq cv_i \leq k^2$. Each case may be assigned to one of the k^2 classes, concluding into a set, $\boldsymbol{S(E)}$, of at most $\boldsymbol{k^{2n}}$ states for E.

Definition 3. Let E, CC, $s(E)$, and $S(E)$ as defined in Definitions 1 and 2. The *algorithm* function, $\boldsymbol{\alpha}$, is defined as follows:

$$\boldsymbol{\alpha}: S(E) \rightarrow S(E)$$

with values, $\boldsymbol{\alpha}(s(E)) = s'(E)$

It is obvious that function α encodes both *induction* and *execution* (deduction) phases of a CLFE algorithm. Function $\boldsymbol{\alpha}$ operates (runs) over the set of training cases. Then, the learning outcome is executed over the same set of cases in order to classify them. So, from a state $s(E)$, a new state $s'(E)$ is reached, with potential multi-class assignment to the cases.

Attempts to rectify multi-class assignment include addition or deletion of attributes, attribute-values and training cases, Baim (1988), Spangler (1989). In all of the approaches multi-class assignment is attributed to external factors and causes and not to domains internal characteristics. Alternatively, we resort on in-between class invention where, multi-class assignment is considered as an intrinsic characteristic of the application domain. Invention leans on *background knowledge* (BK) about the domain. Identification of representative and borderline cases, signals a serendipity effect; a result following the re-assignment of cases to in-between classes.

In-between class invention follows from the intuitive observation that the user will tend to solve the puzzle. This intuition supports class invention strategically. It also positions it as a post-processing operation coupling learning outcome with BK specifics that the learning system is unable to exploit in the first place. The operation ends either when the user achieves a desired learning aspiration threshold or, when further improvement is not possible. Keeping in mind these, rather, intuitive observations, and the definitions presented above, we introduce two additional definitions to support our approach, and to achieve *synergy* between available BK and learning outcome.

Definition 4. The *resolve* function $\boldsymbol{\varepsilon}$ over the set of combined class *CC* is defined as follows:

$$\boldsymbol{\varepsilon}: CC \times CC \rightarrow CC$$

with values, $\boldsymbol{\varepsilon}(c_{xy}, c_{x'y'}) = c_{zw}$

Function ε takes as input two single- or, combined-classes form set *CC*, and outputs a respective class from *CC*. For example, $\varepsilon(c_1, c_2)=c_{12}$, resolves between classes c_1 and c_2 by inventing and forming a new combined class c_{12}, to be assigned to a multi-classified case. As a further example, $\varepsilon(c_1, c_{12})=c_1$, takes as input one single-, c_1, and one combined-class c_{12}, and resolves them to the single-class c_1. When function α operates on a set of cases *E*, it will (potentially) multi-classify them, producing a state *s(E)* of *E*. Operating ε on *s(E)*, we conclude into a new *resolve state* of *E*, $\boldsymbol{\varepsilon}$ ***(s(E))*** .

The instantiation of the resolve function is totally dependent on domain specifics and user's requirements. For example, $\varepsilon(c_1, c_{12})=c_1$, pre-assumes some from of background knowledge that resolves multi-class assignment in favor of their *common* class assignment (i.e., c_1 is a common part for both c_1 and c_{12}). Note that only *2-place* combined classes are allowed. This restriction is posted in order to make the resolve operations computationally tractable and easy to follow. Furthermore, *CC* was defined to hold all the 2-place combined classes from *C*. As it will be shown in the sequel, the user is allowed to consider and define just a subset of these classes, and declare some, and not all, of the possible resolve instantiations. In the current version of the SIR process a simple default rule strategy is followed. As an example assume a 4 class domain where, class c_{34} is not declared as a valid combined class. When a case is pre-assigned to class c_3 and the learning outcome classifies it as c_4, the adopted *default resolve rule* operation assigns to the case its incoming (original) class, c_3.

Now we are in the position to define the core function of the SIR process which, encompasses the consecutive application of both $\boldsymbol{\alpha}$,and $\boldsymbol{\varepsilon}$.

Definition 5. Let *s(E)*, *S(E)*, α, and ε, as defined in Definitions 2,3, and 4. The *transform* function $\boldsymbol{\mu}$ is defined as follows:

$$\boldsymbol{\mu} = \varepsilon \bullet \alpha: S(E) \rightarrow S(E)$$

with values, $\boldsymbol{\mu}(s(E)) = \boldsymbol{\varepsilon}(\boldsymbol{\alpha}(s(E))) = \boldsymbol{\varepsilon}(s'(E)) = s''(E)$,

the resolve state of *E* after applying algorithm α on state *s(E)*, producing state *s'(E)*, and then applying ε on state s'(E)

Function μ implements the kernel of the SIR process. Applying it *iteratively*, a sequence of *ordered* resolved states, $\boldsymbol{s_j}$, are generated where, $\boldsymbol{\mu(s_j) = s_{j+1}}$, and $\boldsymbol{s_{j+1}}$ represents the transform of $\boldsymbol{s_j}$. By Definition 2, a maximum of k^{2n} states of a set of cases *E* could be generated. Without loss of generality, assume that the starting state, s_0, corresponds to the given training set of cases (i.e., with cases pre-assigned to their original single-classes). Then, after at most k^{2n} application of μ, the original state will repeated again and from that point an identical sequence of states will be generated. So, it is natural to consider the state before the repeated one as the *terminating* or, *final state*. In each of the sub-sequent generated states, the invented in-between classes are assigned to respective cases. In the next iteration the inductive procedure, applied on the newly formed state, tries to induce rules or, other descriptions, for the in-between classes. These rules *explain* and *model* the invented in-between classes. Furthermore, at the final state the remaining in-between classes are not only *explainable* but, could be also considered as the only combined classes that are *conceptually valid* for the modeling of the application domain. In Figure 1, the pseudo-code of the SIR process is shown.

Let E the given training set of cases

Set $i = 0$, $s_i = s(E)$, *and* $S(E) = \{s_i\}$
repeat
apply the algorithm function α on s_i and derive $\alpha(s_i)$;
apply the resolve function ε on $\alpha(s_i)$ and derive $\varepsilon(\alpha(s_i)) = \mu(s(_0)) = s_{i+1}$
if $s_{i+1} \in S(E)$ then ***stop***
else
set $i = i + 1$, $S(E) = S(E) \cup \{s_{i+1}\}$

Figure 1. The Synergistic Iterative Re-assignment (SIR) algorithmic process

SIR AND LEARNING ALGORITHMS

The generality in the definition of the algorithm function allows for different implementations of the SIR process itself. That is, different inductive learning algorithms could be used as the base framework for implementing the SIR process. Up to now we have used the CN2 algorithm, and a simple instance-based learning (IBL) process, Aha, et. al. (1991), as our base frameworks.

CN/SIR: SIR can really operate and proceed to "in-between" class invention only if the inductive algorithm allows for *borderline* case identification and induction of respective multi-class rules. CN2, Clark (1989), is such an algorithm as contrast to decision tree algorithms, like plain c4.5 where, over-specialization does not allow overlapping concept identification. CN2 is drawing towards the identification of *significant* concept characteristics and in that sense, leave space for the identification not only of *inter-case* but of *intra-case* concept similarities as well. Inter-case similarities are mostly related to discrimination between concepts operations. Intra-case similarities are mostly related to hidden *causal* relationships between features, Kahneman (1979), Tversky (1982), which are critical for taking a particular decision. Applying SIR process we are able to identify, validate, and finally elaborate these causal relationships into the learning process. So the learning outcome would reflect such causal relationships.

The current CN2/SIR coupling is most suitable for domains where, the classes themselves are *ordered*. For example, consider a domain with three classes, c_1, c_2, and c_3, where, $c_1 < c_2 < c_3$, is a conceptually valid ordering of classes. Then the following valid combined classes are defined, accompanied with respective ground resolve function heuristics:

'c_1 & $c_2 \rightarrow c_{12}$', 'c_2 & $c_3 \rightarrow c_{23}$', 'c_1 & $c_3 \rightarrow c_2$', 'c_1 & $c_{12} \rightarrow c_1$', 'c_1 & $c_{23} \rightarrow c_2$', 'c_{12} & $c_2 \rightarrow c_2$', 'c_{12} & $c_{23} \rightarrow c_2$', 'c_{12} & $c_3 \rightarrow c_2$', 'c_2 & $c_{23} \rightarrow c_3$', 'c_2 & $c_3 \rightarrow c_{23}$', 'c_{23} & $c_3 \rightarrow c_2$'

The semantics behind the class ordering operation '<' are most of the times domain dependent. That is, the user may define the class ordering according to domain specifics and his/her view about the conceptually valid in-between classes that could be formed.

CWRV/SIR: The IBL classification method that we follow is based on an algorithmic process that forms *an ordered vector* for each of the classes. Assume a domain with M attribute-values and C class-values. Then, a number of C, M-places ordered class-vectors will be formed, one for each class-value. Each place of the class-vector holds a *weight* for the respective attribute-value; that is why the method is called ***Class Weighted Relevant Vector*** (CWRV) based classification. Accordingly, an ordered *case vector* is formed, for each of the training or, test cases. The case vector is also an M-places vector, but now the value for each place is *binary*, i.e., in {0,1}, depending on the occurrence or not, of the specific attribute-value in the case.

Various techniques exist for computing attribute-value weights. Here we rely on a very-well known, and widely used, metric borrowed from the *information retrieval* discipline, Salton (1983). The metric is based on a separation between *relevant* and *non-relevant* collections of documents. Considering cases assigned to one class as the relevant documents, and all other cases as the non-relevant ones then, the *contingency matrix*, shown in Table 1, could be formed.

	C_c	$C_{c'},\ c \neq c'$	
$v_{i,j}$ present	$r_{i,j}$	$n_{i,j} - r_{ij}$	$n_{i,j}$
$v_{i,j}$ absent	$E_c - r_{i,j}$	$E - n_{i,j} - (E_c - r_{i,j})$	$E - n_{i,j}$
	E_c	$E - E_c$	E

Table1. Occurrence characteristics for an attribute-value v_{ij} (v_{ij}: value j of attribute i ; **E**: total number of cases; r_{ij} cases, from the total E_c cases assigned to class C_c , contain attribute-value v_{ij} ; $n_{ij} - r_{ij}$ cases assigned to a class c', different from c, contain attribute-value v_{ij})

Then, the following formula, Salton (1981), is elaborated in order to compute the weight of an attribute-value, w_{ij}, with respect to a specific class, c:

$$w_{ij}(c) = \frac{(r_{iij} + 0.5)(E - n_{ij} - (E_c - r_{iij}) + 0.5)}{(n_{ij} - ri_{ij} + 0.5)(E_c - n_{ij} + 0.5)}$$

Based on the above formula, a class-weighted vector is formed for each of the relevant classes. The *classification* of a training or, test (unseen) case is made according to a *similarity* match between the binary vector representative of the case and the respective CWRVs. In the current CWRV/SIR elaboration, the simple *cosine measure* was used for the implementation of the similarity metric. The class that shows, the *highest* similarity match between its respective CRWV and the case binary vector is assigned to the case. Because, all attribute-values are present in the similarity computation it is profound that borderline cases may be identified, making the CWRV classification scheme suitable for the SIR process.

BACKGROUND KNOWLEDGE AND SIR PROCESSES

As it has been stated already, the resolve function, ε, acts as a form of background knowledge which tries to resolve multi-class assignment. Take for example a particular medical domain where, two or more diseases share common symptoms (attribute-values). Then, it is natural to assume that the consideration of a concept formed by the combination of two or more diseases, is *conceptually valid* (with respect to medical theory and practice as well) at least, in the early stages of the diagnostic process. This combined concept may be interpreted as a domain dependent heuristic used for guiding the diagnostic process. The validity of the combined concept is based on domain depended characteristics and should be stated to be a member of the set of all-applicable diseases, i.e., classes in *CC*. On the contrary, the consideration of a concept, formed by the combination of completely separable diseases, at least from early examined patients' clinical manifestations, should not be considered as valid and should not be declared as a conceptually valid class.

Putting SIR in a more general perspective, we can envisage a set of pre-established domain rules as a form of (potentially) incomplete and inconclusive background knowledge. In that sense, the given set of rules plays the role of the resolve function and its elaboration in the SIR process will result to an "*amalgam*" of pure theoretical domain knowledge (reflected in the rule set) with case based knowledge. Such a setting of the SIR process acts as a *knowledge refinement* or, *revision* process, a critical aspect with increasing interest in machine learning research, Saitta (1993), Wrobel (1993), Potamias (1997).

EXPERIMENTS

In this section we examine and demonstrate the behavior and utility of the SIR process. First, we demonstrate the use of SIR on two real world domains namely, *venture capital assessment* (VCA), and treatment of *acute abdominal pain in children* (AAPC). The VCA domain is representative of a diverse range of such domains all of which belong to the area of *financial decision-making* and share a basic characteristic, their concept classes are ordered. The AAPC is an indicative medical decision-making domain where, the alternative therapeutic decisions are not enough separable in the early stages of the diagnostic process.

Venture capital assessment (VCA): Venture capital decision-making represents a complex, ill-structured decision-making task, Tyebjee (1984), presuming the integration of alternative methodologies to solve it. The exemplar presented in this section draws from a real world venture capital assessment discussed in, Siskos (1987). The task is to

rank order 25 firms, seeking venture capital by using nine criteria. Firms are evaluated each with respect to the nine criteria and placed into one of nine classes, see Table 2 below. Assignment of firms to the nine classes was carried out by domain experts and presents the initial state of the firms' data set.

Criterion (attribute)	***Values***
Information security	1, 2, 3
Market trend	1, 2, 3
Market niche/position	1, 2, 3
Conjecture sensibility	1, 2, 3
Result trend	1, 2, 3
Expected dividend rate	1, 2, 3
Quality of management	1, 2, 3
R & D	1, 2, 3, 4
Accessibility to financial markets	1, 2, 3
Initial Ranks	*1, 2, 3, 4, 5, 6, 7, 8, 9*
Conceptually-Valid Ranks	*12, 23, 34, 45, 56, 67, 78, 89*

Table 2. The VCA domain

The ranking of firms to ordered solutions validates the introduction and consideration of in-between solutions. So, instead of the given nine rank classes we may introduce all the ranks between two alternative solutions (see Table 2). The CN2/SIR framework was used for our experimentation.

CN2/SIR on the VCA domain reached a final state after three (3) iterations. The results are summarized in Table 3. In this table note the achieved increase in accuracy (c4.5rules run over the final state of cases, and the learning outcome executed over the same set of cases). This was to be expected, even not warranted, because each SIR iteration 'feeds-back' previous results by examining and resolving conflicting class assignments to cases. The final set of explainable classes (both single and combined) are more (10) than the original given ones (9). This result gives a better understanding to the ranking of firms because *finer* distinctions between them are now available; these distinctions are reflected into the rules induced by the final state of cases.

	#Rules	***#Ranks***	***Ranks***	***Accuracy***
Initial *state*	12	9	*see Table 2*	80 %
Final *CN2/SIR state*	11	10	12, 2, 23, 4, 45, 5, 56, 7, 8, 9	92 %

Table 3. CN2/SIR results on the VCA domain

Acute abdominal pain in children (AAPC): Acute abdominal pain in children encompasses a set of symptoms that cause severe pain, discomfort and increased tenderness in the abdomen of the child. AAPC originates from disorders either in the intra-abdominal or, extra-abdominal areas, De Dombal (1991). [In the current case study we rely on a set of 81 predicates to represent AAPC patient cases. These predicates cover demographic, clinical and laboratory tests' characteristics of patient cases. A total of about 300 past AAPC patient cases was selected randomly from a data base (installed and running in the Pediatric Surgery Clinic, University Hospital at Heraklion, Crete, Greece)].

Management of AAPC patients is based on an explicit protocol, proposed by de Dombal, De Dombal, (1986, 1991). The protocol captures pain specifics, related symptoms and results of laboratory tests. The attending physician needs to diagnose the cause of pain and then, make one of the following decisions, either "`discharge`" the child (in case the cause of the pain is not pathologic), or, to proceed to immediate "`operation`", or, to "`follow-up`" the case for a period of six to eight hours at the end of which, patient condition is re-assessed and the child is either discharged or admitted for operation. In case that an operation decision is followed the physician should already have in mind a spectrum of different potential causes which are to be confirmed or rejected and treated accordingly during the surgery operation.

In a recent case study from us, Potamias (1997), the AAPC background knowledge (provided by experts in the field) achieved an overall accuracy of about 70%, when applied on the given set of cases; a really low figure. Such an uncertainty in the decision-making process should be attributed to the poor domain representation, and mainly to the three single-valued therapeutic decisions. It is really a tricky 'exercise' for the physician to conclude into a clear-cut therapeutic decision in the early diagnostic phase. In special cases where, patients' characteristics are obscured, the three decisions are more or less equivalent. Thus, application of SIR seems a natural direction to follow.

The CWRV/SIR procedure was followed for the AAPC domain. The following conceptual valid combined classes were introduced (suggested by domain experts) accompanied by the respective ground resolve function (`f: follow_up, o: operation, d: discharge`):

```
f & d → f_d
f & o → f_o
f & f → d_f
f & o → d_s
o & f → o_f
o & d → o_d
```

CWRV/SIR reached a final state after five (5) iterations, and the following (single and in-between) classes were induced: `d, s_d, s_f, f_d, f_s,` and `d_f`. The final state fed the c4.5rules system, and the learning outcome was executed over the original set of cases. Inspecting the classification results the following was observed: most (over 80 %) of the '`follow_up`' (`f`) and '`operate`' (`o`) cases, were miss-classified as '`follow_up` OR `discharge`' (`f_d`), and '`operate` OR `discharge`' (`o_d`), respectively.

The above result is not to be considered as disappointing. The combined therapeutic decisions, '`f_d`', or '`o_d`' could be utilized and support the medical decision-making into the early phases of the diagnostic process. Consider an expert decision support system encompassing rules for the aforementioned in-between combined classes. Consultation and classification of a particular AAPC case into a combined class is of major importance for the physician decision-maker. For example, '`follow_up OR discharge`' excludes the '`operate`' decision, which is surely a valuable guide for the decision-making process. It is then, on the responsibility of the physician to resolve and prioritize on the components of the in-between combined class (e.g., based on other external factors like patient's social status or, other).

Because the mixed-up single-classes share a common part with the combined ones, the following resolve prioritizing 'heuristic' could be applied (also suggested by experts in the field): "*combined classes '`f_d`', and '`o_d`', should be transformed into the single classes '`f`', and '`o`', respectively*". After applying the transformation, the final state of cases was used again as input to the c4.5rules algorithm. The learning outcome was executed over the original set of cases and a total accuracy of *92 %* was achieved. This figure is comparable with the original achieved accuracy of *94 %* (i.e., the original data as input to c4.5rules and execution of the learning outcome on the same data set).

CONCLUSION, REMARKS AND FUTURE WORK

We presented, in a formal manner, a Synergistic Iterative Re-assignment (SIR) process for tackling the multi-class assignment problem in a CLFE environment. The synergistic nature of SIR is drawn from the use of a CLFE induction algorithm coupled with specially devised heuristics for resolving between the classes of multi-class assigned cases.

The SIR process operates *iteratively* between the different states of a given set of cases. Given a set of classes then, we can form the set of all possible combinations of them. Of course domain depended, or other, restrictions of this set may apply, concluding into domain dependent and conceptually valid in-between classes. All the different combinations of assigning the cases to the single- or, combined-classes realize the different states of a given set of cases. The SIR process receives as input one of these states and transforms it to a different one. The transformation is realized by three basic iterative operations: (a) application of the CLFE induction algorithm on the set of cases, (b) execution of the learning outcome on the cases, and application of heuristics for resolving between the classes of multi-class assigned cases, and (c) termination of the process when a newly formed state of cases was already generated into a previous iteration.

The instantiation of the SIR process by different types of CLFE algorithms and resolve heuristics, presents a general enough framework for tackling diverse sets of domains where, "*hidden*" similarities between concept classes are obscured and need to be revealed. The SIR process realizes this need by properly inventing classes able to capture class similarities and by that, explain and model vague concepts. Especially, in domains where the set of classes receives an ordering interpretation, the concept of in-between class and the corresponding resolve heuristics are naturally defined. In the current study we presented the coupling of the SIR process with the CN2 algorithm, and an IBL classification method based on information retrieval metrics and techniques.

Pre-established domain knowledge, i.e. some sort of background knowledge BK, may utilize the resolve heuristics or even take the place of them. In such a case, class invention would be based on more "*knowledgable*" sources. In this

paper, we presented two case studies on two real-world domains (from finance and medicine) and the utility of the approach in supporting decision-making operations was indicated.

This setting points to the ongoing research of *knowledge refinement* or, *revision*, where some sort of knowledge is to be rectified with respect to its consistency to a set of cases, Morik (1993), Ourston (1994). In our approach BK acts as a *post-processing* element suggesting specific class assignments after learning is performed. Keeping BK suggestions active, SIR would actually test their *validity* in each iteration. The final state would reflect 'equilibrium' between case based and background knowledge composing an "*amalgam*" between the two sources of knowledge. We plan to apply these ideas to domains where, some sort of BK is available and examine the utility of SIR as a knowledge refinement process. In this course we also plan to test the SIR behavior and trends under the use of different CLFE algorithms.

REFERENCES

Aha, D. W; D. Kibler; M. K. Albert, 1991, Instance-based learning algorithms. *Machine Learning*, 6, pp. 37-66.

Baim, P.W, 1988, "A method for attribute selection in inductive learning systems", *Pattern Analysis and Machine Intelligence* (PAMI), 10(6), pp. 888-896.

Clark, P; T. Niblett, 1989, "The CN2 induction algorithm", *Machine Learning*, 3, pp. 261-283.

De Dombal, F. T, 1986, "How do surgeos assimilate information", *Theoretical Surgery*, 1, pp. 47-54.

De Dombal, F. T, 1991, "*Diagnosis of Abdominal Pain*", Churchill Livingstone.

Kahneman, D; A. Tversky, 1979, "Intutive prediction: biases and corrective procedures*", TIMS Studies in the Management Sciences*, 12, 313-327.

Michalski, R.S, 1993, "Inferential Theory of Learning as Conceptual Basis for Multi-strategy Learning*", Machine Learning*, 11(2/3), pp. 111-152.

Mintzberg, H, 1976, "Planning on the left side and managing on the right", *Harvard Business Review*, July-August pp. 49-58.

Morik, K; G. Potamias; V. Moustakis; G. Charissis, 1993, "Model based learning support to knowledge acquisition: A clinical case study", *Artificial Intelligence in Medicine - Europe* (AIME '93), IOS Press.

Moustakis, V; G. Potamias; L. Gaga; M. Blazadonakis; P. Vassilakis; G. Charissis, 1992, "Bias identification using inductive learning techniques: a medical case study", In B.G. Silverman (ed*.), Expert Judgement, Human Error and Intelligent Systems* workshop, (ECAI '92), pp. 117-123.

Ourston, D; R. Mooney, 1994, "Theory refinement combining analytical and empirical methods", *Artificial Intelligence*, 66, pp. 273-309.

Potamias, G; V. Moustakis; G. Charissis, 1997, "Interactive knowledge based construction and maintenance*", Applied Artificial Intelligence*, 11, pp. 697-717.

Quinlan, J.R, 1986, "Induction of decision trees", *Machine Leaning*, 1, pp. 81-106.

Quinlan, J.R, 1993, *"C4.5: Programs for Machine Learning"*, Morgan Kaufmann, San Mateo, California.

Saitta, L; Bota, ; Neri, F, 1993, "Multi-strategy learning and theory revision", *Machine Learning*, 11(2/3), pp. 153-172.

Salton, G; H. Wu, 1981, "A term weighting model based on utility theory", S.E. Robertson, C.J. van Rijsbergen, P.W. Williams (Eds), *Information Retrieval Research*, Butterworths & Co. Publishers, Boston/MA, USA, pp. 9-22

Salton, G; M.J. McGill, 1983, "*Introduction to Modern Information Retrieval*", McGraw-Hill Book Company, New York.

Simon, H. A, 1978, "Rationality as a process and product of thought", *American Economic Review*, 68(2), pp. 1-16.

Siskos, J; C. Zopounidis, 1987, "The evaluation criteria for the venture capital investment activity: an interactive assessment", *European Journal of Operational Research* (EJOR), 31, pp. 304-313.

Spangler, S; Fayyad, U.M; R. Uthurusamy, 1989, Induction of decision trees from inconclusive data, In *Proceedings of the 6th International Conference on Machine Learning* (ICML '89), pp. 146-150. Morgan Kaufmann.

Tversky, A; Kahneman, D, 1982, "Judgement under Uncertainty : Heuristics and Biases", Cambridge University Press.

Tyebjee, T; A. Bruno, 1984, "A Model of Venture Capitalist Investment Activity, *Management Science*, 30(9), pp. 1051-1066.

Wrobel, S, 1993, On the proper definition of minimality in specialization and theory revision, *In Proceedings of the European Conference on Machine Learning* (ECML93).

Zytkow, J.M, Cognitive autonomy in machine discovery, Machine Learning, 12(1-3), pp. 7-16.