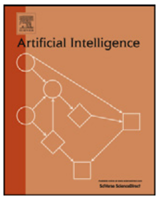
Artificial Intelligence 191–192 (2012) 61–95

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



GAMoN: Discovering M-of-$N^{\{\neg,\vee\}}$ hypotheses for text classification by a lattice-based Genetic Algorithm

Veronica L. Policicchio [*], Adriana Pietramala, Pasquale Rullo

Dept. of Mathematics, University of Calabria, Italy

A R T I C L E I N F O

Article history:
Received 16 November 2011
Received in revised form 4 July 2012
Accepted 11 July 2012
Available online 20 July 2012

A B S T R A C T

While there has been a long history of rule-based text classifiers, to the best of our knowledge no M-of-N-based approach for text categorization has so far been proposed. In this paper we argue that M-of-N hypotheses are particularly suitable to model the text classification task because of the so-called "family resemblance" metaphor: "the members (i.e., documents) of a family (i.e., category) share some small number of features, yet there is no common feature among all of them. Nevertheless, they resemble each other". Starting from this conjecture, we provide a sound extension of the M-of-N approach with negation and disjunction, called M-of-$N^{\{\neg,\vee\}}$, which enables to best fit the true structure of the data. Based on a thorough theoretical study, we show that the M-of-$N^{\{\neg,\vee\}}$ hypothesis space has two partial orders that form complete lattices.
GAMoN is the task-specific Genetic Algorithm (GA) which, by exploiting the lattice-based structure of the hypothesis space, efficiently induces accurate M-of-$N^{\{\neg,\vee\}}$ hypotheses.
Benchmarking was performed over 13 real-world text data sets, by using four rule induction algorithms: two GAs, namely, BioHEL and OlexGA, and two non-evolutionary algorithms, namely, C4.5 and Ripper. Further, we included in our study linear SVM, as it is reported to be among the best methods for text categorization. Experimental results demonstrate that GAMoN delivers state-of-the-art classification performance, providing a good balance between accuracy and model complexity. Further, they show that GAMoN can scale up to large and realistic real-world domains better than both C4.5 and Ripper.

1. Introduction

An M-of-N hypothesis, also called Boolean threshold function, may be thought of intuitively as follows. Given a set of N features, whenever an example satisfies at least M of such features, it is a positive example; otherwise, it is a negative one. That is, an M-of-N hypothesis is a description that involves "counting properties". There is quite a literature on methods for building M-of-N hypotheses. For instance, in [1] algorithms for extracting M-of-N hypotheses from neural networks are reported. M-of-N concepts are also constructed as tests for the induction of decision trees [2–5,7].

However, to the best of our knowledge, no M-of-N-based approach for text classification has been so far proposed. Despite this, we conjecture that M-of-N hypotheses are well suited to model the text classification task.

Text categorization (TC) is aimed at assigning natural language texts to one or more thematic categories on the basis of their contents. It is a difficult task essentially because of two main factors: on one hand, TC has to do with the complexity and richness of the natural language, which allows a concept to be expressed by a variety of constructs and words. This aspect is often amplified by the presence in a category of documents which are not about a single narrow subject with

[*] Corresponding author.
E-mail address: policicchio@mat.unical.it (V.L. Policicchio).

0004-3702/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.artint.2012.07.003

limited vocabulary. On the other hand, the TC task deals with highly dimensional data sets (i.e., with many features). Both such factors concur to make quite unlikely the existence of a set of features, or even a single feature, that occur in all documents of a given category. It may even happen that documents that belong to the same category do not share any content words. However, as argued in [6], the relationship of "family resemblance" holds. That is, documents under the same category share a (usually small) set of N features, yet this set is not present in every document. Instead, each document contains (at least) $M \leqslant N$ of such features, and different documents may not share features at all. That is, the text classification task deals with the kind of data that M-of-N hypotheses are able to explain.

A shortcoming of the M-of-N approach, however, is that its propositions handle positive information only, whereas negative evidence is deemed to play a crucial role in text categorization. This is mainly because natural languages are intrinsically ambiguous, and negation helps to disambiguate concepts – e.g., the word "ball" may ambiguously refer to either the concept "sport" or "dance", whereas the conjunction "ball and not ballroom" much likely refers to "sport".

To overcome this drawback, we extend classical M-of-N hypotheses by negation. In addition, to best fit the true structure of the data, we allow disjunctions of hypotheses. That is, we define a new hypothesis language for text classification, called M-of-$N^{\{\neg,\vee\}}$, which generalizes the classical M-of-N language through negation and disjunction (a preliminary description of the proposed approach can be found in [8]).

In our approach, a classifier is a propositional formula of the form $\mathcal{H}_c = \mathcal{H}_c^1 \vee \cdots \vee \mathcal{H}_c^r$, where each $\mathcal{H}_c^i = p_i$-of-$Pos \wedge \neg n_i$-of-Neg is an *atom* (note that all atoms forming $\mathcal{H}_c$ share the *same* sets *Pos* and *Neg*). Here, *Pos* is the set of *positive* terms, *Neg* the set of *negative* terms, and $p_i \geqslant 0$ and $n_i > 0$ are integers called *thresholds*. The meaning of an atom $\mathcal{H}_c^i$ is: classify document d under category c if *at least* p_i positive terms occur in d and (strictly) *less* than n_i negative terms occur in d. That is, M-of-$N^{\{\neg,\vee\}}$ provides support for explicitly modeling the interactions between positive and negative features. Of course, $\mathcal{H}_c$ classifies document d under c if any of $\mathcal{H}_c^1, \ldots, \mathcal{H}_c^r$ classifies d under c.

The special case of M-of-$N^{\{\neg,\vee\}}$ where a hypothesis is an atom with thresholds $p = n = 1$ is OlexGA [10].

There is a natural ordering in the space of M-of-$N^{\{\neg,\vee\}}$ hypotheses determined by two kinds of subsumption relationships: the *feature* and the *threshold* relationships. The feature relationship is determined by the feature sets *Pos* and *Neg* appearing in a classifier $\mathcal{H}_c$. As an example, assume that $\mathcal{H}_c$ is the atomic classifier p-of-$Pos \wedge \neg n$-of-Neg, with $p = 2$ and $n = 1$. Clearly, the larger *Pos*, the higher the probability that the condition "at least two positive features occur in a document" is satisfied. Dually, the smaller *Neg*, the more likely a document will contain no negative feature in *Neg*. In summary, the larger *Pos*, the smaller *Neg*, the more general $\mathcal{H}_c$. The threshold relationship, in turn, is determined by the thresholds appearing in $\mathcal{H}_c$. For an instance, if in the above classifier we replace $p = 2$ by $p = 1$, we get a new classifier which is more general than the previous one – intuitively, only one instead of two positive features is necessary for classifying a document. These relationships define two hierarchies of hypotheses (more precisely, complete lattices) exploitable for an effective exploration of the hypothesis space. To this end, we provide suitable refinement operators whereby "navigating" the hypothesis lattices.

As argued in [7], the evolutionary approach seems to be particularly suited for the M-of-N learning task, as the global search style of GAs (as opposed to the "one-attribute-at-a-time" of the greedy approach) makes them capable of catching the hidden interactions among attributes that strongly characterize the induction of M-of-N hypotheses. However, the purely non-deterministic nature of conventional genetic operators does not enable the search strategy to benefit of the structure of the hypothesis space. To overcome this drawback, we define a *task-specific* Genetic Algorithm (GA), called GAMoN, relying on specialized evolutionary operators representing a stochastic implementation of the refinement operators defined over the subsumption lattices. At a glance, the following are the main characteristics of GAMoN:

- It relies on a variable-length individual representation, where each individual encodes a candidate classifier (Pittsburgh approach [9]).
- It combines the standard search strategy of GAs with *ad hoc* generalizing/specializing (GS) reproduction operators which exploit the structure of the hypothesis space.
- It dynamically adapts the probability of selecting the GS operators over the standard ones.
- It maintains a number of competing subpopulations.
- It uses the F-measure to assess the fitness of an individual.

Unlike in the classical approach, where the feature space is simply a subset of terms from the vocabulary, the one on which GAMoN builds its hypotheses consists of both a set of positive and a set of negative candidate features. One main issue that in general arises when inducing a classifier is that of selecting the appropriate dimensionality of the feature space, i.e., how many features the classifier can access during the learning process. This is a very important design choice, as the quality of the selected features strongly determines the quality of the learned classifier, especially in text classification, where data sets are usually highly dimensional, noisy and ambiguous. For most systems, the size of the feature space is managed as a tuning parameter, that is, the learning process is rerun over feature spaces of different dimensions and the best results are eventually taken. Unfortunately, this may require very long training times, especially over large data sets. To get over this inconvenience, GAMoN was provided with techniques to automatically detect convenient dimensionality of the feature space. This way, no manual feature selection is preliminarily needed.

GAMoN was designed as a binary classification system. We use the "one-vs-all" approach to produce one (independent) model for each class in a multi-class classification task (this technique is frequently used in multi-label classification, where each example may have more than one label – as it is the case in text classification).

We performed an extensive empirical analysis aimed at comparing the proposed approach with other well known learning algorithms. The experimental results show that GAMoN provides an appealing combination of strengths:

- First, it provides state-of-the-art predictive accuracy in a wide class of problem domains.
- Second, it constructs simple and compact models, thereby facilitating human comprehension of what it has learned.
- Third, it can scale up to large data sets better than other state-of-the-art rule-based classifiers.

GAMoN was implemented in Java as a plug-in of the Weka platform [11].

This paper is organized as follows. In Section 2 we discuss different current learning techniques. In Section 3 we provide an overview of the proposed hypothesis language. In Section 4 we develop a thorough theoretical investigation of its properties. In Section 5 we define suitable refinement operators exploiting the structure of the hypothesis space. In Section 6 we state the learning problem and show its complexity. In Section 7 we describe the GA, paying particular attention to the task-specific operators. In Section 8 we describe the experimental framework applied in our empirical analysis. In Section 9 we report the results of a comparative study of GAMoN against two GA-based rule induction systems, namely, OlexGA [10] and BioHEL [12,13], and three non-GA systems, namely, Ripper [14], C4.5 [15] and the Platt's Sequential Minimal Optimization (SMO) method for linear SVM training [16]. In Section 10 we provide a discussion on the proposed method and relate it to other learning algorithms. Finally, Section 11 concludes the paper.

2. Background

Various supervised machine learning techniques have been applied to document classification. An excellent overview can be found in [17].

SVMs are a class of learning algorithms that showed to be highly accurate in many data mining tasks. In [19,6], Joachims has investigated their application to text classification. The results of the empirical study showed that SVMs are more effective than other learning algorithms, namely, Naive Bayes, Rocchio, C4.5 and k-Nearest Neighbor. Further, linear SVM showed to perform as well as non-linear kernels, but substantially more efficiently.

Naive Bayes (NB) has been a very popular technique to classify texts due to its computational efficiency and simplicity. McCallum and Nigam [20] investigated the two main document representations for NB text classification, the Bernoulli and multinomial. They concluded that the latter is superior in accuracy in most cases. However, one problem with Multinomial NB (MNB) is that, when one class has more training examples than another, it selects poor weights for the decision boundary. One additional problem is that MNB does not model text well. To improve the performance of MNB, Rennie et al. [21] proposed Complement Naive Bayes (CNB). While learning the conditional probability of one class, CNB uses the frequency information pertaining to all other classes (that is, uses negative information).

In a different view, rule learning algorithms have become a successful strategy for classifier induction. Direct methods extract rules directly from data, while indirect methods extract rules from other classification models, such as decision trees (e.g., C4.5 [15]). Representative examples of direct methods include Inductive Rule Learning (IRL) systems, such as FOIL [22] and Ripper [14], and Associative Rule Learning (ARL) systems, such as CMAR [23], CPAR [24] and TFPC [25]. A sub-class of the inductive rule learners is that of Genetics-Based Machine Learning algorithms (GBML) [26], which rely on the Evolutionary Algorithms as search mechanisms. Examples of such systems are XCS [27], SIA [28], GAssist [29] and BioHEL [12,13]. Many GBML systems have explicit generalization/specialization operators [30–35].

The most well-known rule-based classifiers used to learn from texts, notably, Ripper and C4.5, actually originate from non-text data mining (see, e.g., [14,19,36]). Among the few examples of rule-based systems specifically designed to classify texts, we mention the associative classifier NeW [37] and the IRL systems Olex [38] and OlexGA [10]. Olex induces rules consisting of one positive conjunction and (zero or) more negative conjunctions. It relies on a search technique that greedily selects at each step the conjunct, either positive or negative, that maximizes the F-measure over the training set. OlexGA is a GBML which is a special case of GAMoN, where a classifier is an M-of-$N^{\{\neg,\vee\}}$ atom with thresholds $p = 1$ and $n = 1$.[1] A peculiarity of such systems is that of explicitly dealing with negated features.

Even if prior studies found SVMs and Complement Naive Bayes to be particularly effective for text categorization, rule-based text classifiers are often preferred in real-world applications as they provide interpretable models. Readability is indeed a very desirable property of classification models, which allows a human being to understand and possibly modify them based on his a priori knowledge.

However, one drawback with most rule-based systems is the high computational cost, especially on high dimensional data sets. In ARL systems, the time cost for frequent pattern mining may increase very sharply when the size of data set grows. In addition, the high number of rules generated usually requires an additional pruning step where redundant rules are discarded. Also IRL systems typically rely on a two-stage process: a greedy heuristics constructs an initial rule set and,

[1] The Olex and OlexGA suite is downloadable from http://www.mat.unical.it/OlexGA.

then, one or more optimization phases improve compactness and accuracy of the rule set (a similar approach is used for decision tree as well). All this makes it difficult for most rule induction methods to scale up to large and realistic real-world data sets.

3. Language overview

The M-of-$N^{\{\neg,\vee\}}$ representation generalizes the classical notion of M-of-N concepts by allowing negation and disjunction. An M-of-$N^{\{\neg,\vee\}}$ classifier for category c is a propositional formula of the form $\mathcal{H}_c = \mathcal{H}_c^1 \vee \cdots \vee \mathcal{H}_c^r$, where each $\mathcal{H}_c^i = p_i$-of-$Pos \wedge \neg n_i$-of-Neg is an *atom* expressing the following condition: classify document d under category c if *at least* p_i positive features in *Pos* and *less* than n_i negative features in *Neg* occur in d. Integers $p_i \geqslant 0$ and $n_i > 0$ are called *thresholds*. Of course, $\mathcal{H}_c$ classifies document d under c if any among $\mathcal{H}_c^1, \ldots, \mathcal{H}_c^r$ classifies d under c.

Since all atoms forming $\mathcal{H}_c = \mathcal{H}_c^1 \vee \cdots \vee \mathcal{H}_c^r$ share the same sets of features *Pos* and *Neg*, a convenient notation for $\mathcal{H}_c$ is $\langle Pos, Neg, \mathcal{T}\rangle$, where $\mathcal{T} = \{(p_1, n_1), \ldots, (p_r, n_r)\}$ is the set of threshold pairs appearing in the atoms of $\mathcal{H}_c$ ($\mathcal{T}$ is called *threshold set*). For example, (1-of-*Pos* $\wedge$ $\neg$2-of-*Neg*) $\vee$ (2-of-*Pos* $\wedge$ $\neg$3-of-*Neg*) can be simpler represented as $\langle Pos, Neg, \{(1,2), (2,3)\}\rangle$.

As a concrete example, consider the classifier constructed by GAMoN for category "grain" from the Reuters data set:

$$\mathcal{H}_{grain} = \big\langle Pos = \{barley, cereals, corn, grain, maize, rice, sorghum, wheat\},$$
$$Neg = \{acquisition, bank, earning, pay, profit, tax, york\}, \mathcal{T} = \big\{(1,1), (2,2)\big\}\big\rangle.$$

This is a classifier of order 2 (as its threshold set has two elements, i.e., (1, 1) and (2, 2)), with 8 positive features (barley, cereals, etc.) and 7 negative ones (acquisition, bank, etc.). The meaning of $\mathcal{H}_{grain}$ is the following: classify document d under category "grain" if either one of the following conditions holds: (1) d contains (exactly) one positive feature and no negative features, or (2) d contains more than one positive feature and less than two negative ones. That is to say, one single positive feature has no effect on predicting the category "grain" if any negative feature occurs in d, while one single negative feature has no effect in denying the classification of d if more positive features occur in d.

As the above example shows, one beneficial aspect of the M-of-$N^{\{\neg,\vee\}}$ representation is readability. This is a very important feature, as it makes possible for people to visually inspecting and understanding the induced model.

The M-of-$N^{\{\neg,\vee\}}$ hypothesis space has a structure determined by two kinds of subsumption relationships: the *feature* and the *threshold* subsumptions.

Intuitively, positive features are indicative of membership for a category, contrary to negative ones that are indicative of non-membership. Thus, the more elements are in *Pos*, the less are in *Neg*, the more general a classifier $\langle Pos, Neg, \mathcal{T}\rangle$ is (i.e., it classifies more documents). Feature subsumption encodes this intuition. As an example, $\langle\{t_0, t_1\}, \{t_3, t_4\}, \mathcal{T}\rangle$ subsumes $\langle\{t_0, \}, \{t_3, t_4\}, \mathcal{T}\rangle$ and is subsumed by $\langle\{t_0, t_1\}, \{t_3\}, \mathcal{T}\rangle$.

The threshold subsumption relationship is in turn determined by the threshold sets appearing in the classifiers. For an instance, the hypothesis $\langle Pos, Neg, \{(1,1)\}\rangle$ subsumes $\langle Pos, Neg, \{(2,1)\}\rangle$ as only one, instead of two positive features, is necessary for it to classify a document.

Thus, both the above hierarchies capture the intuitive notion of general-to-specific ordering, that is, if $\mathcal{H}_c$ subsumes $\mathcal{H}_c'$ in either hierarchy, then whatever is classified by $\mathcal{H}_c'$ is classified by $\mathcal{H}_c$ as well. One interesting property of such relationships is that they form complete lattices in the hypothesis space (thus, any hypothesis can be reached in the search space).

We can take advantage of this general-to-specific ordering in order to selectively search the hypothesis space. For an instance, if the classifier $\langle Pos, Neg, \{(2,1)\}\rangle$ is too specific (i.e., it covers too few positive examples) it can be generalized either (i) through the threshold subsumption, by replacing the threshold set $\{(2,1)\}$ by one less restrictive, say, $\{(1,1)\}$, or (ii) by the feature subsumption, i.e., by adding some term to *Pos* or removing some term from *Neg*.

Another way of generalizing or specializing a hypothesis is by "interaction" with another one. To this end, we exploit the lattice structure of the hypothesis space. That is, the least upper bound (resp. greatest lower bound) of two hypotheses can be taken, in any of the two lattices, in order to get a more general (resp. specific) one. As an example, given two hypotheses sharing the same threshold sets, say, $\langle\{t_0, t_1\}, \{t_3\}, \{(2,1)\}\rangle$ and $\langle\{t_0, t_4\}, \{t_5\}, \{(2,1)\}\rangle$, we can specialize both by taking the greatest lower bound in the feature subsumption lattice, that is, $\langle\{t_0\}, \{t_3, t_5\}, \{(2,1)\}\rangle$ – a classifier whose sets of positive and negative features are $\{t_0\} = \{t_0, t_1\} \cap \{t_0, t_4\}$ and $\{t_3, t_5\} = \{t_3\} \cup \{t_5\}$, respectively. Likewise, given two hypotheses sharing the same feature sets, say, $\langle Pos, Neg, \{(1,1)\}\rangle$ and $\langle Pos, Neg, \{(2,2)\}\rangle$, we can specialize both by taking the greatest lower bound in the threshold subsumption lattice, that is, $\langle Pos, Neg, \{(2,1)\}\rangle$ – a classifier whose threshold set is $\{max(1,2), min(1,2)\}$ (see Fig. 1). It can be easily verified that both greatest lower bounds are more specific than the respective parents.

As we will see later on this paper, the above concepts are at the basis of the definition of the *refinement operators*. These are the abstract tools for searching the hypothesis space, that find concrete application in the definition of the reproduction operators of GAMoN.

4. Language definition and hypothesis space

Now that we have an intuitive view of the basic ideas, in the next subsections we will provide formal definitions of them. In particular, we will start from the notion of *feature space*, i.e., the set of features which provide the lexicon from

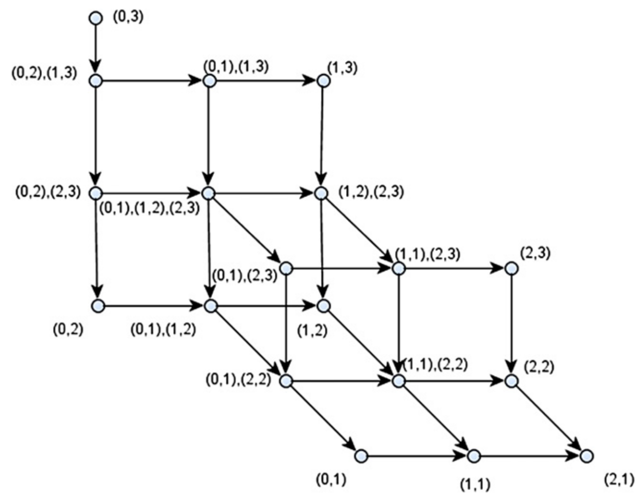


Fig. 1. τ-subsumption lattice with threshold bounds $P = 2$ and $N = 3$.

which hypotheses are built. Then we formalize the M-of-$N^{\{\neg,\vee\}}$ language and define the feature $\succcurlyeq_\phi$ and the threshold $\succcurlyeq_\tau$ subsumption relationships, showing a number of interesting properties. In particular, we will prove that they form complete lattices in the hypothesis space and provide a constructive definition of the meet and the join operators in both lattices. Finally, we will give a deep insight into the structure of the hypothesis space, and show the notion of *decision boundary* for M-of-$N^{\{\neg,\vee\}}$ classifiers.

Note. For the proofs of the propositions reported in this section, the reader is referred to Appendix A.

4.1. Feature space

We are given a set T of training documents (also called “examples”) and a set $\mathcal{C}$ of categories (also called “concepts”). A document is a set of features (also called “terms”), a feature being a sequence of one or more words (or word stems). Each document in T is associated with a category in $\mathcal{C}$. We denote by $T_c \subseteq T$ the training set of c, i.e., the set of training documents associated with category c. We call *vocabulary* the set of features occurring in the documents of T.

Unlike in the classical definitions, where the feature space is simply a subset of the vocabulary, in our definition the feature space consists of both a set of positive and a set of negative features. This is because M-of-$N^{\{\neg,\vee\}}$ hypotheses explicitly models the interaction between positive and negative features, the latter being regarded as “first class citizens”.

Definition 4.1 *(Feature space).* We are given a vocabulary V, a non-negative integer k and a scoring function σ which assigns a score to every feature in V based on its correlation with category c (e.g., CHI Square [39]). Define the *feature space* $\mathcal{F}_c(k)$ (of size k) for category c as the pair $\langle Pos^*_c(k), Neg^*_c(k)\rangle$, where $Pos^*_c(k) \subseteq V$ and $Neg^*_c(k) \subseteq V$ are as follows:

- $Pos^*_c(k)$ is the set of the k highest scoring features in V for category c, according to σ; we say that $t \in Pos^*_c(k)$ is a *candidate positive feature* of c.
- given $Pos^*(k)$, consider the set N of terms co-occurring with positive candidate features within negative examples, i.e., $N = \{t \in V \mid t \notin Pos^*_c(k) \text{ and } (\Theta^+ \cap \Theta(t) \setminus T_c) \neq \emptyset\}$ where $\Theta(t) \subseteq T$ is the set of training documents containing feature t, $\Theta^+ = \bigcup_{t \in Pos^*_c(k)} \Theta(t)$ and T_c is the training set of c. With each feature $t \in N$ we assign a score $\eta(t)$ as follows:

$$\eta(t) = \frac{|\Theta^+ \cap \Theta(t) \setminus T_c|}{|\Theta^+ \setminus T_c| + |\Theta(t) \cap T_c|}.$$

  It can be easily seen that $0 < \eta(t) \leqslant 1$. In particular, a term t occurring in all negative examples and in no positive one containing any positive feature has score $\eta(t) = 1$. On the other hand, $\eta(t) > 0$, $\forall t \in N$ as, by definition, t co-occurs with a candidate positive feature in some negative example. Then, we define $Neg^*_c(k)$ as the set of the best k elements of N according to η; we say that $t \in Neg^*_c(k)$ is a *candidate negative feature* of c.

The rationale behind the above definition is rather intuitive: candidate positive features are supposed to capture most of the positive examples, as they are characterized by high scoring values. On the contrary, candidate negative features, defined as terms co-occurring with positive candidate terms within negative examples, are supposed to discard most of the (potentially) false positive examples. For an instance, if the feature “ball” is a candidate positive and “ballroom” co-occurs with “ball” within some negative examples, “ballroom” becomes a negative candidate feature. Clearly, the higher the scoring $\sigma(t)$ (resp. $\eta(t)$) of a term t, the higher its value as a candidate positive (resp. negative) feature.

4.2. Hypothesis space

A hypothesis is a propositional formula used to describe the examples of a given concept. The hypothesis language we propose in this section is an extension of the M-of-N language, called M-of-$N^{\{\neg,\vee\}}$.

Definition 4.2 *(Hypothesis language).* We are given the feature space $\mathcal{F}_c(k) = \langle Pos^*_c(k), Neg^*_c(k)\rangle$, along with two integers, P and N, called *threshold bounds*. An M-of-$N^{\{\neg,\vee\}}$ *hypothesis* (or “classifier”) for category c over $\mathcal{F}_c(k)$ is inductively defined as follows:

- Basis: p-of-$Pos \wedge \neg n$-of-Neg is an *atom* (or 1-*order* classifier), where $0 \leqslant p \leqslant P$ and $0 < n \leqslant N$ are integers called *positive* and *negative thresholds*, respectively, and $Pos \subseteq Pos^*_c(k)$ and $Neg \subseteq Neg^*_c(k)$ are (possibly empty) sets of features. In particular, *Pos* is the set of *positive* features and *Neg* the set of *negative* features. This classifier classifies a document d under category c if *at least* p positive features occur in d and *less* than n negative features occur in d. A convenient notation for p-of-$Pos \wedge \neg n$-of-Neg is $\langle Pos, Neg, \{(p,n)\}\rangle$.
- Induction: let $\mathcal{H}^1_c = \langle Pos, Neg, \mathcal{T}_1\rangle$ be an r-order classifier and $\mathcal{H}^2_c = \langle Pos, Neg, \mathcal{T}_2\rangle$ an s-order classifier (note that $\mathcal{H}^1_c$ and $\mathcal{H}^2_c$ share the same sets of features). Then $\mathcal{H}_c = \mathcal{H}^1_c \vee \mathcal{H}^2_c$ is a classifier of order $q \leqslant r+s$. $\mathcal{H}_c$ classifies document d under c if either $\mathcal{H}^1_c$ or $\mathcal{H}^2_c$ classifies d under c. A convenient notation for $\mathcal{H}_c$ is $\langle Pos, Neg, \mathcal{T}\rangle$, where $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$.

As already noticed, all atoms forming an r-order classifier hold the same sets *Pos* and *Neg*. That is why we can denote $\mathcal{H}_c = \mathcal{H}^1_c \vee \cdots \vee \mathcal{H}^r_c$ by using the compact notation $\langle Pos, Neg, \mathcal{T}\rangle$, where $\mathcal{T} = \{(p_1,n_1),\ldots,(p_r,n_r)\}$ is the set of all threshold pairs appearing in the atoms of $\mathcal{H}_c$. Clearly, the size of $\mathcal{T}$ is the order of the classifier.

Example 4.1. The 2-order classifier (1-of-$Pos \wedge \neg$2-of-Neg) $\vee$ (2-of-$Pos \wedge \neg$3-of-Neg) can be represented as $\langle Pos, Neg, \{(1,2),(2,3)\}\rangle$.

An atom with $p = 0$ and $n > |Neg|$ acts as an acceptor, while one with $p > |Pos|$ is to be understood as a rejector. An atom $\langle Pos, Neg, \{(1,1)\}\rangle$ coincides with an OlexGA classifier [10]. In general, a (non-acceptor, non-rejector) atom $\langle Pos, Neg, \{(p,n)\}\rangle$ is logically equivalent to the following propositional formula:

$$c \leftarrow (T_1 \vee \cdots \vee T_k) \wedge \neg(T_{k+1} \vee \cdots \vee T_{k+m})$$

where $T_1 \cdots T_k$ are all possible conjunctions made of p positive terms in *Pos*, and $T_{k+1} \cdots T_{k+m}$ are all possible conjunctions made of n negative terms in *Neg*. The rule-based semantics of an r-order classifier $\mathcal{H}_c = \mathcal{H}^1_c \vee \cdots \vee \mathcal{H}^r_c$ is the obvious generalization of the base case: $\mathcal{H}_c$ is equivalent to the union of the rule sets of all $\mathcal{H}^i_c$, $1 \leqslant i \leqslant r$.

We finally provide the definition of hypothesis space.

Definition 4.3. The *hypothesis space* $\mathbf{H}(\mathcal{F}_c(k), P, N)$ is the set of all hypotheses constructible over a feature space $\mathcal{F}_c(k)$ and for given thresholds bounds P and N.

4.3. Ordering the hypothesis space

There is a natural ordering in the hypothesis space determined by two kinds of subsumption relationships, namely, the *feature* and the *threshold* subsumption.

4.3.1. Ordering along the feature dimension

Let $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k)) \subseteq \mathbf{H}(\mathcal{F}_c(k), P, N)$ be the hypothesis subspace consisting of all hypotheses in $\mathbf{H}(\mathcal{F}_c(k), P, N)$ having the same given threshold set $\mathcal{T}$. Hypotheses in $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$ are said τ-*homogeneous*. On $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$ there exists a binary relation that we call *feature-subsumption* (ϕ-*subsumption*, for short).

Definition 4.4 *(Feature-subsumption).* We are given two classifiers in $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$, say, $\mathcal{H}^1_c = \langle Pos_1, Neg_1, \mathcal{T}\rangle$ and $\mathcal{H}^2_c = \langle Pos_2, Neg_2, \mathcal{T}\rangle$. $\mathcal{H}^1_c$ ϕ-*subsumes* $\mathcal{H}^2_c$ (and $\mathcal{H}^2_c$ *is* ϕ-*subsumed by* $\mathcal{H}^1_c$) if both $Pos_2 \subseteq Pos_1$ and $Neg_1 \subseteq Neg_2$ (write $\mathcal{H}^1_c \succcurlyeq_\phi \mathcal{H}^2_c$). $\mathcal{H}^1_c$ is called a ϕ-*generalization* of $\mathcal{H}^2_c$ (and $\mathcal{H}^2_c$ a ϕ-*specialization* of $\mathcal{H}^1_c$).

Example 4.2. According to the above definition, the classifier $\mathcal{H}_c = \langle\{t_0,t_1\},\{t_3,t_4\},\{(1,1)\}\rangle$ ϕ-subsumes $\mathcal{H}'_c = \langle\{t_0\},\{t_3,t_4\},\{(1,1)\}\rangle$ and is ϕ-subsumed by $\mathcal{H}''_c = \langle\{t_0,t_1\},\{t_4\},\{(1,1)\}\rangle$. Intuitively, the former subsumption holds as the classification condition on the positive features “at least one of t_0 and t_1 must occur in d” is clearly weaker than “t_0 must occur in d”, so as (*ceteris paribus*) more documents will be classifier by $\mathcal{H}_c$ than by $\mathcal{H}'_c$. Dually, $\mathcal{H}_c$ is ϕ-subsumed by $\mathcal{H}''_c$ as the condition on the negative features expressed by the latter "t_4 must not occur in d" is weaker than that expressed by the former "neither t_3 nor t_4 can occur in d".

Next we show that if $\mathcal{H}_c^1 \succcurlyeq_\phi \mathcal{H}_c^2$ then $\mathcal{H}_c^1$ classifies all documents classified by $\mathcal{H}_c^2$. In the following, we will denote by $D(\mathcal{H}_c) \subseteq D$ the set of documents classified by $\mathcal{H}_c$, for a document set D.

Proposition 4.1. *Let $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ be two classifiers in $\mathbf{H}_\mathcal{T}(\mathcal{F}_c(k))$. Then $\mathcal{H}_c^1 \succcurlyeq_\phi \mathcal{H}_c^2$ implies $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$.*

The following proposition shows that $(\mathbf{H}_\mathcal{T}(\mathcal{F}_c(k)), \succcurlyeq_\phi)$ is a complete lattice.

Proposition 4.2. $(\mathbf{H}_\mathcal{T}(\mathcal{F}_c(k)), \succcurlyeq_\phi)$ *is a complete lattice. Indeed, for any $\mathcal{H}_c^1, \mathcal{H}_c^2 \in \mathbf{H}_\mathcal{T}(\mathcal{F}_c(k))$, there are both the greatest lower bound $glb_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2)$ and the least upper bound $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2)$ as follows:*

(a) $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos_1 \cup Pos_2, Neg_1 \cap Neg_2, \mathcal{T}\rangle$.
(b) $glb_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos_1 \cap Pos_2, Neg_1 \cup Neg_2, \mathcal{T}\rangle$.

It is easy to recognize that the bottom element of $\mathbf{H}_\mathcal{T}(\mathcal{F}_c(k))$ is $\langle \emptyset, Neg^*(k), \mathcal{T}\rangle$ and the top $\langle Pos^*(k), \emptyset, \mathcal{T}\rangle$.

4.3.2. Ordering along the threshold dimension

Let $\mathbf{H}_\Phi(P,N) \subseteq \mathbf{H}(\mathcal{F}_c(k), P, N)$ be the hypothesis subspace consisting of all hypotheses having the same $\Phi = \langle Pos, Neg\rangle$, with *Pos* and *Neg* over $\mathcal{F}_c(k)$. We say that two classifiers in $\mathbf{H}_\Phi(P,N)$ are *ϕ-homogeneous*. Next we show that a subsumption hierarchy there exists in $\mathbf{H}_\Phi(P,N)$. We call it *threshold-subsumption*, or *τ-subsumption*, for short.

Notation. Since *ϕ-homogeneous* hypotheses share all the same feature sets, in the following, whenever no ambiguity arises, we shall represent a classifier $\langle Pos, Neg, \mathcal{T}\rangle$ simply by $\mathcal{T}$.

Example 4.3. The 2-order classifier $\langle Pos, Neg, \{(1,2),(2,3)\}\rangle$ may be represented simply as $\{(1,2),(2,3)\}$.

Definition 4.5 *(Threshold-subsumption).* Let $\mathcal{T}_1 = \{(p_1,n_1)\}$ and $\mathcal{T}_2 = \{(p_2,n_2)\}$ be two threshold sets of size 1. Then $\mathcal{T}_1$ *τ-subsumes* $\mathcal{T}_2$, denoted $\mathcal{T}_1 \succcurlyeq_\tau \mathcal{T}_2$, if both $p_1 \leqslant p_2$ and $n_1 \geqslant n_2$ hold. More in general, given two threshold sets (of any size), we say that $\mathcal{T}_1$ *τ-subsumes* $\mathcal{T}_2$ if, for each element $(p,n) \in \mathcal{T}_2$, there exists an element $(p',n') \in \mathcal{T}_1$ such that $\{(p',n')\} \succcurlyeq_\tau \{(p,n)\}$.

The relation $\succcurlyeq_\tau$ induces a relation on $\mathbf{H}_\Phi(P,N)$ as follows. Given $\mathcal{H}_c^1 = \langle Pos, Neg, \mathcal{T}_1\rangle$ and $\mathcal{H}_c^2 = \langle Pos, Neg, \mathcal{T}_2\rangle$ in $\mathbf{H}_\Phi(P,N)$, $\mathcal{H}_c^1$ *τ-subsumes* $\mathcal{H}_c^2$ (and $\mathcal{H}_c^2$ is *τ-subsumed by* $\mathcal{H}_c^1$), if $\mathcal{T}_1 \succcurlyeq_\tau \mathcal{T}_2$ (write $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$). $\mathcal{H}_c^1$ is called a *τ-generalization* of $\mathcal{H}_c^2$ (and $\mathcal{H}_c^2$ a *τ-specialization* of $\mathcal{H}_c^1$).

Example 4.4. Given the *ϕ-homogeneous* atoms $\mathcal{H}_c^1 = \{(1,2)\}$ and $\mathcal{H}_c^2 = \{(2,1)\}$, $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ holds as the positive threshold of $\mathcal{H}_c^1$ is smaller than that of $\mathcal{H}_c^2$, whereas the vice versa holds for the negative thresholds. As a more general case, let $\mathcal{H}_c^1 = \{(1,2),(2,1)\}$ and $\mathcal{H}_c^2 = \{(1,1)\}$ be *ϕ-homogeneous* classifiers. Since $\{(1,2)\} \succcurlyeq_\tau \{(1,1)\}$ holds, $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ follows.

Next we show that if $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ then any document classified by $\mathcal{H}_c^2$ is classified by $\mathcal{H}_c^1$ as well.

Proposition 4.3. *Let $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ be two classifiers in $\mathbf{H}_\Phi(P,N)$. Then $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ implies $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$.*

Unlike $\succcurlyeq_\phi$, the binary relation $\succcurlyeq_\tau$ is not a partial order.

Example 4.5. Classifiers $\mathcal{H}_c^1 = \{(1,2)\}$ and $\mathcal{H}_c^2 = \{(1,2),(2,2)\}$ are such that both $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ and $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_c^1$ hold.

Definition 4.6 *(Equivalence, minimality).* Two *ϕ-homogeneous* classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ are *equivalent*, denoted $\mathcal{H}_c^1 \equiv \mathcal{H}_c^2$, if both $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ and $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_c^1$. If a classifier $\mathcal{H}_c$ can be expressed as $\mathcal{H}_c^1 \vee \mathcal{H}_c^2$ such that either $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ or $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_c^1$, then $\mathcal{H}_c$ is *redundant*. Otherwise $\mathcal{H}_c$ is *minimal*. If $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$ is minimal, $\mathcal{T}$ is *minimal*. $\mathcal{H}_c^1$ *strictly* τ-subsumes $\mathcal{H}_c^2$, denoted $\mathcal{H}_c^1 \succ_\tau \mathcal{H}_c^2$, if $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ and not $\mathcal{H}_c^1 \equiv \mathcal{H}_c^2$.

Example 4.6. Classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ of Example 4.5 are equivalent. Classifier $\mathcal{H}_c = \{(1,1),(3,1),(2,2)\}$ is redundant as $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$, where $\mathcal{H}_c^1 = \{(1,1),(2,2)\}$ and $\mathcal{H}_c^2 = \{(3,1)\}$ and, further, $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ holds. On the contrary, $\mathcal{H}_c^1$ is minimal. It can be easily recognized that $\mathcal{H}_c \equiv \mathcal{H}_c^1$ holds.

The notion of equivalence encodes the intuition that equivalent hypotheses provide the same classification behavior. In fact, from Proposition 4.3 it immediately follows that equivalent classifiers do classify the same documents. The next lemma and proposition show that the vice versa holds as well (i.e., classifiers that classify the same documents are equivalent).

Lemma 4.1. *Let $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\hat{\mathcal{H}}_c = \hat{\mathcal{H}}_c^1 \vee \hat{\mathcal{H}}_c^2$ be given. Then, $D(\mathcal{H}_c) \supseteq D(\hat{\mathcal{H}}_c)$ only if ($D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^1)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1)$) and ($D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^2)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^2)$).*

Proposition 4.4. *Let $\mathcal{H}_c$ and $\mathcal{H}'_c$ be two classifiers in $\mathbf{H}_\Phi(P, N)$. Then $D(\mathcal{H}_c) \supseteq D(\mathcal{H}'_c)$ implies $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}'_c$.*

From the above proposition and Proposition 4.3 it immediately follows the following statement.

Corollary 4.1. *Given classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$, $\mathcal{H}_c^1 \equiv \mathcal{H}_c^2$ iff $D(\mathcal{H}_c^1) = D(\mathcal{H}_c^2)$.*

Next we show a number of further interesting properties of classifiers.

Proposition 4.5. *Let $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$ be given. Then:*

1. *$\mathcal{H}_c$ is redundant iff there exist $(p_i, n_i), (p_j, n_j) \in \mathcal{T}$ such that $\{(p_i, n_i)\} \succcurlyeq_\tau \{(p_j, n_j)\}$.*
2. *$\mathcal{H}_c$ is minimal iff $\mathcal{T} = \{(p_1, n_1), \ldots, (p_r, n_r)\}$ is such that $p_i < p_j$ and $n_i < n_j$, or vice versa, for each $i, j \in [1, r]$.*
3. *If $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$, then $\mathcal{H}_c \equiv \mathcal{H}_c^1$.*

Example 4.7. According to Part 1 of Proposition 4.5, the classifier $\mathcal{H}_c = \{(1,1), (2,1), (2,2)\}$ is redundant, as $\{(1,1)\} \succcurlyeq_\tau \{(2,1)\}$, while $\mathcal{H}_c^1 = \{(1,1), (2,2)\}$ is minimal. It is easily verified that $\mathcal{H}_c^1$ satisfies the condition $p_1 < p_2$ and $n_1 < n_2$ of Part 2 of Proposition 4.5. Since $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$, where $\mathcal{H}_c^2 = \{(2,1)\}$, and $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$, $\mathcal{H}_c \equiv \mathcal{H}_c^1$ follows from Part 3 of Proposition 4.5.

Another interesting property of $\mathbf{H}_\Phi(P, N)$ is that, for any two classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ in it, there exists another classifier $\mathcal{H}_c$ in it which is a τ-specialization of both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ that classifies exactly the documents classified by both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ (i.e., $\mathcal{H}_c$ is equivalent to the logical AND of $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$). We denote such a classifier by $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$. Next we provide the (constructive) definition of $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$. As we will see shortly after, this definition is a preliminary step for showing that $\succcurlyeq_\tau$ forms a complete lattice over the set of minimal classifiers.

Definition 4.7 *(AND of classifiers).* Given $\mathcal{H}_c^1, \mathcal{H}_c^2 \in \mathbf{H}_\Phi(P, N)$, $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$ is the classifier inductively defined as follows:

- Basis: if $\mathcal{H}_c^1 = \{(p_1, n_1)\}$ and $\mathcal{H}_c^2 = \{(p_2, n_2)\}$ are atoms, then $and(\mathcal{H}_c^1, \mathcal{H}_c^2) = \{(p, n)\}$, where $p = Max\{p_1, p_2\}$ and $n = Min\{n_1, n_2\}$.
- Inductive step: if $\mathcal{H}_c^1 = \mathcal{H}_c^{1,1} \vee \mathcal{H}_c^{1,2}$ and $\mathcal{H}_c^2 = \mathcal{H}_c^{2,1} \vee \mathcal{H}_c^{2,2}$, then $and(\mathcal{H}_c^1, \mathcal{H}_c^2) = \mathcal{H}_1 \vee \mathcal{H}_2 \vee \mathcal{H}_3 \vee \mathcal{H}_4$, where $\mathcal{H}_1 = and(\mathcal{H}_c^{1,1}, \mathcal{H}_c^{2,1})$, $\mathcal{H}_2 = and(\mathcal{H}_c^{1,1}, \mathcal{H}_c^{2,2})$, $\mathcal{H}_3 = and(\mathcal{H}_c^{1,2}, \mathcal{H}_c^{2,1})$ and $\mathcal{H}_4 = and(\mathcal{H}_c^{1,2}, \mathcal{H}_c^{2,2})$.

Example 4.8. If $\mathcal{H}_c^1 = \{(1,2)\}$ and $\mathcal{H}_c^2 = \{(2,3)\}$, then $and(\mathcal{H}_c^1, \mathcal{H}_c^2) = \{max\{1,2\}, min\{2,3\}\} = \{2,2\}$ (base step of the definition). Intuitively, $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$ is more specific than $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ as the higher the positive threshold, the lower the negative one, the more specific an atom is. As another example, if $\mathcal{H}_c^3 = \{(1,1), (2,3)\}$ and $\mathcal{H}_c^4 = \{(0,1), (2,2)\}$, then $and(\mathcal{H}_c^3, \mathcal{H}_c^4) = \{(1,1), (2,1), (2,2)\}$ (inductive step of the definition). Notice that $and(\mathcal{H}_c^3, \mathcal{H}_c^4)$ is not minimal.

Proposition 4.6. *Given $\mathcal{H}_c, \hat{\mathcal{H}}_c \in \mathbf{H}_\Phi(P, N)$, the classifier $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ is such that (1) $and(\mathcal{H}_c, \hat{\mathcal{H}}_c) \in \mathbf{H}_\Phi(P, N)$, (2) $D(and(\mathcal{H}_c, \hat{\mathcal{H}}_c)) = D(\mathcal{H}_c) \cap D(\hat{\mathcal{H}}_c)$, and (3) $\mathcal{H}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ and $\hat{\mathcal{H}}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$.*

Example 4.9. In Example 4.8 we have seen that $and(\mathcal{H}_c^1, \mathcal{H}_c^2) = \{(2,2)\}$, for $\mathcal{H}_c^1 = \{(1,2)\}$ and $\mathcal{H}_c^2 = \{(2,3)\}$. Hence, $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$ classifies a document d if d contains $x \geqslant 2$ positive features and $y < 3$ negative features; it is immediately recognized that a document satisfying such a condition is classified by both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$. On the other hand, d is classified by both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ if it contains $x \geqslant max(1,2)$ positive features and $y < min(2,3)$ negative features, that is, if d is classified by $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$.

The above result shows that the inclusion of the "$\wedge$" operator in the definition of classifier would not increase the expressivity of the language (i.e., it would be redundant).

Now we turn our attention to minimal classifiers. The following proposition shows a key result, that is, the uniqueness of the minimal classifier for an equivalence class.

```
Functions ⊓(T_1, T_2) and ⊔(T_1, T_2)

1. function Minimize($\mathcal{T}$)
2.    drop from $\mathcal{T}$ each $(p,n)$ s.t. $\exists(p',n') \in \mathcal{T}$ s.t. $(p',n') \succcurlyeq_\tau (p,n)$.
4. return $\mathcal{T}$.

5. function $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$
6.    return Minimize($\mathcal{T}_1 \cup \mathcal{T}_2$);

8. function $\sqcap(\mathcal{T}_1, \mathcal{T}_2)$
9.    $\mathcal{T} = \emptyset$;
10.    for each $(p,n) \in \mathcal{T}_1$
11.      for each $(p',n') \in \mathcal{T}_2$
12.         $\mathcal{T} = \mathcal{T} \cup \{(Max\{p,p'\}, Min\{n,n'\})\}$;
13. return Minimize($\mathcal{T}$);
```

Fig. 2. Computation of $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$ and $glb_\tau(\mathcal{T}_1, \mathcal{T}_2)$.

Proposition 4.7. *Any equivalence class into which is partitioned the hypothesis subspace $\mathbf{H}_\Phi(P,N)$ by the relation $\equiv$ has a unique minimal classifier.*

We denote by $Min(\mathcal{H}_c)$ the minimal classifier of the equivalence class of $\mathcal{H}_c$. From now on, we will restrict our attention to the set of minimal classifiers $\mathbf{M}_\Phi(P,N) \subseteq \mathbf{H}_\Phi(P,N)$. It is immediate to recognize that the restriction of the binary relation $\succcurlyeq_\tau$ to $\mathbf{M}_\Phi(P,N)$ is a partial order. More precisely, it is a complete lattice.

Proposition 4.8. *The poset $(\mathbf{M}_\Phi(P,N), \succcurlyeq_\tau)$, where $\mathbf{M}_\Phi(P,N)$ is the set of the minimal classifiers in $\mathbf{H}_\Phi(P,N)$, is a complete lattice. Indeed, for any two elements $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ of $\mathbf{M}_\Phi(P,N)$, there are both the greatest lower bound $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2)$ and the least upper bound $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2)$ as follows:*

(a) *$lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$, that is, the least upper bound of $\mathcal{H}_c^1$, $\mathcal{H}_c^2$ is the minimal classifier of the equivalence class of $\mathcal{H}_c^1 \vee \mathcal{H}_c^2$.*
(b) *$glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$, that is, the greatest lower bound of $\mathcal{H}_c^1$, $\mathcal{H}_c^2$ is the minimal classifier of the equivalence class of $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$.*

Example 4.10. The τ-subsumption lattice, for threshold bounds $P = 2$ and $N = 3$, is depicted in Fig. 1. How we can see, there are 19 classifiers; the most general one is $\{(0,3)\}$ and the most specific one is $\{(2,1)\}$. Further, the maximum order of a classifier is 3 (the order of $\{(0,1),(1,2),(2,3)\}$).

We conclude this section by providing a constructive definition of both lub_τ and glb_τ. Let $\mathcal{H}_c^1 = \langle Pos, Neg, \mathcal{T}_1\rangle$ and $\mathcal{H}_c^2 = \langle Pos, Neg, \mathcal{T}_2\rangle$. Now, by Proposition 4.8, $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$, that is, $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\langle Pos, Neg, \mathcal{T}_1 \cup \mathcal{T}_2\rangle)$ (by Definition 4.2), that is, $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos, Neg, Min(\mathcal{T}_1 \cup \mathcal{T}_2)\rangle$, where $Min(\mathcal{T}_1 \cup \mathcal{T}_2)$ is obtained from $\mathcal{T}_1 \cup \mathcal{T}_2$ simply by discarding every (p,n) such that there exists $(p',n') \in \mathcal{T}_1 \cup \mathcal{T}_2$ such that $\{(p',n')\} \succcurlyeq_\tau \{(p,n)\}$ (immediate from Proposition 4.5, Part 1). We denote $Min(\mathcal{T}_1 \cup \mathcal{T}_2)$ by $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$, so that $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos, Neg, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\rangle$. Likewise, by Proposition 4.8, we have that $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$. We denote by $\sqcap(\mathcal{T}_1, \mathcal{T}_2)$ the threshold set constructed by using Definition 4.7 and then minimized as shown above, so as $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos, Neg, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\rangle$.

Proposition 4.9. *Let $\mathcal{H}_c^1 = \langle Pos, Neg, \mathcal{T}_1\rangle$ and $\mathcal{H}_c^2 = \langle Pos, Neg, \mathcal{T}_2\rangle$ be two (minimal) classifiers in $\mathbf{M}_\Phi(P,N)$. Then*

$$\underset{\tau}{lub}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \big\langle Pos, Neg, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\big\rangle,$$
$$\underset{\tau}{glb}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \big\langle Pos, Neg, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\big\rangle$$

where $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$ and $\sqcap(\mathcal{T}_1, \mathcal{T}_2)$ are constructively defined as shown in Fig. 2.

A proof of correctness of the algorithms of Fig. 2 is reported in Appendix A.

4.4. The minimal hypothesis space

In the previous subsection we defined the notion of minimal classifier as the representative hypothesis of an equivalence class. Minimality is a desirable property of classifiers as, by guaranteeing the uniqueness of representation, imposes an ordered structure within the hypothesis space. For this reason, we restrict ourselves to *minimal* classifiers.

Definition 4.8. Let the feature space $\mathcal{F}_c(k) = \langle Pos^*(k), Neg^*(k)\rangle$ and the threshold bounds P and N be given. The *minimal hypothesis space* constructible over $\mathcal{F}_c(k)$, for the given P and N values, is $(\mathbf{M}(\mathcal{F}_c(k), P, N), \succeq_\tau, \succeq_\phi)$, where

$$\mathbf{M}\big(\mathcal{F}_c(k), P, N\big) = \bigcup_{\Phi} \mathbf{M}_\Phi(P, N) \quad \text{s.t.} \quad \Phi \in \big\{\langle Pos, Neg\rangle \,\big|\, Pos \subseteq Pos^*(k),\ Neg \subseteq Neg^*(k)\big\}.$$

Thus, a minimal hypothesis space is uniquely determined by k, P and N.

Using the previously defined notational convention, in the following we will denote by $\mathbf{M}_\tau(\mathcal{F}_c(k))$ the set of (minimal) classifiers in $\mathbf{M}(\mathcal{F}_c(k), P, N)$ with threshold set $\mathcal{T}$. It is immediate to recognize that, given the minimal threshold set $\mathcal{T}$, $\mathbf{M}_\tau(\mathcal{F}_c(k))$ and $\mathbf{H}_\tau(\mathcal{F}_c(k))$ coincide, as both consist of all (minimal) classifiers with threshold set $\mathcal{T}$ constructible over $\mathcal{F}_c(k)$. It turns out that $(\mathbf{M}_\tau(\mathcal{F}_c(k)), \succeq_\phi)$ and $(\mathbf{H}_\tau(\mathcal{F}_c(k)), \succeq_\phi)$ coincide as well.

Next we discuss on the structure of $(\mathbf{M}(\mathcal{F}_c(k), P, N) \succeq_\tau, \succeq_\phi)$, as determined by the two subsumption relations. Since the ϕ-subsumption and the τ-subsumption lattices are the basic building blocks of a minimal hypothesis space, we start our discussion by preliminarily showing the size of such lattices.

4.4.1. The size of the two types of lattice

A ϕ-subsumption lattice $\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k))$ consists, for a given $\mathcal{T}$, of all hypotheses that can be built over a given feature space $\mathcal{F}_c(k)$, each hypothesis corresponding to a particular choice of the sets *Pos* and *Neg* over $\mathcal{F}_c(k)$. It is immediate to recognize the following fact.

Fact 4.1. *The size of $\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k))$ is equal to the number of sets Pos and Neg constructible over the feature space $\mathcal{F}_c(k) = \langle Pos^*(k), Neg^*(k)\rangle$, that is, $|\mathbf{H}_\mathcal{T}(\mathcal{F}_c(k))| = 2^{2k}$.*

A τ-subsumption lattice $\mathbf{M}_\Phi(P, N)$ consists, for a given $\Phi = \langle Pos, Neg\rangle$, of all hypotheses that can be built for the given threshold bounds P and N, each hypothesis corresponding to a particular threshold set satisfying P and N. The next lemma and proposition show both the size of $\mathbf{M}_\Phi(P, N)$ and the maximum order of a classifier.

Lemma 4.2. *Given threshold bounds P and N, along with $k \leqslant Min(P+1, N)$, let $T_k^+ = \{p_1, \ldots, p_k\}$ and $T_k^- = \{n_1, \ldots, n_k\}$ be sets of integers, where $\forall i \leqslant k$, $0 \leqslant p_i \leqslant P$ and $0 < n_i \leqslant N$. Then there exists a unique subset $S \subseteq T_k^+ \times T_k^-$ having size k which is a minimal threshold set.*

Proposition 4.10. *Given the threshold bounds P and N,* (1) *the maximum order of a classifier in $\mathbf{M}_\Phi(P, N)$ is $Min(P+1, N)$, and* (2) *the number of minimal threshold sets that can be constructed for the given bounds is*

$$\lambda(P, N) = \sum_{j=1}^{Min\{P+1,N\}} \binom{P+1}{j}\binom{N}{j}. \tag{1}$$

4.4.2. The landscape from the τ-subsumption perspective

Given threshold bounds P and N, let us consider two lattices $(\mathbf{M}_\Phi(P, N), \succeq_\tau)$ and $(\mathbf{M}_{\Phi'}(P, N), \succeq_\tau)$, where $\Phi = \langle Pos, Neg\rangle$ and $\Phi' = \langle Pos', Neg'\rangle$. By Proposition 4.10, they have the same number $\lambda(P, N)$ of classifiers, i.e., all those constructible for the given P and N. Hence, there is a one to one correspondence g between the classifiers of $\mathbf{M}_\Phi(P, N)$ and those of $\mathbf{M}_{\Phi'}(P, N)$, two related classifiers having the same threshold sets. Since the τ-subsumption relation among classifiers is determined by the τ-subsumption relation among the respective threshold sets (see Definition 4.5), clearly $\mathcal{H}_c^1 \succeq_\tau \mathcal{H}_c^2$ holds in $(\mathbf{M}_\Phi(P, N), \succeq_\tau)$ iff $g(\mathcal{H}_c^1) \succeq_\tau g(\mathcal{H}_c^2)$ holds in $(\mathbf{M}_{\Phi'}(P, N), \succeq_\tau)$. That is, the two lattices are isomorphic. Further, all classifiers in $\mathbf{M}_\Phi(P, N)$ share the feature sets $\langle Pos, Neg\rangle$, while those in $\mathbf{M}_{\Phi'}(P, N)$ share the feature sets $\langle Pos', Neg'\rangle$, so as $\mathbf{M}_\Phi(P, N)$ and $\mathbf{M}_{\Phi'}(P, N)$ are disjoint. Since the number of different Φs (i.e., pairs of sets *Pos* and *Neg*) constructible over a given feature space $\mathcal{F}_c(k)$ is 2^{2k}, we may conclude that $(\mathbf{M}(\mathcal{F}_c(k), P, N), \succeq_\tau)$ has a structure made of 2^{2k} isomorphic, disjoint lattices $(\mathbf{M}_\Phi(P, N), \succeq_\tau)$, each of size $\lambda(P, N)$. For an instance, given $P = 2$ and $N = 3$, $(\mathbf{M}(\mathcal{F}_c(k), 2, 3), \succeq_\tau)$ will consists of 2^{2k} lattices whose structure is that depicted in Fig. 1.

Fact 4.2. *The partial order $(\mathbf{M}(\mathcal{F}_c(k), P, N), \succeq_\tau)$ consists of 2^{2k} isomorphic, disjoint lattices $(\mathbf{M}_\Phi(P, N), \succeq_\tau)$, each of size $\lambda(P, N)$.*

4.4.3. The landscape from the ϕ-subsumption perspective

The ϕ-subsumption perspective is of course dual to the τ-subsumption one. Consider two lattices $(\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k)), \succeq_\phi)$ and $\mathbf{M}_{\mathcal{T}'}(\mathcal{F}_c(k), \succeq_\phi)$, for any $\mathcal{T}$, $\mathcal{T}'$. As stated by Fact 4.1, their size is 2^{2k}. Since the relationship $\succeq_\phi$ among classifiers is determined only by the inclusion relationship among the respective sets of features (see Definition 4.4), the structure of the above lattices does not depend on $\mathcal{T}$. Hence, $\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k))$ and $\mathbf{M}_{\mathcal{T}'}(\mathcal{F}_c(k))$ are isomorphic under $\succeq_\phi$. Since any hypothesis in $\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k))$ has threshold set $\mathcal{T}$ and any hypothesis in $\mathbf{M}_{\mathcal{T}'}(\mathcal{F}_c(k))$ has threshold set $\mathcal{T}'$, $\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k))$ and $\mathbf{M}_{\mathcal{T}'}(\mathcal{F}_c(k))$ are disjoint. Therefore, in the hypothesis space $(\mathbf{M}(\mathcal{F}_c(k), P, N), \succeq_\phi)$ there exist $\lambda(P, N)$ isomorphic, disjoint lattices $(\mathbf{M}_\mathcal{T}(\mathcal{F}_c(k)), \succeq_\phi)$, each of size 2^{2k}.

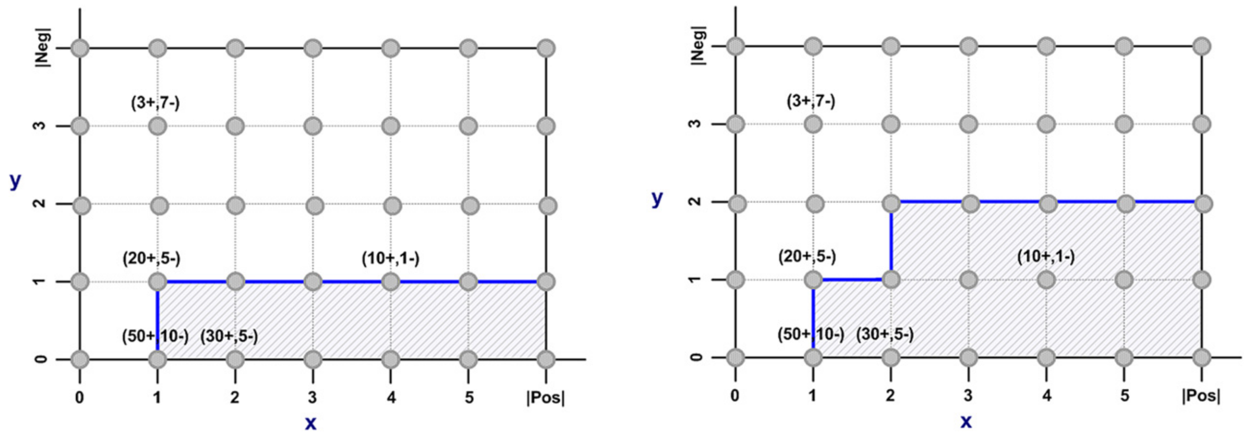


Fig. 3. Decision boundaries of $\{(1, 2)\}$ (left side) and $\{(1, 2), (2, 3)\}$ (right side).

Fact 4.3. *The partial order* $(\mathbf{M}(\mathcal{F}_c(k), P, N), \succcurlyeq_\phi)$ *consists of* $\lambda(P, N)$ *isomorphic, disjoint lattices* $(\mathbf{M}_\tau(\mathcal{F}_c(k)), \succcurlyeq_\tau)$*, each of size* 2^{2k}.

4.5. Decision boundaries

There is an interesting graphical representation of a classifier $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$ on the 2-dimensional space N^2 (see Fig. 3). Here, each point (x, y), with x and y non-negative integers, is labeled by a pair of integers $\langle\pi(x, y), \nu(x, y)\rangle$, where $\pi(x, y)$ is the number of *positive* examples (documents) and $\nu(x, y)$ the number of *negative* ones containing exactly x features from *Pos* and y features from *Neg*. Intuitively, we may think of a point (x, y) as identifying the set of (both positive and negative) examples with x positive features and y negative ones. Hence, the region of the plane

$$\mathrm{R}_{\mathcal{H}_c} = \big\{(x, y) \,\big|\, x \leqslant |Pos|,\ y \leqslant |Neg|,\ \exists(p_i, n_i) \in \mathcal{T} \text{ s.t. } x \geqslant p_i,\ y < n_i\big\},$$

whose points satisfy the threshold conditions, identifies the documents that are classified by $\mathcal{H}_c$ (we call $\mathrm{R}_{\mathcal{H}_c}$ *classification region*). It turns out that the number of documents classified by $\mathcal{H}_c$ is $\sum_{(x,y)\in \mathrm{R}_{\mathcal{H}_c}}(\pi(x, y) + \nu(x, y))$. The border of the region $\mathrm{R}_{\mathcal{H}_c}$ is the *decision boundary* of $\mathcal{H}_c$.

As an example, the classification regions of the (ϕ-homogeneous) classifiers $\mathcal{H}_c = \{(1, 2)\}$ and $\mathcal{H}'_c = \{(1, 2), (2, 3)\}$ are those depicted in Fig. 3. Here, the following should be noted:

1. the decision boundary of the atom $\mathcal{H}_c$ is a rectangle (left side of Fig. 3), while that of the 2-order classifier $\mathcal{H}'_c$ is the overlapping of two rectangles, one for each atom (right side of Fig. 3), and
2. the classification region of $\mathcal{H}_c$, which is a τ-specialization of $\mathcal{H}'_c$, is contained in the classification region of $\mathcal{H}_c$.

The above two statements can be generalized. In particular, concerning point (2), it can be easily verified that, for any two classifiers $\mathcal{H}'_c$, $\mathcal{H}_c$ such that $\mathcal{H}'_c \succcurlyeq_\tau \mathcal{H}_c$, the condition $\mathrm{R}_{\mathcal{H}'_c} \supseteq \mathrm{R}_{\mathcal{H}_c}$ holds, and vice versa (it suffices to use the above definition of classification region along with Definition 4.5). As for point (1), we can state that the decision boundary of a classifier $\mathcal{H}^1_c \vee \cdots \vee \mathcal{H}^r_c$ is a step-wise non-decreasing polyline in N^2 consisting alternately of vertical and horizontal segments. To see why, it suffices to observe the following:

a. the decision boundary of each single atom $\mathcal{H}^i_c = \{(p_i, n_i)\}$, $1 \leqslant i \leqslant r$, is a rectangle subtending the points (x, y) which satisfy the test conditions $p_i \leqslant x \leqslant |Pos|$ and $0 \leqslant y < n_i$, and
b. the r atoms $\mathcal{H}^1_c \cdots \mathcal{H}^r_c$ are such that $p_{i-1} < p_i$ and $n_{i-1} < n_i$, for each $i \in [1, r]$ (see Proposition 4.5, Part 2).

Intuitively, the non-decreasingness of decision boundaries implies that documents which are less likely to belong to a category c (that is, documents with few positive features and many negative ones) are also less likely to be classified by $\mathcal{H}_c$. For an instance, consider two documents $d(x, y)$ and $d'(x', y)$, having x and x' positive features, respectively, with $x' \geqslant x$, and both containing the same number y of negative features. Intuitively, $d(x, y)$ is less likely to be a positive example for c than $d'(x', y)$ (as it holds less positive features, which are indicative of membership, for the same number of negative ones). On the other hand, since the boundary is non-decreasing, it also happens that $d(x, y)$ is less likely to fall within $\mathrm{R}_{\mathcal{H}_c}$ than $d'(x', y)$, that is, $d(x, y)$ is less likely to be classified by $\mathcal{H}_c$.

4.6. Remarks on the proposed language

The "family resemblance" metaphor. In a binary classification task there are two families (classes): the positive, call it P, and the negative, call it N. Let us assume that the atom p-of-*Pos* $\wedge \neg n$-of-*Neg* is used to characterize the members of P.

Here, *Pos* is the set of features that such members share, while the threshold p states how many of such features each member must hold. Symmetrically, *Neg* is the set of features shared by the members of the other family N. Actually, not all members, but more specifically only those that are most similar to the members of P (recall that, by Definition 4.1, the features in *Neg* are those that characterize members of N holding some features of the family P). Thus, an example that exhibits p positive features is a member of P provided that it holds less than n negative features. To use an analogy, imagine that the members of the Brown family hold at least two of the following features: green eyes, black hair and tallness. However, also in the White family there are members that are tall and have green eyes, but they also hold at least two of the following features: fair hair, long nose and high forehead. Thus, an individual that is tall and has green eyes belongs to the Brown family provided that he possesses less than two of such features (that would play the role of negative features for the Brown family with threshold $n = 2$).

On the expressivity of M*-of-*$N^{\{\neg,\vee\}}$. M-of-N hypotheses can be regarded as M-of-$N^{\{\neg,\vee\}}$ atoms with *only* positive features, i.e., atoms of the form p-of-*Pos*. Simple M-of-N hypotheses are often sufficient for the classification of new and unseen data, but it is well known that there are cases where the need for negative features cannot be avoided. A simple example is the following: document $\{t_0, t_1\}$ belongs to c, document $\{t_0, t_1, t_2\}$ belongs to c' and document $\{t_1, t_2\}$ belongs to c''. It is easy to recognize that this scenario can be modeled by the atoms $\mathcal{H}_c = \langle\{t_0\}, \{t_2\}, \{(1,1)\}\rangle$, $\mathcal{H}_{c'} = \langle\{t_0, t_2\}, \emptyset, \{(2,1)\}\rangle$, and $\mathcal{H}_{c''} = \langle\{t_1\}, \{t_0\}, \{(1,1)\}\rangle$, where the negative features are needed to discriminate among classes.

Although M-of-$N^{\{\neg,\vee\}}$ atoms surpass classical M-of-N hypotheses in expressive power, there are data sets that cannot be represented simply by atoms. As an example, assume that documents $d_1 = \{t_0\}$, $d_2 = \{t_0, t_1\}$ and $d_3 = \{t_0, t_1, t_2\}$ are associated with category c, while $d_4 = \{t_0, t_2\}$ is not. Intuitively, to correctly classify such data we need a hypothesis $\mathcal{H}_c$ stating the following: the occurrence of either t_0 or t_1 is sufficient in order for a document d be classified under c, provided that t_2 does not appear in d; but, if t_2 does appear in d, a stronger condition is needed, that is, both t_0 and t_1 must occur in d. We can easily recognize that $\mathcal{H}_c$ is the 2-order classifier $\langle\{t_0, t_1\}, \{t_2\}, \{(1,1), (2,2)\}\rangle$, and that no atomic equivalent classifier there exists.

However, though the proposed language improves the expressive power of M-of-N concepts, M-of-$N^{\{\neg,\vee\}}$ does not actually reach the full expressiveness of DNF. For an instance, there is no M-of-$N^{\{\neg,\vee\}}$ hypothesis capable of explaining the following data: $d_1 = \{t_0\}$, $d_2 = \{t_1, t_2\}$ and $d_3 = \{t_0, t_2\}$, with d_1 and d_2 belonging to class c and d_3 to its complement. It is easy to recognize that the reason for this limitation is that all atoms forming a hypothesis share the same sets of positive and negative features. As we will see in the next sections, the rationale for this choice is that it drastically restricts the search space. That is, effectiveness is traded-of against efficiency.

Why subsumption relations are important. As we have seen, the two relations $\succcurlyeq_\tau$ and $\succcurlyeq_\phi$ codify the intuitive notion of "more-general-than" between hypotheses. That is, given $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ such that $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$, any example covered by H_c^2 is covered by $\mathcal{H}_c^1$ as well.

The idea of ordering the concept space by a "more-general-than" relation is not new in Inductive Logic Programming (see, e.g., [40]). What is actually original in our approach is the ordering along two dimensions, the feature and the threshold dimensions.

The ordering relations are important because they provide the learning algorithm with a means to *selectively* search the hypothesis space. For an instance, the search strategy can move towards a more general hypothesis whenever too few positive examples are covered by the current one or, vice versa, towards a more specific hypothesis if too negative examples are covered.

The implementation of a selective search requires the definition of suitable operators which, by exploiting the subsumption relations, enable the generalization/specialization of a hypothesis. This is what we will do in the next section.

5. Refinement operators

Informally, a refinement operator is a function which enables to "navigate" the space of the minimal classifiers through the partial order relations. We next provide two classes of refinement operators: *unary* and *binary* refinement operators.

Notation. For the sake of simplicity, in the following definitions we will often denote the set of minimal hypotheses $\mathbf{M}(\mathcal{F}_c(k), P, N,)$ simply by $\mathbf{M}$.

5.1. Unary refinement operators

A *unary refinement operator* is a non-deterministic function which returns a "neighbor" of $\mathcal{H}_c$ either in the ϕ-subsumption or in the τ-subsumption relationship. It is used to move a classifier "one step" upward or downward in either one of the two hierarchies.

Definition 5.1 *(Unary refinement operators).* A *unary refinement operator* is a non-deterministic function from $\mathbf{M}$ to $\mathbf{M}$. In particular, the *unary x-generalization* operator, denoted by $\uparrow^x$, with $x \in \{\phi, \tau\}$, is a function such that: $\uparrow^x (\mathcal{H}_c) = \mathcal{H}_c$ if $\nexists \mathcal{H}'_c \in \mathbf{M}$ such that $\mathcal{H}'_c >_x \mathcal{H}_c$ (i.e., $\mathcal{H}_c$ is the top element); otherwise, $\uparrow^x (\mathcal{H}_c) = \mathcal{H}'_c$ where $\mathcal{H}'_c >_x \mathcal{H}_c$ and $\nexists \mathcal{H}''_c$ such that $\mathcal{H}'_c >_x \mathcal{H}''_c >_x \mathcal{H}_c$. The *unary x-specialization* operator $\downarrow^x$ is defined accordingly.

We first provide a constructive definition of both $\uparrow^{\phi}(\mathcal{H}_c)$ and $\downarrow^{\phi}(\mathcal{H}_c)$ in the ϕ-subsumption lattice. Informally, a direct ancestor of $\mathcal{H}_c$ in the ϕ-subsumption hierarchy is obtained from $\mathcal{H}_c$ either by adding to *Pos* a candidate positive term or by removing any term from *Neg* (a direct descendant is obtained in a dual way).

◇ `COMPUTATION of` $\uparrow^{\phi}(\mathcal{H}_c)$ `and` $\downarrow^{\phi}(\mathcal{H}_c)$. Given $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$, compute:
$\uparrow^{\phi}(\mathcal{H}_c) = \mathcal{H}_c$ if $Pos = Pos^*(k)$ and $Neg = \emptyset$ (i.e., if $\mathcal{H}_c$ is the top element in the ϕ-subsumption lattice), otherwise

$$\uparrow^{\phi}(\mathcal{H}_c) = \begin{cases} \langle Pos \cup \{t\}, Neg, \mathcal{T}\rangle & \text{where } t \in Pos^*(k), \text{ or} \\ \langle Pos, Neg \setminus \{t\}, \mathcal{T}\rangle & \text{where } t \in Neg. \end{cases}$$

$\downarrow^{\phi}(\mathcal{H}_c) = \mathcal{H}_c$ if $Pos = \emptyset$ and $Neg = Neg^*(k)$, otherwise

$$\downarrow^{\phi}(\mathcal{H}_c) = \begin{cases} \langle Pos \setminus \{t\}, Neg, \mathcal{T}\rangle & \text{where } t \in Pos, \text{ or} \\ \langle Pos, Neg \cup \{t\}, \mathcal{T}\rangle & \text{where } t \in Neg^*(k). \end{cases} \qquad \square$$

A proof of correctness of the above computation is reported in Appendix A.

Example 5.1. Given $\mathcal{H}_c = \langle \{t_0, t_1\}, \{t_2\}, \mathcal{T}\rangle$, let $t \in Pos^*(k)$ and $t' \in Neg^*(k)$ be two candidate features. Then, the following hypotheses are "neighbors" of $\mathcal{H}_c$ in the ϕ-subsumption hierarchy:

$$\uparrow^{\phi}(\mathcal{H}_c) = \big\langle \{t_0, t_1, t\}, \{t_2\}, \mathcal{T}\big\rangle, \qquad \uparrow^{\phi}(\mathcal{H}_c) = \big\langle \{t_0, t_1\}, \emptyset, \mathcal{T}\big\rangle,$$
$$\downarrow^{\phi}(\mathcal{H}_c) = \big\langle \{t_1\}, \{t_2\}, \mathcal{T}\big\rangle, \qquad \downarrow^{\phi}(\mathcal{H}_c) = \big\langle \{t_0, t_1\}, \big\{t_2, t'\big\}, \mathcal{T}\big\rangle.$$

Let us now see how a neighbor $\uparrow^{\tau}(\mathcal{H}_c)$ or $\downarrow^{\tau}(\mathcal{H}_c)$ of $\mathcal{H}_c$ in the τ-subsumption lattice is computed. Clearly, to obtain, say, $\uparrow^{\tau}(\mathcal{H}_c)$, we have to replace in $\mathcal{H}_c$ the threshold set $\mathcal{T}$ by an immediate ancestor $\uparrow \mathcal{T}$ in the τ-subsumption lattice. So as the problem reduces to the computation of $\uparrow \mathcal{T}$.

◇ `COMPUTATION of` $\uparrow^{\tau}(\mathcal{H}_c)$ `and` $\downarrow^{\tau}(\mathcal{H}_c)$. Given $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$, compute $\uparrow^{\tau}(\mathcal{H}_c)$ and $\downarrow^{\tau}(\mathcal{H}_c)$ as follows

$$\uparrow^{\tau}(\mathcal{H}_c) = \langle Pos, Neg, \uparrow \mathcal{T}\rangle \quad \text{and} \quad \downarrow^{\tau}(\mathcal{H}_c) = \langle Pos, Neg, \downarrow \mathcal{T}\rangle$$

where the non-deterministic operator $\uparrow$ (resp. $\downarrow$) applied to $\mathcal{T}$ returns an immediate ancestor (resp. descendant) of $\mathcal{T}$ in the τ-subsumption hierarchy. $\uparrow \mathcal{T}$ is constructed from $\mathcal{T}$ by the algorithm of Fig. 4 (we do not report the dual algorithm for $\downarrow \mathcal{T}$ for space reason). □

For a description of the algorithm of Fig. 4 the reader is referred to Appendix B.

5.2. Binary refinement operators

Binary refinement operators are aimed at exploiting the lattice structure of both the ϕ-subsumption and the τ-subsumption hierarchies. In particular, given two classifiers, they return a classifier which is either the *lub* or the *glb* of the two classifiers in any of the two subsumption lattices, depending on whether a generalization or a specialization is needed, respectively.

Definition 5.2 *(Binary refinement operators).* A binary refinement operator is a function from $\mathbf{M} \times \mathbf{M}$ to $\mathbf{M}$. Let classifiers $\mathcal{H}_c^1 = \langle Pos_1, Neg_1, \mathcal{T}_1\rangle$ and $\mathcal{H}_c^2 = \langle Pos_2, Neg_2, \mathcal{T}_2\rangle$ be given. There are two *binary generalization operators*, the τ-generalization $\bigvee_{\tau}$ and the ϕ-generalization $\bigvee_{\phi}$, defined as follows:

$$\bigvee\nolimits_{\tau}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \big\langle Pos_1, Neg_1, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\big\rangle,$$
$$\bigvee\nolimits_{\phi}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \langle Pos_1 \cup Pos_2, Neg_1 \cap Neg_2, \mathcal{T}_1\rangle$$

and two *binary specialization operators* $\bigwedge_{\tau}$ and $\bigwedge_{\phi}$ defined as follows:

$$\bigwedge\nolimits_{\tau}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \big\langle Pos_1, Neg_1, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\big\rangle,$$
$$\bigwedge\nolimits_{\phi}\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \langle Pos_1 \cap Pos_2, Neg_1 \cup Neg_2, \mathcal{T}_1\rangle.$$

It should be noted that all the above operators are not commutative. In fact, $\bigvee_x(\mathcal{H}_c^1, \mathcal{H}_c^2)$, with $x \in \{\tau, \phi\}$, yields a generalization of $\mathcal{H}_c^1$ (through $\mathcal{H}_c^2$), and $\bigwedge_x(\mathcal{H}_c^1, \mathcal{H}_c^2)$ yields a specialization of $\mathcal{H}_c^1$ (through $\mathcal{H}_c^2$).

Non-deterministic function $\uparrow\mathcal{T}$

```
Input: threshold bounds $P$ and $N$; a minimal threshold set $\mathcal{T}=\{\tau_1,\ldots,\tau_k\}$, where
  $\tau_i=(p_i,n_i)$ for each $i\in[1,k]$ and $p_i<p_{i+1}$, $n_i<n_{i+1}$, for each $i\in[1,k)$ (see Proposition 4.5)
Output: a direct ancestor $\uparrow\mathcal{T}$ of $\mathcal{T}$;

function NewElement$(X,Y)$
1. if $(p_x>p_y)$ then swap $X=(p_x,n_x)$ and $Y=(p_y,n_y)$
2. $\delta^+=|p_y-p_x|$, $\delta^-=|n_y-n_x|$;
3. if $(\delta^+>1$ and $\delta^->1)$ then compute the most specific threshold pair $(p,n)$ such that
4.       $p_x<p<p_y$ and $n_x<n<n_y$, i.e., $(p,n)=(p_y-1,n_x+1)$
5. else compute the most specific threshold pair $(p,n)$ such that $p_x\leqslant p\leqslant p_y$
6.      and $n_x\leqslant n\leqslant n_y$ and:
7.      if $\delta^+>1$ then $(p,n)\succcurlyeq_\tau Y$; set $(p,n)=(p_y-1,n_y)$;
8.      else if $\delta^->1$ then $(p,n)\succcurlyeq_\tau X$; set $(p,n)=(p_x,n_x+1)$
9.         else $(p,n)\succcurlyeq_\tau X$ and $(p,n)\succcurlyeq_\tau Y$; set $(p,n)=(p_x,n_y)$
10. return $\{(p,n)\}$.

11. begin
12.    if $\mathcal{T}=\{(0,N)\}$ (i.e., $\mathcal{T}$ is the top of the lattice) return $\emptyset$;
13.    $\tau_0=(p_0,n_0)=(-1,0)$; $\tau_{k+1}=(p_{k+1},n_{k+1})=(P+1,N+1)$;
14.    randomly select $i\in[1,k]$;
15.    if $i=1$ and $p_i=0$ then $adj=\tau_{i+1}$ // right adjacent
16.    else if $i=k$ and $n_i=N$ then $adj=\tau_{i-1}$; // left adjacent
17.       else randomly select $adj\in\{\tau_{i-1},\ \tau_{i+1}\}$; ;
18. return $\uparrow\mathcal{T}=Minimize(\mathcal{T}\cup NewElement(\tau_i,adj))$;
```

Fig. 4. Pseudo code for the random selection of a direct ancestor of a threshold set in the τ-subsumption lattice.

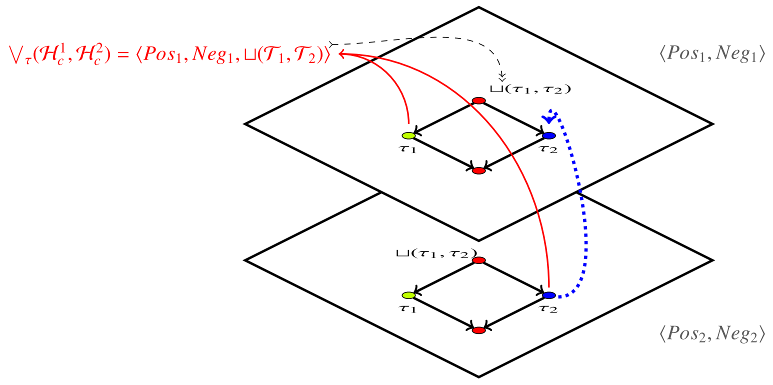

$\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)=\langle Pos_1,Neg_1,\sqcup(\mathcal{T}_1,\mathcal{T}_2)\rangle$
$\langle Pos_1,Neg_1\rangle$
$\sqcup(\tau_1,\tau_2)$
τ_1
τ_2
$\sqcup(\tau_1,\tau_2)$
τ_1
τ_2
$\langle Pos_2,Neg_2\rangle$

Fig. 5. Given $\mathcal{H}_c^1=\langle Pos_1,Neg_1,\mathcal{T}_1\rangle$ and $\mathcal{H}_c^2=\langle Pos_2,Neg_2,\mathcal{T}_2\rangle$, the hypothesis $\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)$ is the least upper bound of $\mathcal{H}_c^1$ and the ϕ-homogeneous classifier $\langle Pos_1,Neg_1,\mathcal{T}_2\rangle$, i.e., $\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)=\langle Pos_1,Neg_1,\sqcup(\mathcal{T}_1,\mathcal{T}_2)\rangle$.

Intuitively, $\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)$ is the least upper bound of $\mathcal{H}_c^1$ and the ϕ-homogeneous classifier having the same threshold set of $\mathcal{H}_c^2$ (see Fig. 5). Dually, $\bigvee_\phi(\mathcal{H}_c^1,\mathcal{H}_c^2)$ is the least upper bound of $\mathcal{H}_c^1$ and the τ-homogeneous classifier having the same feature sets of $\mathcal{H}_c^2$ (the specialization operators are defined accordingly).

Example 5.2. Consider $\mathcal{H}_c^1=\langle Pos_1,Neg_1,\mathcal{T}_1\rangle$ and $\mathcal{H}_c^2=\langle Pos_2,Neg_2,\mathcal{T}_2\rangle$, where $\mathcal{T}_1=\{(2,2)\}$ and $\mathcal{T}_2=\{(1,1)\}$. According to Definition 5.2, we have that

$$\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)=\langle Pos_1,Neg_1,\sqcup(\mathcal{T}_1,\mathcal{T}_2)\rangle=\langle Pos_1,Neg_1,\{(1,1),(2,2)\}\rangle,$$
$$\bigwedge_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)=\langle Pos_1,Neg_1,\sqcap(\mathcal{T}_1,\mathcal{T}_2)\rangle=\langle Pos_1,Neg_1,\{(2,1)\}\rangle.$$

It is easily verified that $\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)$ is a generalization of $\mathcal{H}_c^1$, while $\bigwedge_\tau$ is a specialization of $\mathcal{H}_c^1$.

Now, assume that $Pos_1=\{t_1,t_2\}$, $Neg_1=\{t_3,t_4\}$, $Pos_2=\{t_2,t_5\}$, $Neg_2=\{t_3\}$. We generalize $\mathcal{H}_c^1$ (through $\mathcal{H}_c^2$) by using the $\bigvee_\phi$ operator as follows:

$$\bigvee_\phi(\mathcal{H}_c^1,\mathcal{H}_c^2)=\langle Pos_1\cup Pos_2,Neg_1\cap Neg_2,\mathcal{T}_1\rangle=\langle\{t_1,t_2,t_5\},\{t_3\},\mathcal{T}_1\rangle$$

and specialize $\mathcal{H}_c^1$ (through $\mathcal{H}_c^2$) by $\bigwedge_\phi$ as follows:

$$\bigwedge_\phi\big(\mathcal{H}_c^1, \mathcal{H}_c^2\big) = \langle Pos_1 \cap Pos_2, Neg_1 \cup Neg_2, \mathcal{T}_1\rangle = \big\langle \{t_2\}, \{t_3, t_4\}, \mathcal{T}_1\big\rangle.$$

It is easy to recognize that both $\bigvee_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_\phi \mathcal{H}_c^1$ and $\mathcal{H}_c^1 \succcurlyeq_\phi \bigwedge_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2)$ hold.

Proposition 5.1. *Given classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$, the following hold*:

- $\bigvee_x(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_x \mathcal{H}_c^1$,
- $\mathcal{H}_c^1 \succcurlyeq_x \bigwedge_x(\mathcal{H}_c^1, \mathcal{H}_c^2)$,

where $x \in \{\tau, \phi\}$.

6. Learning problem and complexity

Before providing an effective algorithm for the learning of classifiers, in this section we give a definition of the learning problem and show its complexity.

The goal is to find, for each category $c \in \mathcal{C}$, a (minimal) hypothesis $\mathcal{H}_c \in \mathbf{M}(\mathcal{F}_c(k), P, N)$ that best fits the training data. To this end, we assume that categories in $\mathcal{C}$ are mutually independent, so as the whole learning task consists of $|\mathcal{C}|$ independent binary sub-tasks, one for each category.

To assess $\mathcal{H}_c$ we use the F-measure. This is a measure that trades off precision *Pr* versus recall *Re* and is defined as the harmonic mean of *Pr* and *Re* as follows[2]:

$$F = \frac{2\,Pr\,Re}{Pr + Re}. \tag{2}$$

Let us denote by $F(\mathcal{H}_c, T)$ the F-measure obtained by $\mathcal{H}_c$ when it is applied to the documents of the training set T. Now, the learning problem can be formulated as the following optimization problem.

Definition 6.1 *(Learning problem).* Let the feature space $\mathcal{F}_c(k)$ and the threshold bounds P, N be given. The learning problem is to find a (minimal) classifier $\mathcal{H}_c \in \mathbf{M}(\mathcal{F}_c(k), P, N)$ that maximizes the F-measure $F(\mathcal{H}_c, T)$ of $\mathcal{H}_c$ over the training set T.

The above learning problem is essentially an instance of Inductive Logic Programming (ILP) [41], which deals with the general problem of inducing logic programs from examples in the presence of background knowledge. It is well known that ILP problems are computationally intractable.

Proposition 6.1. *The decision version of the learning problem is NP-complete.*

The reader is referred to Appendix A for a proof of the above statement.

The theory of *PAC-learnability*, first proposed by Valiant in [42], provides a model of *approximated* polynomial learning where the polynomially bound amount of resources (both number of examples and computational time) is traded-off against the accuracy of the induced hypothesis. However, as shown by the above proposition, there is no algorithm that produces a consistent M-of-$N^{\{\neg,\vee\}}$ hypothesis on p examples in time polynomial in p, so as M-of-$N^{\{\neg,\vee\}}$ hypotheses are not PAC-learnable (this should not be surprising, given that M-of-N concepts are not PAC-learnable – see Pitt and Valiant [43]).

7. Learning a classifier: a GA-based approach

So far, we have seen the structural properties of the M-of-$N^{\{\neg,\vee\}}$ hypothesis space and designed a set of refinement operators that are the search abstract tools. Further, we have defined the learning problem and showed that it is computationally difficult. In this section we provide an effective algorithm for learning classifiers in the M-of-$N^{\{\neg,\vee\}}$ hypothesis space. In particular, we propose a heuristic approach based on a Genetic Algorithm (GA).

A GA represents a well known and powerful domain-independent search technique based on natural evolutionary operators. A standard GA can be regarded as composed of three basic elements: (1) A *population*, i.e., a set of candidate solutions (classifiers), called *individuals* or *chromosomes*, that will evolve during a number of iterations (*generations*); (2) a *fitness*

[2] This is also known as the $F1$-measure, because recall and precision are evenly weighted.

function used to assign a score to each individual of the population; (3) an evolution mechanism based on operators such as *elitism*, *selection*, *crossover* and *mutation*. A comprehensive description of GAs can be found in [44].

GAs showed to be well suited for learning classification rules (see, e.g., [29,28,27]) as well as M-of-N hypotheses [7], as they perform a thorough search of the hypothesis space, not limited by any greedy search bias. However, GAs also have some disadvantages for rule discovery. For instance, conventional genetic operators, such as crossover and mutation, are normally applied without directly trying to optimize the quality of the new candidate solution by exploiting the structure of the hypothesis space. A recent research trend aimed at overcoming this drawback is that of combining the standard search strategy of GAs with that of *task-specific* genetic operators which incorporate the knowledge about the specific application [30–33] (here, by "application" we mean the task of inducing classification rules).

Next we present GAMoN, the task-specific GA designed to induce M-of-$N^{\{\neg,\vee\}}$ hypotheses. As we will see, GAMoN relies on a search strategy where *ad hoc*, selective reproduction operators, aimed at exploiting the structure of the hypothesis space, are combined with standard ones.

Detecting the "best" hypothesis space $\mathbf{M}(\mathcal{F}_c(k), P, N)$ to be explored by GAMoN is a fundamental task which strongly affects the quality of the learning process. In principle, we might either (1) manage the model parameters k, P and N (which uniquely determines the hypothesis space) as parameters to be manually tuned, or (2) embed them in the evolutive dynamics of the GA, letting it to adaptively evolve the best values. GAMoN incorporates this latter approach. To this end, evolution relies on a number of competing subpopulations $S(k_1, P_1, N_1), \ldots, S(k_n, P_n, N_n)$, where each $S(k_i, P_i, N_i)$ consists of individuals encoding classifiers in the same hypothesis space $\mathbf{M}_i(\mathcal{F}_c(k_i), P_i, N_i)$, $1 \leqslant i \leqslant n$.

A preliminary step for the creation of the subpopulations $S(k_1, P_1, N_1), \ldots, S(k_n, P_n, N_n)$ is the detection of a suitable range $[k_{min}, k_{max}]$ for the feature space dimensionality k_i of each subpopulation. This is the subject of the next subsection. Afterward, we will discuss on individual encoding and reproduction operators. Then, we report a detailed description of the genetic algorithm GAMoN and, finally, we provide some remarks on the proposed GA.

7.1. Detecting the feature space dimensionality

The feature space $\mathcal{F}_c(k)$ provides the basic symbols from which the classifiers of a given hypothesis space $\mathbf{M}(\mathcal{F}_c(k), P, N)$ are constructed. Behind its definition there is the implicit assumption that only the selected terms are representative of the category being learned, while the rest are redundant. Thus, predicting the right value of the dimensionality k is a crucial step. On one hand, a reduced feature space is desirable as redundant or noisy features may "deceive" the learning algorithm and have detrimental effect on classification results (this is particularly true in the text classification task, where data sets are usually noisy and ambiguous). Further, reducing the number of features makes the learning process more efficient (especially in the evolutionary approach, where large feature spaces may entail large individuals and, thus, more match operations). On the other hand, an aggressive feature selection might discard features that carry essential information.

Next we provide a criterion, inspired to the one proposed in [36], for detecting a range of dimensionality values based on the statistical characteristics of the data set at hand.

Definition 7.1 *(Dimensionality range).* We are given a vocabulary V_c and a scoring function σ. We define the *dimensionality range* $[k_{min}, k_{max}]$ for category c as follows:

$$k_{min} = \left|\left\{t \in V_c \;\middle|\; \sigma(t, c) \geqslant m + s\right\}\right|,$$
$$k_{max} = \left|\left\{t \in V_c \;\middle|\; \sigma(t, c) \geqslant m + 3s\right\}\right|$$

where $\sigma(t, c)$ is the score of feature $t \in V_c$ w.r.t. category c, and m and s are the average and standard deviation of the scoring values, respectively.

We notice that the above definition is essentially aimed at selecting a good set $Pos_c^*(k)$ of candidate positive features (recall that $Neg_c^*(k)$ consists of terms *co-occurring* with terms in $Pos_c^*(k)$ – see Definition 4.1). Indeed, to determine k_{min} (resp. k_{max}) we compute the scoring function σ for all features, and then count the number of features whose score is higher than 1 (resp. 3) standard deviations above the average, i.e., features with high discriminating power.

7.2. Individual encoding

Given a hypothesis space $\mathbf{M}(\mathcal{F}_c(k), P, N)$, a candidate (minimal) classifier $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle \in \mathbf{M}(\mathcal{F}_c(k), P, N)$ is encoded by a bit string $I = \langle I^+, I^-, I^{\succcurlyeq\tau}\rangle$, where:

1. the positive component I^+ is used to encode $Pos \subseteq Pos_c^*(k)$. It is made of k bits, each associated with a candidate feature $t_i \in Pos_c^*(k)$. A '1' or '0' in the gene $I^+[t_i]$, $1 \leqslant i \leqslant k$, indicates whether or not $t_i \in Pos_c^*(k)$ belongs to Pos.
2. The negative component I^- is used to encode $Neg \subseteq Neg_c^*(k)$. It is made of k bits, each associated with a candidate feature $t_i \in Neg_c^*(k)$. A '1' or '0' in the gene $I^-[t_i]$, $1 \leqslant i \leqslant k$, indicates whether or not the i-th candidate feature t_i belongs to Neg.

3. The threshold component $I^{\succcurlyeq_\tau}$ is used to encode the threshold set $\mathcal{T}$. The encoding of $\mathcal{T}$ relies on a straightforward binary representation of all pairs $(p,n) \in \mathcal{T}$, with $0 \leqslant p \leqslant P$ and $0 < n \leqslant N$. One additional bit for each element (p,n) is used to represent presence/absence of that element. Thus, the length of $I^{\succcurlyeq_\tau}$ is $Min(P+1,N)(\lceil \log(N(P+1)) \rceil + 1)$, where $Min(P+1,N)$ is the maximum order of a classifier with threshold bounds P and N (see Proposition 4.10). In the following we will denote by *enc* the encoding function, i.e., $I^{\succcurlyeq_\tau} = enc(\mathcal{T})$.

It turns out that the length $L(I)$ of I is the following function of k, P and N

$$L(I) = L\big(I^+\big) + L\big(I^-\big) + L\big(I^{\succcurlyeq_\tau}\big) = 2k + Min(P+1,N)\big(\big\lceil \log\big(N(P+1)\big)\big\rceil + 1\big).$$

Clearly, individuals encoding classifiers in the same hypothesis space $\mathbf{M}(\mathcal{F}_c(k),P,N)$ are of equal length.

Example 7.1. Let the hypothesis space $\mathbf{M}(\mathcal{F}_c(k),P,N)$ be given, where $k=50$, $P=2$ and $N=3$. According to Proposition 4.10, the maximum order of a classifier is $Min(P+1,N) = Min(3,3) = 3$ (see also Example 4.10). Thus, an individual encoding a classifier $\langle Pos, Neg, \mathcal{T}\rangle \in \mathbf{M}(\mathcal{F}_c(k),P,N)$ consists of $2k = 100$ bits needed to represent sets *Pos* and *Neg*, and further $Min(P+1,N)(\lceil \log(N(P+1)) \rceil + 1) = 15$ bits to encode the threshold set $\mathcal{T}$.

7.3. Fitness

The performance measure used for evaluating the fitness of an individual is the objective function of the learning problem (see Definition 6.1).

Definition 7.2 *(Fitness).* We are given a chromosome I, encoding classifier $\mathcal{H}_c$, and the training set T. The fitness of I is $F(\mathcal{H}_c, T)$.

7.4. Task-specific GA operators and stochastic refinement

Next we propose some application-specific reproduction operators as an implementation of the refinement operators defined in Section 5. Such operators provide a concrete means whereby the learning algorithm selectively searches the hypothesis space. In particular, we next define two classes of Generalizing/Specializing (GS) operators: GS Crossover and GS Mutation.

7.4.1. Generalizing/specializing crossover

Crossover is the operation of swapping genetical material between two individuals (parents). GS crossover (GSX) is a special kind of crossover aimed at making a classifier more general or more specific.

The GSX operators we are defining are an application of the binary refinement operators given by Definition 5.2. As we have seen, they combine two classifiers of the same hypothesis space and provide a new classifier in the same space. Thus, GSX operators combine two parents belonging to the same subpopulation (i.e., encoding classifiers in the same hypothesis space) and yields an individual in the same subpopulation. Therefore, they operate on individuals of equal length (and isomorphic).

Notation. With a small abuse of notation, in the following we will denote by $\bigvee_x(I_1,I_2)$ and $\bigwedge_x(I_1,I_2)$ the individuals encoding the classifiers $\bigvee_x(\mathcal{H}_c^1,\mathcal{H}_c^2)$ and $\bigwedge_x(\mathcal{H}_c^1,\mathcal{H}_c^2)$, respectively, where $x \in \{\tau,\phi\}$ and I_i is the binary encoding of $\mathcal{H}_c^i$ $(1 \leqslant i \leqslant 2)$. Further, we write $I_1 \succcurlyeq_x I_2$ if $\mathcal{H}_c^1 \succcurlyeq_x \mathcal{H}_c^2$.

Definition 7.3 *(GSX operators).* We are given individuals I_1 and I_2 encoding classifiers $\mathcal{H}_c^1, \mathcal{H}_c^2 \in \mathbf{M}(\mathcal{F}_c(k),P,N)$, respectively. The *generalization* crossover $GX(I_1,I_2)$ of I_1 and I_2 is the individual encoding either the binary ϕ-generalization $\bigvee_\phi(\mathcal{H}_c^1,\mathcal{H}_c^2)$ or the binary τ-generalization $\bigvee_\tau(\mathcal{H}_c^1,\mathcal{H}_c^2)$ of $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$. More precisely, using the above agreed notation

$$GX(I_1,I_2) = \begin{cases} \bigvee_\tau(I_1,I_2) & \text{with probability } p = 0.5, \\ \bigvee_\phi(I_1,I_2) & \text{otherwise.} \end{cases}$$

The *specialization* crossover operator $SX(I_1,I_2)$ is defined accordingly (i.e., using $\bigwedge_x$ in place of $\bigvee_x$, with $x \in \{\tau,\phi\}$).

Based on Definition 5.2, the implementation of $\bigvee_\phi(I_1,I_2)$ and $\bigwedge_\phi(I_1,I_2)$ can be achieved by simple bitwise logical operations (OR and AND) on I_1 and I_2 as follows:

- $\bigvee_\phi(I_1,I_2) = I$ s.t. $I^+ = OR(I_1^+, I_2^+)$, $I^- = AND(I_1^-, I_2^-)$, $I^{\succcurlyeq_\tau} = I_1^{\succcurlyeq_\tau}$.
- $\bigwedge_\phi(I_1,I_2) = I$ s.t. $I^+ = AND(I_1^+, I_2^+)$, $I^- = OR(I_1^-, I_2^-)$, $I^{\succcurlyeq_\tau} = I_1^{\succcurlyeq_\tau}$.

Here $I^+ = OR(I_1^+, I_2^+)$ stands for $\forall i \in [1,k]$, $I^+[t_i] = OR(I_1^+[t_i], I_2^+[t_i])$ (AND is defined accordingly).

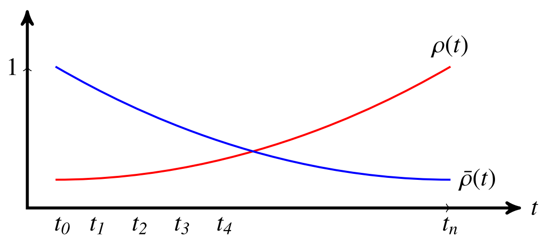


Fig. 6. Relevance $\rho(t)$ and irrelevance $\bar{\rho}(t)$ functions. Terms $t_0, \ldots, t_n$ are in increasing order of scoring value.

However, there is a problem with the above implementation. In fact, by performing a "blind" OR or AND, the two individuals exchange 0's and 1's with no regard for the relevance of the features they represent. This may be detrimental, as a surplus of low-quality features may increase the risk of overfitting the training data. To overcome this drawback, we introduce the *probabilistic OR* (pOR) and the *probabilistic AND* (pAND), which are logical operators biased towards high-quality features. They both rely on the notion of *relevance* of a candidate feature.

Definition 7.4. Given the feature space $\langle Pos_c^*(k), Neg_c^*(k)\rangle$, let σ and η be the scoring functions for the positive and the negative candidate terms, respectively (see Definition 4.1). With each $t \in Pos^*(k) \cup Neg^*(k)$ we assign the *relevance* measure $\rho(t)$ as follows:

$$\rho(t) = \frac{f(t)}{Max\{f(t_i) | t_i \in S\}},$$

where $f(t) = \sigma(t)$ and $S = Pos_c^*(k)$ if t is candidate positive feature,[3] or $f(t) = \eta(t)$ and $S = Neg^*(t)$ otherwise. Dually, we define the *irrelevance* $\bar{\rho}(t)$ of t as

$$\bar{\rho}(t) = \frac{Min\{f(t_i) | t_i \in S\}}{f(t)}$$

where f and S are as above. Clearly, $0 < \rho(t) \leqslant 1$ takes on the value 1 for the highest scoring term t, while $0 < \bar{\rho}(t) \leqslant 1$ takes on the value 1 for the lowest scoring term t (see Fig. 6).

Definition 7.5 *(Probabilistic logical operators).* Given two individuals I_1 and I_2, and a feature t, define the *probabilistic OR* as follows:

$$pOR\big(I_1[t], I_2[t]\big) = \begin{cases} OR(I_1[t], I_2[t]) & \text{with probability } p = \rho(t), \\ I_1[t] & \text{alternatively,} \end{cases}$$

where $\rho(t)$ is the relevance of t (see Definition 7.4). The *pAND* operator is defined accordingly, using $\bar{\rho}(t)$ in place of $\rho(t)$.

We note that neither operators are commutative. An important property of *pOR* is that $pOR(I_1[t], I_2[t]) \leqslant OR(I_1[t], I_2[t])$ holds, i.e., $pOR(I_1[t], I_2[t])$ may be 0 while $OR(I_1[t], I_2[t])$ is not, but not vice versa. In particular, $pOR(I_1[t], I_2[t]) = OR(I_1[t], I_2[t])$ when either $I_1[t] = 1$ or $I_1[t] = I_2[t] = 0$. Otherwise, i.e., $I_1[t] = 0$ and $I_2[t] = 1$, $pOR(I_1[t], I_2[t]) = OR(I_1[t], I_2[t]) = 1$ with probability $p = \rho(t)$. Clearly, the higher the relevance $\rho(t)$ (recall that $\rho(t) = 1$ when t is the highest scoring term), the higher the probability that a 1 is "moved" from I_2 to I_1 (at phenotypic level, this means that the classifier encoded by I_1 acquires a new feature t from the classifier encoded by I_2). Thus, the overall effect of *pOR* is that of "moving" preferably the most relevant features from I_2 to I_1. The *pAND* operator works in a dual way. That is, the effect of *pAND* is that of "discarding" from I_1 (by moving zeroes from I_2 to I_1) preferably the least relevant features.

Now, by using the above probabilistic logical operators in place of the standard ones, we compute an "approximation" of both $\bigvee_\phi$ and $\bigwedge_\phi$ as follows.

◇ COMPUTATION of $\approx \bigvee_\phi(I_1, I_2)$ and $\approx \bigwedge_\phi(I_1, I_2)$. Given the individuals I_1 and I_2, compute $\approx \bigvee_\phi(I_1, I_2)$ and $\approx \bigwedge_\phi(I_1, I_2)$ as follows:

- $\approx \bigvee_\phi(I_1, I_2) = I$ s.t. $I^+ = pOR(I_1^+, I_2^+)$, $I^- = pAND(I_1^-, I_2^-)$, $I^{\succcurlyeq\tau} = I_1^{\succcurlyeq\tau}$.
- $\approx \bigwedge_\phi(I_1, I_2) = I$ s.t. $I^+ = pAND(I_1^+, I_2^+)$, $I^- = pOR(I_1^-, I_2^-)$, $I^{\succcurlyeq\tau} = I_1^{\succcurlyeq\tau}$. □

It can be easily verified that $\approx \bigvee_\phi(I_1, I_2)$ is a generalization of I_1 and $\approx \bigwedge_\phi(I_1, I_2)$ a specialization of I_1.

[3] We assume that $\sigma(t) > 0$ for any $t \in Pos^*(t)$.

Unlike the implementation of the ϕ-subsumption primitives $\bigvee_\phi$ and $\bigwedge_\phi$, which relies on bit-wise operations performed at the genotype level, the implementation of the τ-subsumption primitives $\bigvee_\tau$ and $\bigwedge_\tau$ is performed at phenotype level. To see this point, we preliminarily recall that $\bigvee_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos_1, Neg_1, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\rangle$ (see Definition 5.2). Thus, to implement $\bigvee_\tau(I_1, I_2)$, we first extract from the individuals I_1 and I_2 the threshold sets $\mathcal{T}_1$ and $\mathcal{T}_2$ by using the inverse of the encoding function, i.e., $\mathcal{T}_i = enc^{-1}(I_i^{\succeq\tau})$ $(1 \leqslant i \leqslant 2)$. Then, we compute $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$ by using the algorithm of Fig. 2 and, finally, we apply the encoding function $enc(\sqcup(\mathcal{T}_1, \mathcal{T}_2))$. Therefore, $\bigvee_\tau(I_1, I_2)$ is the individual having the same positive and negative components of I_1 and threshold component $enc(\sqcup(\mathcal{T}_1, \mathcal{T}_2))$.

◇ `COMPUTATION of` $\bigvee_\tau(I_1, I_2)$ `and` $\bigwedge_\tau(I_1, I_2)$. Let I_1 and I_2 be two individuals, and let $\mathcal{T}_i = enc^{-1}(I_i^{\succeq\tau})$ be the threshold set of the classifier encoded by I_i $(1 \leqslant i \leqslant 2)$. Then compute
- $\bigvee_\tau(I_1, I_2) = I$ such that $I^+ = I_1^+$, $I^- = I_1^-$, $I^{\succeq\tau} = enc(\sqcup(\mathcal{T}_1, \mathcal{T}_2))$,
- $\bigwedge_\tau(I_1, I_2) = I$ such that $I^+ = I_1^+$, $I^- = I_1^-$, $I^{\succeq\tau} = enc(\sqcap(\mathcal{T}_1, \mathcal{T}_2))$

where $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$ and $\sqcap(\mathcal{T}_1, \mathcal{T}_2)$ are constructed by the algorithm of Fig. 2. □

The correctness of the above computation directly follows from Definition 5.2.

7.4.2. Generalizing/specializing (GS) mutation

The GS mutation (GSM) operators are an implementation of the unary refinement operators defined in Definition 5.1. Therefore, the GSM applied to an individual encoding $\mathcal{H}_c$ returns another individual encoding a neighbor (generalization or specialization) of $\mathcal{H}_c$ in either one of the two hierarchies.

Notation. In the following we will denote, with a small abuse of notation, by $\uparrow^x(I)$ (resp. $\downarrow^x(I)$) the individual encoding the classifiers in $\uparrow^x(\mathcal{H}_c)$ (resp. $\downarrow^x(\mathcal{H}_c)$), where $x \in \{\phi, \tau\}$ and I is the encoding of $\mathcal{H}_c$ (see Definition 5.1).

Definition 7.6 *(GSM operators).* Let I be an individual encoding $\mathcal{H}_c \in \mathbf{M}(\mathcal{F}_c(k), P, N)$. The *generalization mutation GM*(I) of I is an individual encoding a direct ancestor of $\mathcal{H}_c$ in $\mathbf{M}(\mathcal{F}_c(k), P, N)$ either in the τ- or in the ϕ-generalization hierarchy, i.e., either $GM(I) = \uparrow^\tau(I)$ or $GM(I) = \uparrow^\phi(I)$. More precisely,

$$GM(I) = \begin{cases} \uparrow^\phi(I) & \text{with probability } p = 0.5, \\ \uparrow^\tau(I) & \text{otherwise.} \end{cases}$$

The *specialization mutation SM*(I) is defined accordingly.

That is, the GS mutation of I yields an individual encoding, with equal probability, a neighbor of the classifier encoded by I either in the ϕ- or the τ-hierarchy (this is exactly what in our model is a "small" change in a hypothesis).

The computation of the non-deterministic primitives $\uparrow^\phi(I)$ and $\downarrow^\phi(I)$ is clearly the transposition at genotype level of the computation of the unary refinement operators $\uparrow^\phi(\mathcal{H}_c)$ and $\downarrow^\phi(\mathcal{H}_c)$ shown in Section 5.1. Hence, we obtain the binary encoding $\uparrow^\phi(I)$ of a classifier $\uparrow^\phi(\mathcal{H}_c)$ simply by flipping either one 0 into 1 in I^+ (i.e., add a positive feature to $\mathcal{H}_c$) or a 1 into 0 in I^- (i.e., remove a negative feature from $\mathcal{H}_c$). Dually, we get the binary encoding $\downarrow^\phi(I)$ of a classifier $\downarrow^\phi(\mathcal{H}_c)$ by flipping a 1 into 0 in I^+ or a 0 into 1 in I^-.

However, like in the case of the GS crossover, we bias the GM mutation towards high-relevance features. To this end, we introduce the notions of *insertion* probability $ip(t)$ and *removal* probability $rp(t)$ of a candidate feature t as follows

$$ip(t) = \frac{\rho(t)}{\sum_{i=1,k} \rho(t_i)}, \qquad rp(t) = \frac{\bar{\rho}(t)}{\sum_{i=1,k} \bar{\rho}(t_i)}$$

where $\rho(t)$ and $\bar{\rho}(t)$ are the relevance and the irrelevance measures of t, respectively (see Definition 7.4). Intuitively, the probability $ip(t)$ represents the chance that $I[t]$ is flipped from 0 to 1, that is, the chance that the candidate feature t is selected as a term (either positive or negative) for the classifier encoded by individual I. The meaning of $rp(t)$ is dual. We notice that, since $\rho(t_i) \leqslant 1$, the condition $\sum_{i=1,k} \rho(t_i) \leqslant k$ holds (recall that k is the number of both positive and negative features). Therefore, for the highest scoring feature t (for which $\rho(t) = 1$) we have that $ip(t) = 1/\sum_{i=1,k} \rho(t_i) \geqslant 1/k$, i.e., the maximum insertion probability is not smaller than $1/k$ (defined in [45] as the lower bound of the optimal mutation rate). Dually, the removal probability $rp(t)$ is maximum for the lowest scoring feature t, for which the relation $rp(t) \geqslant 1/k$ holds as well. We are now ready to provide the computation of $\uparrow^\phi(I)$ and $\downarrow^\phi(I)$.

◇ `COMPUTATION of` $\uparrow^\phi(I)$ `and` $\downarrow^\phi(I)$. Let the individual I be given. Compute $\uparrow^\phi(I)$ and $\downarrow^\phi(I)$ as follows:

(a) $\uparrow^\phi(I)$: select randomly (with probability 0.5) either one of the options below:
1. probabilistically select a bit $I^+[t] = 0$ according to the insertion probability distribution $ip(t)$; mutate it from 0 to 1, or
2. probabilistically select a bit $I^-[t] = 1$ according to the removal probability distribution $rp(t)$; mutate it from 1 to 0.

(b) $\downarrow^{\phi}(I)$: select randomly (with probability 0.5) either one of the options below:
1. probabilistically select a bit $I^{+}[t] = 1$ according to the removal probability distribution $rp(t)$; mutate it from 1 to 0, or
2. probabilistically select a bit $I^{-}[t] = 0$ according to the insertion probability distribution $ip(t)$; mutate it from 0 to 1. $\square$

Now let us consider the τ-subsumption primitives $\uparrow^{\tau}$ and $\downarrow^{\tau}$. Like in the case previously seen of the τ-subsumption primitives $\bigvee_{\tau}$ and $\bigwedge_{\tau}$, also the implementation of $\uparrow^{\tau}$ and $\downarrow^{\tau}$ is performed at phenotype level. To this end, we first extract from the individual I the threshold set $\mathcal{T}$ by using the inverse of the encoding function, i.e., $\mathcal{T} = enc^{-1}(I^{\succcurlyeq\tau})$. Then, we compute $\uparrow \mathcal{T}$ by using the algorithm of Fig. 4 and, finally, we apply the encoding function $enc(\uparrow \mathcal{T})$. Therefore, $\uparrow \mathcal{T}$ is the individual having the same positive and negative components of I and threshold component $enc(\uparrow \mathcal{T})$.

◇ `COMPUTATION of` $\uparrow^{\tau}(I)$ `and` $\downarrow^{\tau}(I)$. Let the individual I be given, and let $\mathcal{T} = enc^{-1}(I^{\succcurlyeq\tau})$ be the threshold set encoded by $I^{\succcurlyeq\tau}$. Now, $\uparrow^{\tau}(I)$ is implemented simply by replacing the encoding $I^{\succcurlyeq\tau}$ of $\mathcal{T}$ by the encoding $enc(\uparrow \mathcal{T})$ of any direct ancestor $\uparrow \mathcal{T}$ computed by the algorithm of Fig. 4. $\downarrow^{\tau}(I)$ is implemented accordingly. That is:
- $\uparrow^{\tau}(I) = I'$, where $I'^{+} = I^{+}$, $I'^{-} = I^{-}$, and $I'^{\succcurlyeq\tau} = enc(\uparrow \mathcal{T})$.
- $\downarrow^{\tau}(I) = I'$, where $I'^{+} = I^{+}$, $I'^{-} = I^{-}$, and $I'^{\succcurlyeq\tau} = enc(\downarrow \mathcal{T})$. $\square$

7.5. The genetic algorithm

First, the dimensionality range $[k_{min}, k_{max}]$ is computed from the input vocabulary by applying Definition 7.1. Then, given the (user-defined) input values P_{max} and N_{max}, for each randomly generated triple (k, P, N), with $k \in [k_{min}, k_{max}]$, $0 \leqslant P \leqslant P_{max}$ and $0 < N \leqslant N_{max}$, a random number (> 1) of individuals of length $2k + Min(P+1, N)(\lceil log(N(P+1))\rceil + 1)$ is created. Each of such individuals encodes a classifier in the hypothesis space $\mathbf{M}(\mathcal{F}_c(k), P, N)$. The set of individuals created for the same triple (k, P, N) form a subpopulation $S(k, P, N)$. Each individual $I \in S(k, P, N)$ is initialized as follows: the k bits in I^{+} and I^{-} are set to 1 with probability 0.5, while $I^{\succcurlyeq\tau}$ is randomly set to a (minimal) threshold set $\{(p_1, n_1), \ldots, (p_r, n_r)\}$ such that $0 \leqslant p_i \leqslant P$ and $0 < n_i \leqslant N$, for each $i = 1, r$ (see Definition 4.2). Afterwards, evolution takes place by iterating elitism, selection, crossover and mutation, until a pre-defined number of generations is created. Finally, the phenotype of the best generated chromosome is returned.

Next we give some details about selection, crossover and mutation.

Selection. We want to be able to preserve subpopulations under the pressure of selection, in order to guarantee a certain degree of population diversity (niching methods are often used for this purpose [46,47]). At the same time, we want to avoid premature convergence within subpopulations that consist of a small number of individuals. At these aims, we maintain a set of mating pools, each being the union of all subpopulations with the same threshold bounds P and N, i.e., $M(P, N) = \bigcup_i S(k_i, P, N)$. So, individuals in $M(P, N)$ may have different length, while belonging to isomorphic τ-subsumption lattices. In particular, the lengths of two individuals in $M(P, N)$ may differ only as far as the feature components are concerned, the threshold components being of equal length (as they have the same threshold bounds P and N – see Section 7.2). Now, selection is performed as follows: a mating pool is randomly selected, and tournament selection is then applied over its individuals.

Crossover. We are given individuals $I_1 \in S(k_1, P, N)$ and $I_2 \in S(k_2, P, N)$, thus belonging to the same mating pool $M(P, N)$. The GAMoN crossover of I_1, I_2 combines a slightly modified version of the uniform crossover (called MUX) with the GS crossover operators defined in the previous sections. A sketch of the proposed method is shown in Fig. 7. It basically relies on two steps:

Step 1: Decide probabilistically whether or not $MUX(I_1, I_2)$ takes place (line 18). This decision is made positively with (user-defined) probability p_x. MUX is an adaptation of the uniform crossover UX to deal with (i) the different lengths of the feature components of two mating individuals, and (ii) the presence of threshold sets. Informally, $MUX(I_1, I_2)$ can be regarded as $UX(I_1, I_2)$ where (1) only the first $min(k_1, k_2)$ bits of the positive and negative components of I_1 and I_2 are probabilistically exchanged (lines 2–4), and (2) $I_1^{\succcurlyeq}$ and $I_2^{\succcurlyeq}$ are swapped as if they were single bits (line 5). Note that the offspring J_1 and J_2 are such that $J_1 \in S(k_1, P, N)$ and $J_2 \in S(k_2, P, N)$ (i.e., they belong to the same subpopulations of their parents).

Step 2: If the decision for MUX has not been made positively in Step 1, then perform the GS crossover by invoking function $GSX(I_1, I_2)$ (line 20). This is executed with a probability equal to the F-measure of the classifier $\mathcal{H}_c$ encoded by I_1 (line 13). This way, we give fitter individuals a higher chance to generalize or specialize so as to allow them for further refinement – see discussion in Section 7.7. Whether a generalization or a specialization of I_1 is to be performed, depends on whether $\mathcal{H}_c$ is too specific or too general (line 14). However, to carry out $GSX(I_1, I_2)$, individual I_2 has preliminarily to be "promoted" citizen of the subpopulation $S(k_1, P, N)$ of I_1 as, by Definition 7.3, the GS crossover can be applied only to members of the same subpopulation. To this end, I_2 is made of the same length of I_1 by invoking function *promote* (line 15). The following two cases may arise (recall that the threshold set components of I_1 and I_2 are of equal length):

- $k_1 \leqslant k_2$. Only the first k_1 bits of I_2^{+} and I_2^{-} are picked up (lines 7–8).

```
GAMoN Xover($I_1, I_2$)
Input: individuals $I_1 \in S(k_1, P, N)$ and $I_2 \in S(k_2, P, N)$; MUX probability $p_x$;
Output: offspring $J_1 \in S(k_1, P, N)$ and $J_2 \in S(k_2, P, N)$
function MUX($I_1, I_2$)
1. $J_1 = I_1,\ J_2 = I_2$;
2. for $i = 1$ to $min(k_1, k_2)$ do
3.       Swap($J_1^+[i], J_2^+[i]$) with probability 0.5;
4.       Swap($J_1^-[i], J_2^-[i]$) with probability 0.5;
5. Swap($J_1^{\succcurlyeq\tau}, J_2^{\succcurlyeq\tau}$) with probability 0.5;
return $J_1, J_2$.
function promote($I_2$, generalize)
6. $J^{\succcurlyeq\tau} = I_2^{\succcurlyeq\tau}$;    /* copy the threshold component of $I_2$ into $J$ */
7. for $i = 1, min(k_1, k_2)$ do    /* copy the first $min(k_1, k_2)$ positive and negative features from $I_2$ to $J$ */
8.     $J^+[i] = I_2^+[i],\ J^-[i] = I_2^-[i]$;
9. for $i = k_2 + 1, k_1$ do    /* when $k_1 > k_2$ add further $k_1 - k_2$ bits to both $J^+$ and $J^-$ so that their length becomes $k_1$ */
10.    if generalize then
11.        $J^+[i] = 0,\ J^-[i] = 1$    /* pad $J^+$, $J^-$ with $k_1 - k_2$ 0's and 1's, resp., */
12.    else $J^+[i] = 1,\ J^-[i] = 0$;    /* pad $J^+$, $J^-$ with $k_1 - k_2$ 1's and 0's, resp. */
return $J$.
function GSX($I_1, I_2$)
13. with probability equal to $Fmeasure(\mathcal{H}_c)$ do    /* $\mathcal{H}_c$ is the classifier encoded by $I_1$ */
14.    $generalize = (precision(\mathcal{H}_c) > recall(\mathcal{H}_c))$;
15.    $\widehat{I_2} = promote(I_2, generalize)$;    /* $I_2$ becomes a citizen of $S(k_1, P, N)$ */
16.    if generalize then $J = GX(I_1, \widehat{I_2})$
17.    else $J = SX(I_1, \widehat{I_2})$    /* GX or SX are performed according to Definition 7.3 */
return $j$.
begin
18.    with probability $p_x$ set $\langle J_1, J_2 \rangle = MUX(I_1, I_2)$;
19.    if MUX has not been performed then
20.        $J_1 = GSX(I_1, I_2)$; $J_2 = GSX(I_2, I_1)$
return $J_1, J_2$.
```

Fig. 7. Pseudocode for the GAMoN Xover.

- $k_1 > k_2$. Both I_2^+ and I_2^- are extended with further $n = k_1 - k_2$ bits. In particular, I_2^+ is padded with n 1's (resp. 0's) and I_2^- with n 0's (resp. 1's) if a generalization (resp. specialization) is to be performed (lines 9–12).

At this point, either $GX(I_1, I_2)$ or $SX(I_1, I_2)$ is computed according to Definition 7.3 (lines 16–17). Once $GSX(I_1, I_2)$ has been carried out, $GSX(I_2, I_1)$ is performed likewise (line 20). Again, the offspring J_1 and J_2 belong to subpopulations $S(k_1, P, N)$ and $S(k_2, P, N)$, respectively.

Mutation. Mutation is performed by using a similar framework. This is a combination of a modified version of standard mutation (denoted MSM), which takes into account threshold sets, with the GS mutation (GSM) operators previously defined. In particular, MSM(I) works as follows: first, it randomly decides (with probability 0.5) whether to operate over the feature or the threshold component. In the former case, MSM(I) randomly flips the bits of I^+ and I^- with probability $1/(2k)$. In the latter case, MSM(I) replaces $I^{\succcurlyeq}$ by the encoding of a randomly chosen neighbor of the threshold set encoded by $I^{\succcurlyeq}$ (contrary to GSM which selectively chooses either a direct ancestor or a direct descendant depending on whether generalization or specialization is to be performed, respectively). Note that, in all cases, MSM(I) causes small changes of position in the subsumption lattices. Now, GAMoN mutation works as follows (for each offspring):

- *Step* 1: decide probabilistically whether or not GSM(I) takes place. This decision is made positively with probability $F\text{-}measure(\mathcal{H}_c)$, $\mathcal{H}_c$ being the phenotype of I.
- *Step* 2: If the decision for GSM(I) has not been made positively in step 1, execute MSM(I).

7.6. GAMoN time complexity

It is immediate to recognize that the cost of the task-specific reproduction operators is $O(k)$, while the cost of the fitness computation is $O(km)$, where k is the size of the feature space and m the number of examples in the training set (in fact, the evaluation of the fitness of an individual requires the evaluation of the number of candidate (both positive and negative) features occurring in each document of the training set). Now, since the number of different features (words) occurring in the training set is asymptotically independent of m (as the lexicon is finite), irrespective of feature selection, k is (asymptotically) independent of m. Thus, technically, we have that $O(km) = O(m)$, that is, the asymptotic behavior of GAMoN is linear in the size of the training set. Quite obviously, for relatively small values of m (like those that characterize real-life data sets), the practical complexity is $O(km)$.

7.7. Remarks on the proposed GA

Individual encoding. There are two basic approaches, according to whether a chromosome of the population is used to represent *a single rule* or a *rule set* [9]. Within the former approach (i.e., "chromosome = one rule") there are rule induction GAs like XCS [27], SIA [28], COGIN [48]. In the second approach (i.e., the "chromosome = set of rules"), called Pittsburgh approach, a rule is used to code an entire classifier. GAssist [29], OlexGA [10] and BioHEL [12,13] fall in this category.

From one side, the "chromosome = one rule" approach makes the individual encoding simpler, but the fitness of a genotype may not be a meaningful indicator of the quality of the phenotype [49]. Further, under the competitive style of GA, there may be a conflict between individual and collective interests of the rules forming a classifier [9]. On the other side, the "chromosome = set of rules" approach requires a more sophisticated encoding of individuals, but the fitness provides a more reliable indicator [49]. Moreover, no conflict of interests can happen in this case, as competition occurs among classifiers (and not single rules).

In our approach, an individual encodes a candidate classifier – so it falls in the class of Pittsburgh methods. Despite this, individual encoding is very simple and compact – $2k$ bits for the encoding of *Pos* and *Neg* (a few tens of bits altogether) and a handful of bits to encode the threshold set. Thus, GAMoN combines the advantages of both the above mentioned approaches, i.e., the individual simplicity and compactness of the "chromosome = one rule" approach, along with the effectiveness of both the reproductive competition and the fitness function of the "chromosome = set of rules" approach.

Search strategy. It is well known that standard reproduction operators are rather disruptive, in the sense that the offspring may be very different from the parents. On one hand, this has the advantage of making unlikely the GA getting stuck into local optima but, on the other hand, the high degree of unpredictability in the generation of new candidate classifiers may make the GA converge very slowly. In contrast, GS operators move hypotheses from one position to another in either one of the two hierarchies in a controlled way, depending on the "state" of the current hypothesis. Such a search bias, however, forces a search strategy which may quickly converge to local optima.

As we have seen, GAMoN combines the space search of a standard GA with that based on GS operators. The rationale behind this choice is that of exploiting the latter to perform a selective search, and to compensate the selectiveness of this search by introducing a certain degree of diversity through the standard operators. In particular, GAMoN runs the GS operators (both crossover and mutation) with increasing probability, this being defined as the F-measure achieved by an individual I over the training set. This way, as the generations pass and the algorithm more and more approaches the optimal solution, a more controlled search of the space is performed.

8. Empirical investigation framework

8.1. Machine learning algorithms

To evaluate the GAMoN approach proposed in this paper, we focused on comparisons with other rule learning algorithms. To this end, we selected two rule induction GAs, namely, BioHEL and OlexGA, and two non-evolutionary algorithms, namely, C4.5 and Ripper. Further, we included in our study Platt's Sequential Minimal Optimization (SMO) method for linear SVM training [16], as it is reported to be one of the best methods for text categorization.

Our interest for OlexGA was that of assessing to what extent GAMoN is an effective extension (see Section 2). BioHEL was chosen as it is one of best performing GA-based methods. We are not aware of experimental results of BioHEL on textual data sets. Finally, C4.5 and Ripper were selected as they are standard decision tree/rule learners widely used for text classification.

All the selected learning algorithms are implemented in Java, but not all are available on the same platform. In particular, GAMoN runs only on the Weka platform, while BioHEL is available only on the KEEL platform (KEEL – Knowledge Extraction based on Evolutionary Learning – is a suite of machine learning software tools – [50]). C4.5, Ripper, SMO and OlexGA run on both platforms. For the purpose of our work, we used Weka (version 3.5.8) for all algorithms, but BioHEL.

8.2. Data sets

We carried out our empirical work on 13 real-world data sets whose properties are summarized in Table 1. As we can see, they span over a wide range of sizes, from a minimum of around 900 (Oh15) to a maximum of nearly 204,000 (market) documents. The rarest category has 51 documents (Oh0), while the most frequent has 85,440 documents (Market). Most of these datasets have been widely used in large scale text classification tasks, and are publicly available.

Market is a data set of 203,926 documents extracted from Reuters Corpus Volume I (RCV1) [63]. R10 is the standard subset of the Reuters-21578 Distribution 1.0 which consists of 12,897 documents and uses the 10 most frequent Topics categories [51]. Ohsumed is from the Ohsumed-233445 collection subset of MEDLINE database [52] and is made of all 34,389 cardiovascular diseases abstracts out of 50,216 medical abstracts contained in the year 1991. The classification scheme consists of the 23 cardiovascular diseases MeSH categories. Ohscale, Oh0, Oh5, Oh10 and Oh15 are other subsets of Ohsumed-233445 [53]. Data sets SRAA and 20NG (20-newsgroups) are articles from newsgroups. In particular, 20NG is

Table 1
Data set description.

Name	Source	Original format	#Doc	#Feat	#Cat	Cat size	
						Min	Max
Oh15	Ohsumed-233445	arff	913	3100	10	53	157
Oh5	Ohsumed-233445	arff	918	3012	10	59	149
Oh0	Ohsumed-233445	arff	1003	3182	10	51	194
Oh10	Ohsumed-233445	arff	1050	3238	10	52	165
BlogsGender	Blog author gender	text	3232	15,026	2	1548	1684
Ohscale	Ohsumed-233445	arff	11,162	11,465	10	709	1621
R10	Reuters-21578	text	12,897	21,363	10	237	3964
20NG	20 newsgroups	csv	18,846	59,903	20	628	999
Ohsumed	Ohsumed-233445	text	34,389	34,359	23	427	9611
Cade12	Gerindo Proj.	csv	40,983	69,470	12	625	8473
SRAA	UseNet	text	73,218	63,966	4	4796	41,351
ODP-S22	ODP	text	107,262	25,068	22	88	28,286
Market	Rcv1	text	203,926	68,604	4	26,036	85,440

a collection of 18,846 newsgroup documents organized into 20 different categories. We used the version sorted by date, which does not include newsgroup-identifying headers. The SRAA [54] data set contains 73,218 articles from four discussion groups on simulated auto racing, simulated aviation, real autos, and real aviation. BlogsGender is a binary data set of 3232 blogs used for author gender classification [55]. Cade12 is a subset of the CADE Web Directory consisting of 40,983 web pages classified across 12 categories [56]. ODP-S22 is a subset of ODP (Open Directory Project) [57] whose documents are stored as RDF files. For our experimentation, we used the subset of 107,262 documents classified under the categories of the *Top/Science* subtree, which has 25 first-level categories. We first collapsed each of the 25 subtrees into the respective root, thus obtaining a flat structure made of 25 categories. Then, we grouped together into one category "Misc" the 4 smallest categories, namely, Search Engines (7 documents), Charts-and-forums (16 documents), Directories (27 documents) and Events (38 documents), thus getting a set of 22 categories. From each document (web page), we extracted the title and the description (thus, discarding the URL).

8.3. Experimental setup

We preliminarily pre-processed all data sets downloaded in textual format, by performing tokenization (word unigrams) and stopword removal. We used the bag-of-words representation with binary word weighting. Each feature was represented as a numerical attribute.

Experiments were performed in a binary classification setting. To this end, we binarized all data sets by performing multi-class to two-class conversion. This way, the m-class learning problem is decomposed into m independent two-class sub-problems, one for each class, with the i-th classifier separating class i from all the remaining ones.

Finally, for each category, feature scoring by CHI square [39] was performed (on the training set).

Following are two major issues arose during the design of experiments.

1. Dimensionality of the feature space.
   (a) Unlike the other systems, GAMoN automatically detects the appropriate dimensionality of the feature space. That is, no manual feature selection is preliminarily needed. As we will see later on this section, the feature spaces selected by GAMoN usually consist of a few tens of features.
   (b) Previous works show that systems like Ripper, C4.5 and SVM require relatively large vocabularies (usually a few thousands of features) to learn good prediction functions.
   (c) The efficiency of OlexGA, like that of most evolutionary methods, strongly depends on the feature space dimensionality, as many features imply long individuals and, thus, low efficiency.
   (d) BioHEL represents an exception in the evolutionary landscape, as it was designed to efficiently deal with high dimensional data sets. However, the memory space limitations of the KEEL platform severely limits the number of attributes that can actually be used in case of large data sets.

   The above observations demonstrate the difficulty of applying a single feature selection policy to all systems. In fact, it would be unfair using for the non-evolutionary methods the feature dimensionalities detected by GAMoN (too small for their characteristics – see points (a) and (b)). On the other hand, running OlexGA and BioHEL over the same number of features used for the non-evolutionary methods would practically be unfeasible – see points (c) and (d).
2. Time efficiency. As we have seen, our empirical study involves very large data sets (e.g., ODP-S22 and Market), on which most of the experimented systems perform quite inefficiently. In particular, Ripper and C4.5 showed to be extremely slow on such data sets, especially when we tried to use large vocabularies (for an instance, on vocabularies of 10,000 features, we had to stop C4.5 as it was overly inefficient). This prevented us from performing optimization over more vocabularies.

Table 2
Feature space dimensionality range $[k_{min}, k_{max}]$, size of the feature space on which the "optimal" classifier has been found k_{opt} and F-measure for categories of R10.

	acq	corn	crud	earn	grain	int	mon	ship	trad	wheat
k_{min}	105	16	58	37	42	59	86	53	78	18
k_{max}	172	43	106	60	77	102	148	97	141	41
k_{opt}	132	21	100	41	55	68	99	61	86	21
PRavg	86.66	90.20	87.50	95.18	92.09	56.05	68.12	80.41	69.15	88.66

Given the above premises, the following experimental design choices were finally taken:

1. GAMoN was run with 500 individuals, 200 generations, elitism rate 0.2, MUX probability 0.6 (see Section 7.5). The maximum threshold bounds were $P_{max} = N_{max} = 4$.
2. OlexGA and BioHEL were executed over the same vocabularies made of 100 features. The former was run with the default parameters shown at http://www.mat.unical.it/OlexGA, while the latter with those provided by KEEL.
3. The remaining (non-evolutionary) systems were all executed over vocabularies made of 2000 terms, with the default settings provided by Weka. The SMO normalization option was turned off to improve the training time.

Due to efficiency reasons, we performed 5-fold cross validation (80% training, 20% test) only on small data sets (from Oh15 up to R10), while holdout (70% training, 30% test) was applied on the remaining data sets.

8.4. Predictive performance measure and statistical tests

Performance was measured, as is common in text classification, by the arithmetic mean of Precision and Recall – denoted by PRavg (an approximation of the Precision/Recall Break-Even Point). To obtain global estimates over more categories, the standard definitions of micro-averaged Precision and Recall were used, notably:

$$\mu Pr = \frac{\sum_{c \in \mathcal{C}} |TP_c|}{\sum_{c \in \mathcal{C}} (|TP_c| + |FP_c|)},$$
$$\mu Re = \frac{\sum_{c \in \mathcal{C}} |TP_c|}{\sum_{c \in \mathcal{C}} (|TP_c| + |FN_c|)},$$

where TP_c is the set of documents correctly assigned by the classifier to category c, FP_c is the set of documents incorrectly assigned by the classifier to category c, and FN_c is the set of documents incorrectly not assigned by the classifier to category c. We note that micro-averaging gives equal weight to every document (it is called a document-pivoted measure) and is largely dependent on the most common categories.

Each run of the evolutionary algorithms was repeated 3 times, and the average PRavg was taken.

In order to make comparisons statistically significant, we performed the Iman–Davenport test, with the Holm's *post-hoc* test, recommended for comparison of more classifiers on multiple data sets in [58].

9. Experimental results

9.1. A glimpse to M-of-$N^{\{\neg,\vee\}}$ hypotheses

The experimental results show that GAMoN has a bias towards learning compact and readable hypotheses. The following are examples of classifiers induced for categories "corn", "wheat" and "grain" from R10:

$$\mathcal{H}_{wheat} = \big\langle \{wheat\}, \{deficit, investment, net, treasury, york\}, \big\{(1,1)\big\}\big\rangle,$$
$$\mathcal{H}_{corn} = \big\langle \{corn, maize\}, \{london, money, quarter\}, \big\{(1,1)\big\}\big\rangle,$$
$$\mathcal{H}_{grain} = \big\langle \{barley, cereals, corn, grain, maize, rice, sorghum, wheat\},$$
$$\{acquisition, bank, earning, pay, profit, tax, york\}, \big\{(1,1),(2,2)\big\}\big\rangle.$$

As we can see, the former two classifiers are atoms, while the latter is a 2-order classifier (for a description see Section 3). It must be emphasized the high semantic correlation between the positive features and the respective categories.

9.2. Automatic selection of the feature space dimensionality

Table 2 shows, for the categories from R10, the values of k_{min}, k_{max} and k_{opt} given by one execution of GAMoN, where k_{min} and k_{max} define the dimensionality range of the feature space, and k_{opt} ($k_{min} < k_{opt} < k_{max}$) is the size of the feature

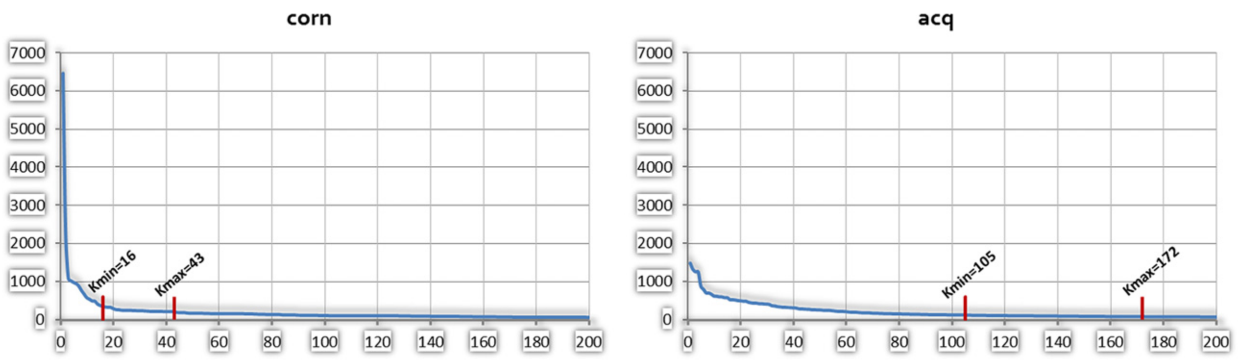


Fig. 8. Distribution of features by CHI square for two categories from R10 – “corn” (left side) and “acq” (right side). Only first 200 features are shown.

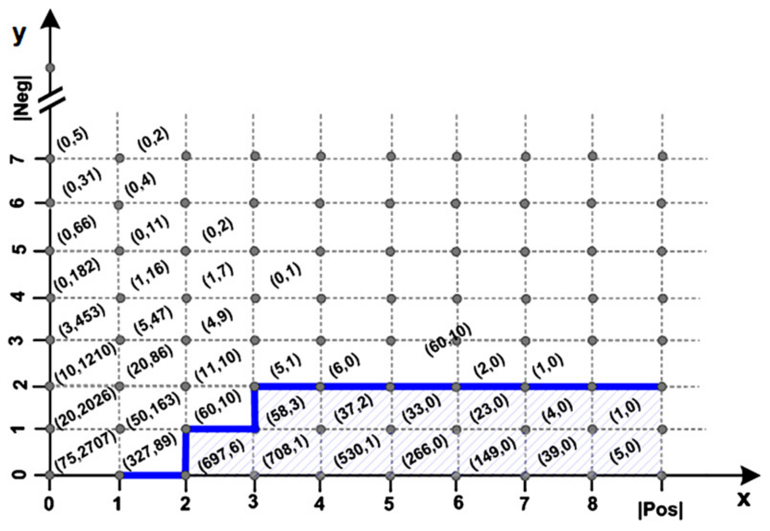


Fig. 9. Decision boundary of the classifier $\langle Pos, Neg, \{(1,1),(2,2),(3,3)\}\rangle$ for category “earn” from R10. Each label $(\pi(x,y), \nu(x,y))$ represents the number $\pi(x,y)$ of positive examples and the number $\nu(x,y)$ of negative ones with x positive features and y negative ones. Labels $(0,0)$ are omitted from the figure.

space on which the “optimal” classifier has been found. As it can be seen, the learning of all categories generally relies on small sets of candidate features. As an example, for “corn” we have $k_{min} = 19$ and $k_{max} = 43$ (and PRavg equal to 90.20), meaning that the positive terms for the “best” classifier are to be found among the first k higher scoring features, with $19 \leqslant k \leqslant 43$. This is clearly indicative of an aggressive feature selection. To see why, let us have a look at Fig. 8 – left side, where the distribution of features by CHI square is reported. As we can see, “corn” has a few features scoring very high, while the remaining ones rapidly approach near-zero values. The sharply declining shape of this graph is indicative of an “easy” category, i.e., a category for which a high performance can be achieved with only a few discriminative words.

In contrast, “acq” is a more “difficult” category. As we can see from Fig. 8 – right side, it has lower initial CHI square values, and the graph has a smooth (decreasing) trend. That is, no features with highly discriminative power there exist. As a consequence, the dimensionality range is shifted rightwards on the x-axis ($k_{min} = 105$ and $k_{max} = 172$), this being indicative of a less aggressive reduction of the feature space.

9.3. Decision boundaries

Fig. 9 shows the decision boundary $DB_{\mathcal{H}_{earn}}$ of the 3-order classifier $\mathcal{H}_{earn} = \langle Pos, Neg, \{(1,1),(2,2),(3,3)\}\rangle$ for category “earn” from R10. Here, *Pos* is made of 14 positive features and *Neg* consists of 16 negative features. As we can see, $DB_{\mathcal{H}_{earn}}$ is a three-step polyline. From the data reported in Fig. 9, it results that the subset of documents classified by $\mathcal{H}_{earn}$ consists of the 2954 positive examples and the 113 negative ones lying in the classification region $\mathrm{R}_{\mathcal{H}_{earn}}$ delimited by $DB_{\mathcal{H}_{earn}}$. The generalization error and the *PRavg* value of $\mathcal{H}_{earn}$ are

$$err = \frac{b+c}{a+b+c+d} = 0.031; \qquad PRavg = \frac{2a^2 + a + ac}{2(a+b)(a+c)} = 0.94$$

Table 3
Micro-averaged PRavg values on each data set obtained by GAMoN and GAMoN* (a version of GAMoN with no GS reproduction operators). Legend – BG: BlogsGender, OhS: Ohscale, Ohsu: Ohsumed, Mkt: Market.

	Oh15	Oh5	Oh0	Oh10	BG	OhS	R10	20NG	Ohsu	cade	SRAA	ODP	Mkt
GAMoN	80.46	84.85	84.27	78.62	68.82	75.03	86.50	75.72	67.25	48.42	85.25	71.14	92.42
GAMoN*	77.59	82.55	82.90	75.54	68.33	74.28	84.56	75.21	67.09	46.43	84.27	70.20	90.59

Table 4
Micro-averaged PRavg results obtained by 5-fold cross-validation on Oh0, Oh5, Oh10, Oh15, BlogsGender, Ohscale and R10 (80/20 split) and by holdout on the remaining data sets (70/30 split).

Dataset	Ripper	SMO	C4.5	OlexGA	BioHEL	GAMoN
Oh15	79.55	79.95	76.75	74.33	66.03	**80.46**
Oh5	83.74	84.29	82.30	80.76	76.72	**84.85**
Oh0	84.37	**84.80**	79.24	81.29	73.20	84.27
Oh10	**78.82**	74.70	74.78	74.46	67.39	78.62
BlogsGender	60.96	60.95	58.33	66.75	62.82	**68.82**
Ohscale	72.96	69.52	70.77	74.36	68.26	**75.03**
R10	85.21	**88.94**	84.67	84.07	83.59	86.50
20NG	72.66	**83.64**	74.86	72.97	70.65	75.72
Ohsumed	60.35	66.94	63.25	65.58	63.79	**67.25**
cade	44.31	**54.06**	48.10	44.10	42.96	48.42
SRAA	81.04	**90.06**	86.85	79.60	81.34	85.25
ODP	66.73	**80.65**	74.76	69.46	71.21	71.14
Market	94.63	**95.56**	95.37	88.95	74.75	92.42
avg microPRavg	74.26	78.00	74.62	73.59	69.44	76.83
avg rank	3.96	**2.08**	3.62	4.31	5.23	**2.08**

where a, b, c and d are computed as follows (see Section 4.5):

$$a = \sum_{(x,y)\in \mathrm{R}_{\mathcal{H}_{hearn}}} \pi(x,y) = 2954, \qquad b = \sum_{(x,y)\in \mathrm{R}_{\mathcal{H}_{hearn}}} \nu(x,y) = 113,$$

$$c = \sum_{(x,y)\notin \mathrm{R}_{\mathcal{H}_{hearn}}} \pi(x,y) = 205, \qquad d = \sum_{(x,y)\notin \mathrm{R}_{\mathcal{H}_{hearn}}} \nu(x,y) = 6988.$$

9.4. *Effect of GS operators*

To see the effect of the GS reproduction operators defined in this paper, we compared the accuracy results of GAMoN over the data sets previously seen with those obtained by running a version of GAMoN where the GS operators (GS Xover and GS mutation) were disabled. The experimental results, reported in Table 3, show that the GS operators improve the accuracy on every single data set, even though on some they induce only trivial improvements (e.g., Ohsu and BG), while on other the gain is remarkable (e.g., Oh15, Oh10, R10, cade and Mkt). On average, the enhancement over all data sets is of around 1.5 points. As discussed earlier in this paper, the aim of GS operators is that of further refining fitter individuals by exploiting the structure of the hypothesis space. This explains why often significant improvements are obtained.

9.5. *Comparison with other systems*

Table 4 shows, for each algorithm and data set, the micro-averaged PRavg. The average values over all data sets along with the average ranking of each algorithm, are also included (bottom of the table). The best results are stressed in boldface. The ranking is obtained by assigning a position to each algorithm depending on its performance on each data set. The algorithm showing the best accuracy on a given data set is assigned rank 1.

As we can see, SMO is the best performer (PRavg = 78.00), followed by GAMoN (PRavg = 76.83). The two algorithms, however, show the same average rank (2.08). We note that GAMoN outperforms all the other rule induction methods. In particular, it behaves uniformly better than OlexGA on all individual data sets, and compares favorably with the other rule learners on most of the data sets.

In order to establish whether the above differences in performance are statistically significant, the Iman–Davenport's test is applied. This is a non-parametric statistical test recommended in [58] for comparing two or more classifiers on multiple data sets. In brief, with 13 data sets, 6 algorithms and confidence $\alpha = 0.05$, the Iman–Davenport statistics is 9.54, greater than the critical value $CV = 2.37$. Thus, the null hypothesis (which states that all the algorithms are equivalent) is rejected. Hence, we apply the Holm's *post-hoc* test [58], with GAMoN as control algorithm, for controlling the family-wise error in multiple hypothesis testing. The results of this test are summarized in Table 5. Based on them, we can reject the null

Table 5
Holm's test with GAMoN as control algorithm. The null hypothesis is rejected when p-value $< \alpha/i$.

i	Method	$z = \frac{(R_0 - R_i)}{SE}$	p-value	α/i
5	BioHEL	−4.2980	0.0005	0.01
4	OlexGA	−3.0400	0.0024	0.0125
3	Ripper	−2.2014	0.0278	0.0167
2	C4.5	−2.0966	0.0366	0.025
1	SMO	0.0000	1.0000	0.05

Table 6
Avg size of the rule-based classifiers on R10.

Algorithm	Avg size of classifiers
GAMoN	#Pos = 20, #Neg = 10, order = 1.8
Ripper	#Rules = 16
C4.5	#Rules = 78
BioHEL	#Rules = 14, #literals/rule = 19
OlexGA	#Pos = 16, #Neg = 15

Table 7
Learning times expressed in hours. Each run of GAMoN, OlexGA and BioHEL was repeated 3 times.

	Ripper	C4.5	SMO	OlexGA	BioHEL	Gamon
Overall learning time (h)	445	488	71	46	185	156
# runs	395	395	395	1185	1185	1185
Avg learning time per category (h)	1.13	1.23	0.18	0.04	0.16	0.12

hypothesis of equivalence only for BioHEL and OlexGA (as p-value $< \alpha/i$ holds). That is, with confidence 95%, we can state that GAMoN performs better than such algorithms, while it is statistically equivalent to SMO, Ripper and C4.5.

9.6. Size of the classifiers

Apart from SMO, the other classifiers yield models as sets of rules. Although we do not have a unique formal definition of size of a classifier, being either the number of rules, number of features, etc., in Table 6 we provide some statistical data (averaged over the five folds) giving an insight into the quantitative characteristics of the classifiers induced on R10. As we can see, Ripper, C4.5 and BioHEL induce classifiers consisting on average of 16 rules, 78 rules and 14 rules, respectively, each BioHEL rule having 19 literals on average. In turn, GAMoN induces classifiers of 20 positive features and 10 negative ones on average, against the 16 positive features and 15 negative ones of OlexGA classifiers. Going beyond the results given in the table, nearly 44% of the classifiers induced by GAMoN are atoms, 38% are of order 2, 18% of order 3 and 2% of order 4 (note that with $P = N = 4$, the maximum order of a classifier is $Min(P + 1, N) = 4$ – see Proposition 4.10).

9.7. Time efficiency

The experiments previously described were performed on an Intel Xeon 2.33 GHz machine with 4 Gb RAM.

The learning times needed to achieve the accuracy results previously seen are reported for each method in Table 7 – first row. As we can see, OlexGA (46 hours) is the best performers, followed by SMO (71), GAMoN (156), BioHEL (185), Ripper (445) and C4.5 (488) (recall that each run of GAMoN, OlexGA and BioHEL was repeated 3 times). Table 7 also reports the average learning times per category. Again, OlexGA is the fastest algorithm (0.04 h/category), followed by GAMoN (0.12), BioHEL (0.16) and SMO (0.18). Ripper and C4.5 are ten times slower than GAMoN (1.13 and 1.23, respectively).

To see the effect of the training set size over learning times, in Fig. 10 we plotted the average learning times per category over each data set (data sets are ordered by increasing size). The graph provides an empirical picture of the progression of learning times with the number of training documents. As we can see, GAMoN asymptotically behaves similarly to OlexGA, BioHEL and SMO, while its has a significantly smoother trend than both Ripper and C4.5. That is, GAMoN scales better than the two non-evolutionary rule induction methods.

10. Discussion and related work

The experimental study described in the previous sections shows that GAMoN induces classifiers that are both accurate and compact. Interestingly, these properties have consistently been observed over all 13 data sets, on which GAMoN showed a uniform behavior. Given the very different application domains the corpora refer to, this is a clear proof of robustness. Further, GAMoN showed to perform efficiently on large data sets.

M-of-$N^{\{\neg,\vee\}}$ representation. As discussed in Section 4.6, the "family resemblance" metaphor provides us with a qualitative understanding on the basic reason why the M-of-N paradigm is well suited for the purpose of text categorization.

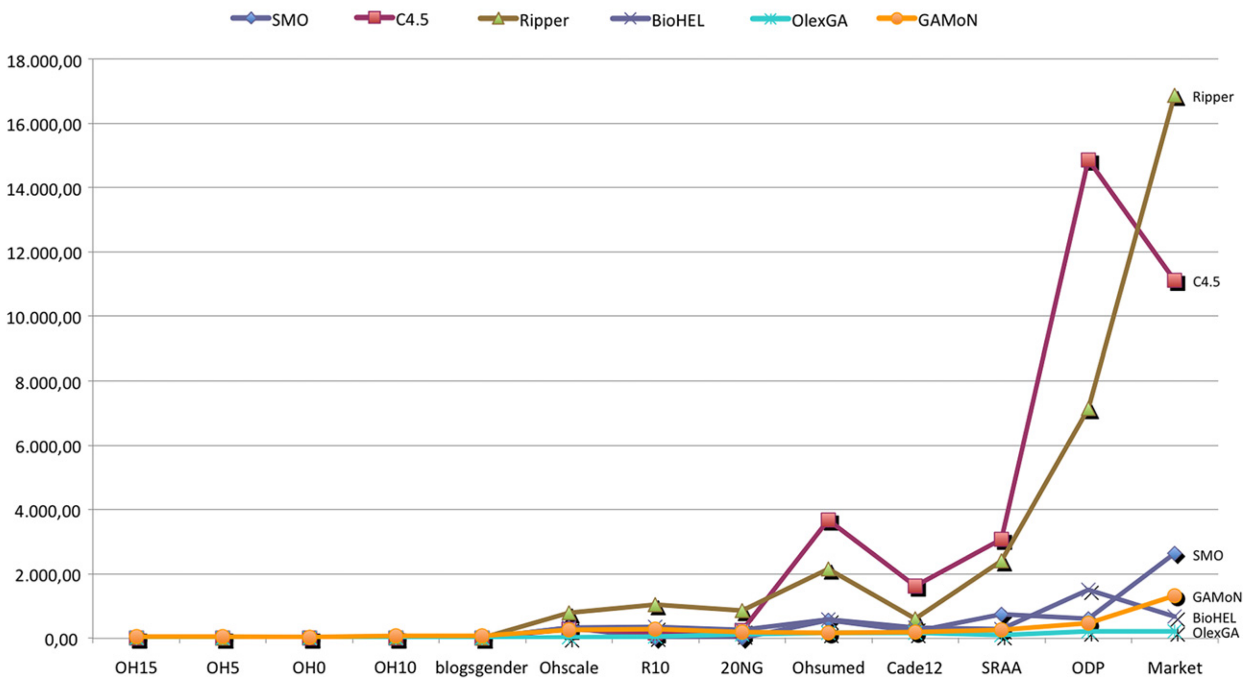


Fig. 10. Comparison of the average learning time per category over each data set (data sets are ordered by increasing size).

M-of-$N^{\{\neg,\vee\}}$ extends M-of-N through negation and disjunction, two constructs that enables to express hypotheses capable of best fitting the true structure of the data (a discussion on the expressivity of M-of-$N^{\{\neg,\vee\}}$ has been reported in Section 4.6). Unlike most of the existing classifiers which focus on features that positively discriminate a class, in our approach negation is used as a “first class citizen” allowing us to explicitly model the interactions between positive and negative features within a given example. In turn, disjunction enables to “modulate” such interactions, by capturing the positive correlation (that simple atoms would miss) existing between positive and negative features (see Section 4.5). Negation takes precision under control, while disjunction improves recall.

One advantage of the proposed language over a DNF-type representation is conciseness. Indeed, although an M-of-$N^{\{\neg,\vee\}}$ hypothesis can be represented in terms of disjunctions of conjunctions (see Section 4.2), a DNF-type representation of an M-of-$N^{\{\neg,\vee\}}$ concept would be prohibitively long (the number of disjuncts is exponential in the size of *Pos* and *Neg*). We believe that the proposed language is one main contribution of this paper. One interesting direction for future work could be that of using a mathematical relationship (e.g., a linear function) between thresholds p and n, instead of the current threshold multiple pairs.

Feature space. M-of-$N^{\{\neg,\vee\}}$ hypotheses are built over a set of pre-selected candidate features. While there has been a long history of applying dimensionality reduction methods, one contribution of this paper is represented by an original definition of the feature space, as consisting of both terms indicative of membership and terms indicative of non-membership for a category. Unlike in the traditional feature selection approach, where only positive terms are selected, using our definition enables the learner to focus on negative information in the same way as it does with positive one (a similar approach distinguishing between positive and negative features was proposed in [59]). A criterion for the automatic detection of a suitable dimension has also been provided (see Definition 7.1). This criterion proved to be very effective in practice. Experimental results showed indeed that a few tens of well-selected features are sufficient to build accurate prediction functions, irrespective of the data set.

GAMoN biases. Apart from the language bias, we can characterize GAMoN in terms of both a search bias and an overfitting avoidance bias [18]. We have extensively discussed about the former, which refers to the way the hypothesis space is searched through the subsumption relationships by means of the task-dependent genetic operators. The proposed approach, actually not new in inductive learning (see, e.g., [30–35]), overcomes a major problem in the use of conventional GAs which do not take into account the structure of the search space. The overfitting avoidance bias is a preference for simpler classifiers. GAMoN includes such a bias in the induction mechanisms by using suitable feature probability distributions (see Section 7.4) which enable the reproduction operators to select few, high-quality features. In combination with the proposed feature selection technique, which provides GAMoN with an effective lexicon, capable of expressing the essential patterns, the overfitting avoidance bias guarantees the induction of classifiers that are parsimonious, made of a handful of well-selected features. This makes them effective on the unseen data, as few high-quality features drastically reduce the risk of overfitting the training data.

GAMoN, C4.5 and Ripper. Unlike GAMoN, the two rule-based non-evolutionary classifiers used in this work, notably, C4.5 and Ripper, achieve the full expressive power of DNF. Despite this, the conducted experimental study showed that they do not outperform GAMoN. This is a clear proof of the effectiveness of the proposed algorithm.

In addition, GAMoN performs significantly more efficiently than both C4.5 and Ripper on large data sets, as the graphs of Fig. 10 show. This should not be surprising, as the time complexity of Ripper is $O(m \log^2 m)$ and that of C4.5 $O(m^3)$, while the complexity of GAMoN is $O(m)$, where m the number of examples in the training set (see Section 7.6). That is, GAMoN can scale up to large and realistic real-world domains better than the other two rule-based classifiers. We point out that further improvements of the learning times may be obtained by, e.g., a more efficient implementation of the task-specific reproduction operators and, in a real application environment, by distributed approaches. Current research is likely to further improve efficiency of GAMoN.

GAMoN, OlexGA and BioHEL. GAMoN is a substantial extension of OlexGA from two respects. The first is the language. An OlexGA hypothesis is the special case of an M-of-$N^{\{\neg,\vee\}}$ atom where both thresholds are equal to 1, i.e., $\langle Pos, Neg, (1,1)\rangle$ (thus, M-of-$N^{\{\neg,\vee\}}$ is strictly more expressive than the language of OlexGA). The second is the genetic algorithm. Since the hypothesis space of OlexGA does not provide any structure, OlexGA relies on a simple, standard GA, where the population is made of fixed-length individuals, and the reproduction operators are the standard uniform crossover and mutation. That is, OlexGA is a special case of GAMoN. As shown in Table 4, the proposed extension results in a statistically significant improvement over OlexGA. Needless to say, the price for that is a slower learning procedure.

BioHEL (Bioinformatics-oriented Hierarchical Evolutionary Learning) is a state-of-the-art GA which showed to perform very effectively on non-textual data sets [60]. To the best of our knowledge, this is the first study where BioHEL has been tested on text classification problems. BioHEL inherits several features from GAssist. It relies on the Pittsburgh representation approach and applies the iterative rule learning approach [28]. BioHEL was explicitly designed to handle large-scale datasets. To this end, a rule, instead of coding all the domain attributes, keeps only a subset of them, thus avoiding hundreds of irrelevant computations. Using such an approach, BioHEL is able to handle problems with hundreds of attributes (in datasets with large sets of instances [13]) or even tens of thousands of attributes (but with few instances). In addition, in order to further reduce the computational cost, BioHEL uses a windowing scheme called ILAS (incremental learning with alternating strata). The experimental results of this paper confirm that BioHEL behaves quite efficiently, with a learning time similar to that of GAMoN (see Table 7). On the contrary, in terms of predictive accuracy, it showed to be statistically inferior to GAMoN. However, we feel that better results could be obtained by a finer tuning of the system. For an instance, a recent publication [62] shows that BioHEL has a parameter which is highly problem sensitive, the coverage breakpoint. Also, an appropriate use of the ILAS windowing scheme, as well as the usage of the C++ implementation[4] (in place of the KEEL implementation), could further improve efficiency.

Other systems learning with negation. As already mentioned, using negative evidence is deemed important in the text classification task. However, apart from OlexGA and GAMoN, none of the experimented systems focuses on the exploitation of negative information. In general, examples of IRL (Inductive Rule Learning) approaches that involve the direct generation of negation are very rare (see, e.g., [61,38]). Outside the realm of rule learners, Complement Naive Bayes (CNB) is among the few text classifiers that leverage negative features [21]. Its peculiarity is that of learning the weights for a class using all training data not in that class. CNB works in a multi-class setting (i.e., it needs at least 3 classes). In [21], the authors claim that CNB approaches the state-of-the-art accuracy of SVMs. Unfortunately, we could not compare GAMoN with CNB in our empirical study, as the (binary) one-versus-all technique was used to deal with multi-label classification (basically, the problem was that Weka does not provide support for a multi-label data set representation that would be necessary in order to provide CNB with the *same* input of the other systems).

Other M-of-N approaches. Several research works have recently been done to develop methods for inducing M-of-N concepts but, to the best of our knowledge, none for text categorization. For an instance, in [1] a technique for extracting M-of-N hypotheses from neural networks is reported. However, most work in this field has been carried out for constructive induction. ID-2-of-3 [2] is an M-of-N induction algorithm which incorporates M-of-N tests in decision-tree learning. It is based on a (greedy) hill-climbing approach to get the best M-of-N hypotheses at each node of a decision tree. XofN [4] is another greedy constructive induction algorithm that learns X-of-N nominal attributes. Both ID-2-of-3 and XofN, when building a decision tree, construct a new attribute for each decision node using the local training set. More recently, a Genetic Algorithm for constructive induction has been proposed in [7]. It relies on a variable length individual representation encoding the set of N attribute-value pairs composing an X-of-N attribute. The fitness is defined as the information gain ratio of the constructed attribute. The genetic operators are the standard uniform crossover along with a mutation which is a simple variant of the standard one. A conventional niching method to foster population diversity is also used.

11. Conclusions

In this paper we proposed a new language, called M-of-$N^{\{\neg,\vee\}}$, for text classification, along with a GA-based approach for constructing M-of-$N^{\{\neg,\vee\}}$ hypotheses from training data.

[4] Available at http://icos.cs.nott.ac.uk/software/biohel.html.

The M-of-$N^{\{\neg,\vee\}}$ representation generalizes the classical notion of M-of-N concepts by allowing negation and disjunction. We conjectured that it is well-suited to express text classification conditions, as it complies with the so-called "family resemblance" metaphor. We have shown that the space of M-of-$N^{\{\neg,\vee\}}$ hypotheses has a structure determined by two kinds of subsumption relationships – the *feature* and the *threshold* relationships, that form complete lattices. Based on that, suitable refinement operators for an effective exploration of the hypothesis space were designed.

To induce M-of-$N^{\{\neg,\vee\}}$ hypotheses, the task-specific genetic algorithm GAMoN was proposed. It is based on the Pittsburgh approach, where an individual encodes a candidate classifier, as well as on *ad hoc* GS reproduction operators which are a stochastic implementation of the refinement operators. GAMoN dynamically adapts the probability of selecting the GS operators. The population is partitioned into a number of competing subpopulations, each consisting of individuals belonging to the same hypothesis subspace. To this end, a statistical criterion for automatically detecting the dimensionality range of the feature space has been proposed.

This paper also presented empirical results obtained by extensive experiments on 13 real-world test collections in a wide spectrum of sizes – from a few hundreds to a few hundreds thousands of documents. We found that GAMoN is competitive with a large collection of state-of-the-art learning techniques belonging to different classes, and that it provides hypotheses that are compact and easily interpretable. In particular, though there are small differences in predictive accuracy between GAMoN and SMO (the latter being a bit more performant), and between GAMoN and both Ripper and C4.5 (the latter two being a bit less performant), all such systems showed to be statistically equivalent. Whereas, GAMoN proved to be superior to the other evolutionary algorithms. In particular, it showed statistically significant improvements over its predecessor OlexGA, thus confirming the effectiveness of the proposed extension. Finally, we observed that, as we scale up the size of the data set, GAMoN performs much more efficiently then both Ripper and C4.5.

Acknowledgement

The authors wish to thank Giuseppe Manco for his helpful comments on earlier versions of this paper.

Appendix A. Proofs

Proof of Proposition 4.1. Let $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ be two classifiers in $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$. Next we show that $\mathcal{H}_c^1 \succcurlyeq_\phi \mathcal{H}_c^2$ implies $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$. The proof proceeds by induction. (*Basis*) $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ are atoms in $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$ of the form, say, $\langle Pos_1, Neg_1, \{(p,n)\}\rangle$ and $\langle Pos_2, Neg_2, \{(p,n)\}\rangle$. A document d is classified by $\mathcal{H}_c^2$ iff $|d \cap Pos_2| \geqslant p$ and $|d \cap Neg_2| \leqslant n$ (recall that a document is a set of features – see Section 4.1). It can be easily seen that, since both $Pos_2 \subseteq Pos_1$ and $Neg_1 \subseteq Neg_2$ hold by hypothesis, $|d \cap Pos_1| \geqslant p$ and $|d \cap Neg_1| \leqslant n$ is verified as well, that is, d is classified by $\mathcal{H}_c^1$. (*Inductive step*) $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ are two generic classifiers $\langle Pos_1, Neg_1, \mathcal{T}\rangle$ and $\langle Pos_2, Neg_2, \mathcal{T}\rangle$ in $\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k))$ such that $\mathcal{H}_c^1 \succcurlyeq_\phi \mathcal{H}_c^2$. Thus, they can be expressed in the following form: $\mathcal{H}_c^1 = \mathcal{H}_c^{1,1} \vee \mathcal{H}_c^{1,2}$ and $\mathcal{H}_c^2 = \mathcal{H}_c^{2,1} \vee \mathcal{H}_c^{2,2}$, where $\mathcal{H}_c^{1,1} = \langle Pos_1, Neg_1, \mathcal{T}_1\rangle$, $\mathcal{H}_c^{1,2} = \langle Pos_1, Neg_1, \mathcal{T}_2\rangle$, $\mathcal{H}_c^{2,1} = \langle Pos_2, Neg_2, \mathcal{T}_1\rangle$ and $\mathcal{H}_c^{2,2} = \langle Pos_2, Neg_2, \mathcal{T}_2\rangle$. By inductive hypothesis, since both $Pos_2 \subseteq Pos_1$ and $Neg_1 \subseteq Neg_2$ hold, any document classified by $\mathcal{H}_c^{2,1}$ is classified by $\mathcal{H}_c^{1,1}$ and any document classified by $\mathcal{H}_c^{2,2}$ is classified by $\mathcal{H}_c^{1,2}$. It turns out that $\mathcal{H}_c^1$ classifies all documents classified by $\mathcal{H}_c^2$, i.e., $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$. $\square$

Proof of Proposition 4.2. Next we show that $(\mathbf{H}_{\mathcal{T}}(\mathcal{F}_c(k)), \succcurlyeq_\phi)$ is a complete lattice. To this end, we first prove statement (a) – $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos_1 \cup Pos_2, Neg_1 \cap Neg_2, \mathcal{T}\rangle$. From Definition 4.4 it immediately follows that both $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_\phi \mathcal{H}_c^1$ and $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_\phi \mathcal{H}_c^2$ hold. Now let us assume, by absurd, the existence of $\mathcal{H}_c' = \langle Pos, Neg, \mathcal{T}\rangle$ such that $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_\phi H_c'$ and, further, $H_c' \succcurlyeq_\phi \mathcal{H}_c^1$ and $H_c' \succcurlyeq_\phi \mathcal{H}_c^2$. From $lub_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) \succcurlyeq_\phi H_c'$ we have that $Pos \subseteq Pos_1 \cup Pos_2$ (see Definition 4.4). However, if so, the conditions $H_c' \succcurlyeq_\phi \mathcal{H}_c^1$ and $H_c' \succcurlyeq_\phi \mathcal{H}_c^2$ cannot hold, as $Pos_1 \subseteq Pos$ and $Pos_2 \subseteq Pos$ cannot be both true (a contradiction). From which statement (a) follows. Now we prove statement (b) – $glb_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos_1 \cap Pos_2, Neg_1 \cup Neg_2, \mathcal{T}\rangle$. By Definition 4.4 we have that $\mathcal{H}_c^1 \succcurlyeq_\phi glb_\phi$ and $\mathcal{H}_c^2 \succcurlyeq_\phi glb_\phi$. Now let us assume, by absurd, the existence of $\mathcal{H}_c' = \langle Pos, Neg, \mathcal{T}\rangle$ such that $H_c' \succcurlyeq_\phi glb_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2)$ and, further, $\mathcal{H}_c^1 \succcurlyeq_\phi H_c'$ and $\mathcal{H}_c^2 \succcurlyeq_\phi H_c'$. From $H_c' \succcurlyeq_\phi glb_\phi(\mathcal{H}_c^1, \mathcal{H}_c^2)$ it turns out that $Pos_1 \cap Pos_2 \subseteq Pos$. But, if so, the conditions $\mathcal{H}_c^1 \succcurlyeq_\phi H_c'$ and $\mathcal{H}_c^2 \succcurlyeq_\phi H_c'$ cannot hold, as Pos is not a subset of both Pos_1 and Pos_2 (a contradiction). From which statement (b) follows. $\square$

Proof of Proposition 4.3. Let $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ be two classifiers in $\mathbf{H}_\Phi(P,N)$. Next we show that $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ implies $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$. The proof proceeds by induction. (*Basis*) $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ are atoms of the form, say, $\langle Pos, Neg, \{(p_1,n_1)\}\rangle$ and $\langle Pos, Neg, \{(p_2,n_2)\}\rangle$. By Definition 4.2, a document d is classified by $\mathcal{H}_c^2$ if (and only if) $|d \cap Pos| \geqslant p_2$ and $|d \cap Neg| \leqslant n_2$. Now, $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ only if $p_1 \leqslant p_2$ and $n_1 \geqslant n_2$ (by Definition 4.5), which implies that a document d classified by $\mathcal{H}_c^2$ is classified by $\mathcal{H}_c^1$ as well, i.e., $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$. (*Inductive step*) $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ only if, for each atom $\mathcal{H}_c^{2,i}$ appearing in $\mathcal{H}_c^2$ there exists an atom $\mathcal{H}_c^{1,j}$ appearing in $\mathcal{H}_c^1$ such that $\mathcal{H}_c^{1,j} \succcurlyeq_\tau \mathcal{H}_c^{2,i}$ (immediate from Definition 4.5). Since $\mathcal{H}_c^{1,j}$ classifies all documents classified by $\mathcal{H}_c^{2,i}$ (inductive hypothesis), it follows that $\mathcal{H}_c^1$ classifies all documents classified by $\mathcal{H}_c^2$. $\square$

Proof of Lemma 4.1. Next we prove that, given $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\hat{\mathcal{H}}_c = \hat{\mathcal{H}}_c^1 \vee \hat{\mathcal{H}}_c^2$, the following holds: $D(\mathcal{H}_c) \supseteq D(\hat{\mathcal{H}}_c)$ only if $(D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^1)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1))$ and $(D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^2)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^2))$. The proof proceeds by induction.

(*Basis*) $\mathcal{H}_c^1 = \{(p_1, n_1)\}$, $\mathcal{H}_c^2 = \{(p_2, n_2)\}$ and $\hat{\mathcal{H}}_c = \{(\hat{p}, \hat{n})\}$ are atoms. By Definition 4.2, $D(\mathcal{H}_c) = \{d \in D \text{ s.t. } |d \cap Pos| \geqslant p_1 \wedge |d \cap Neg| \leqslant n_1 \vee |d \cap Pos| \geqslant p_2 \wedge |d \cap Neg| \leqslant n_2\}$ and $D(\hat{\mathcal{H}}_c) = \{d \in D \text{ s.t. } |d \cap Pos| \geqslant \hat{p} \wedge |d \cap Neg| \leqslant \hat{n}\}$. Thus, $D(\mathcal{H}) \supseteq D(\hat{\mathcal{H}}_c)$ only if either (1) $\hat{p} \geqslant p_1$ and $\hat{n} \leqslant n_1$ or (2) $\hat{p} \geqslant p2$ and $\hat{n} \leqslant n_2$. By Definition 4.5, condition (1) entails $\mathcal{H}_c^1 \succcurlyeq_\tau \hat{\mathcal{H}}_c$ and condition (2) $\mathcal{H}_c^2 \succcurlyeq_\tau \hat{\mathcal{H}}_c$, so as $D(\mathcal{H}) \supseteq D(\hat{\mathcal{H}}_c)$ only if $D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c)$. (*Inductive step*) $D(\mathcal{H}) \supseteq D(\hat{\mathcal{H}}_c)$ only if $D(\mathcal{H}_c^1) \cup D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1) \cup D(\hat{\mathcal{H}}_c^2)$ only if $D(\mathcal{H}_c^1) \cup D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1)$ and $D(\mathcal{H}_c^1) \cup D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^2)$ only if (by inductive hypothesis) $D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^1)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1)$ and $D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^2)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^2)$. □

Proof of Proposition 4.4. Let $\mathcal{H}_c$ and $\mathcal{H}'_c$ be two classifiers in $\mathbf{H}_\Phi(P, N)$. We next show that $D(\mathcal{H}_c) \supseteq D(\mathcal{H}'_c)$ implies $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}'_c$. The proof proceeds by induction. (*Basis*) $\mathcal{H}_c = \{(p, n)\}$ and $\mathcal{H}'_c = \{(p', n')\}$ are atoms. By Definition 4.2, $D(\mathcal{H}_c) = \{d \in D \text{ s.t. } d \cap Pos| \geqslant p \wedge |d \cap Neg| \leqslant n\}$ and $D(\mathcal{H}'_c) = \{d \in D \text{ s.t. } d \cap Pos| \geqslant p' \wedge |d \cap Neg| \leqslant n'\}$. Clearly, $D(\mathcal{H}_c) \supseteq D(\mathcal{H}'_c)$ only if $p \leqslant p'$ and $n \geqslant n'$, that is, only if $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}'_c$. (*Inductive step*) Let $\mathcal{H}_c = \mathcal{H}_1 \vee \mathcal{H}_2$ and $\mathcal{H}'_c = \mathcal{H}'_1 \vee \mathcal{H}'_2$. Now, $D(\mathcal{H}_c) \supseteq D(\mathcal{H}'_c)$ only if $D(\mathcal{H}_1) \cup D(\mathcal{H}_2) \supseteq D(\mathcal{H}'_1) \cup D(H'_2)$ only if (by Lemma 4.1) $(D(\mathcal{H}_1) \supseteq D(\mathcal{H}'_1) \vee D(\mathcal{H}_2) \supseteq D(\mathcal{H}'_1)) \wedge (D(\mathcal{H}_1) \supseteq D(\mathcal{H}'_2) \vee D(\mathcal{H}_2) \supseteq D(H'_2))$ only if (by inductive hypothesis) $\mathcal{H}_1 \succcurlyeq_\tau \mathcal{H}'_1$ or $\mathcal{H}_2 \succcurlyeq_\tau \mathcal{H}'_1$ and $\mathcal{H}_1 \succcurlyeq_\tau \mathcal{H}'_2$ or $\mathcal{H}_2 \succcurlyeq_\tau \mathcal{H}'_2$ only if $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}'_c$. □

Proof of Corollary 4.1. We next prove that, given classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$, $\mathcal{H}_c^1 \equiv \mathcal{H}_c^2$ iff $D(\mathcal{H}_c^1) = D(\mathcal{H}_c^2)$. Indeed, $\mathcal{H}_c^1 \equiv \mathcal{H}_c^2$ iff $\mathcal{H}_c^1 \succcurlyeq \mathcal{H}_c^2$ and $\mathcal{H}_c^2 \succcurlyeq \mathcal{H}_c^1$ iff $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$ and $D(\mathcal{H}_c^2) \supseteq D(\mathcal{H}_c^1)$ (by Proposition 4.3 and Proposition 4.4) iff $D(\mathcal{H}_c^1) = D(\mathcal{H}_c^2)$. □

Proof of Proposition 4.6. Next we show that, given $\mathcal{H}_c, \hat{\mathcal{H}}_c \in \mathbf{H}_\Phi(P, N)$, the classifier $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ is such that (1) $and(\mathcal{H}_c, \hat{\mathcal{H}}_c) \in \mathbf{H}_\Phi(P, N)$, (2) $D(and(\mathcal{H}_c, \hat{\mathcal{H}}_c)) = D(\mathcal{H}_c) \cap D(\hat{\mathcal{H}}_c)$, and (3) $\mathcal{H}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ and $\hat{\mathcal{H}}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$. The proof proceeds by induction. (*Basis*) $\mathcal{H}_c = \{(p, n)\}$ and $\hat{\mathcal{H}}_c = \{(\hat{p}, \hat{n})\}$ are atoms. *Statement* (1). To show that $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ is in $\mathbf{H}_\Phi(P, N)$ it suffices to observe that both $p = Max\{p, \hat{p}\} \leqslant P$ and $n = Min\{n, \hat{n}\} \leqslant N$ hold. *Statement* (2). A document d is classified by $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ iff d contains $x \geqslant Max\{p, \hat{p}\}$ positive features and $y < Min\{n_1, \hat{n}\}$ negative features, iff $x \geqslant p$, $x \geqslant \hat{p}$, $y < n$ and $y < \hat{n}$, iff d is classified by both $\mathcal{H}_c$ and $\hat{\mathcal{H}}_c$, i.e., $D(\mathcal{H}_c) = D(\mathcal{H}_c) \cap D(\hat{\mathcal{H}}_c)$. *Statement* (3). Immediate from Statement 1 and Proposition 4.4. (*Inductive step*) Let $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\hat{\mathcal{H}}_c = \hat{\mathcal{H}}_c^1 \vee \hat{\mathcal{H}}_c^2$ be two classifiers. *Statement* (1). From Definition 4.7, $and(\mathcal{H}_c, \hat{\mathcal{H}}_c) = \mathcal{H}_1 \vee \mathcal{H}_2 \vee \mathcal{H}_3 \vee \mathcal{H}_4$, where $\mathcal{H}_1 = and(\mathcal{H}_c^1, \hat{\mathcal{H}}_c^1)$, $\mathcal{H}_2 = and(\mathcal{H}_c^1, \hat{\mathcal{H}}_c^2)$, $\mathcal{H}_3 = and(\mathcal{H}_c^2, \hat{\mathcal{H}}_c^1)$ and $\mathcal{H}_4 = and(\mathcal{H}_c^2, \hat{\mathcal{H}}_c^2)$. By the inductive hypothesis, $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4$ are in $\mathbf{H}_\Phi(P, N)$ and thus, by Definition 4.2, $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ is in $\mathbf{H}_\Phi(P, N)$. *Statement* (2). By using the inductive step of Definition 4.7, along with the inductive hypothesis of Statement (2), we get $D(and(\mathcal{H}_c, \hat{\mathcal{H}}_c)) = D(\mathcal{H}_c^1) \cap D(\hat{\mathcal{H}}_c^1) \cup D(\mathcal{H}_c^1) \cap D(\hat{\mathcal{H}}_c^2) \cup D(\mathcal{H}_c^2) \cap D(\hat{\mathcal{H}}_c^1) \cup D(\mathcal{H}_c^2) \cap D(\hat{\mathcal{H}}_c^2)$, from which $D(and(\mathcal{H}_c, \hat{\mathcal{H}}_c)) = D(\mathcal{H}_c^1) \cap D(\mathcal{H}_c^2)$ immediately follows. *Statement* (3). By using the inductive step of Definition 4.7, and applying the inductive hypothesis of Statement (3), we have that $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_1$ and $\hat{\mathcal{H}}_c^1 \succcurlyeq_\tau \mathcal{H}_1$ (as $\mathcal{H}_1 = and(\mathcal{H}_c^1, \hat{\mathcal{H}}_c^1)$), $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_2$ and $\hat{\mathcal{H}}_c^2 \succcurlyeq_\tau \mathcal{H}_2$ (as $\mathcal{H}_2 = and(\mathcal{H}_c^1, \hat{\mathcal{H}}_c^2)$), $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_3$ and $\hat{\mathcal{H}}_c^1 \succcurlyeq_\tau \mathcal{H}_3$ (as $\mathcal{H}_3 = and(\mathcal{H}_c^2, \hat{\mathcal{H}}_c^1)$), $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_4$ and $\hat{\mathcal{H}}_c^2 \succcurlyeq_\tau \mathcal{H}_4$ (as $\mathcal{H}_4 = and(\mathcal{H}_c^2, \hat{\mathcal{H}}_c^2)$). Thus, by Definition 4.5, it follows that both $\mathcal{H}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ and $\hat{\mathcal{H}}_c \succcurlyeq_\tau and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ hold (that is, $and(\mathcal{H}_c, \hat{\mathcal{H}}_c)$ is more specific than both $\mathcal{H}_c$ and $\hat{\mathcal{H}}_c$). □

Proof of Proposition 4.5. Let $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$ be given. We next show that the following properties hold:

1. $\mathcal{H}_c$ is redundant iff there exist $(p_i, n_i), (p_j, n_j) \in \mathcal{T}$ such that $\{(p_i, n_i)\} \succcurlyeq_\tau \{(p_j, n_j)\}$.
2. $\mathcal{H}_c$ is minimal iff $\mathcal{T} = \{(p_1, n_1), \ldots, (p_r, n_r)\}$ is such that $p_i < p_j$ and $n_i < n_j$, or vice versa, for each $i, j \in [1, r]$.
3. If $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$, then $\mathcal{H}_c \equiv \mathcal{H}_c^1$.

(1) Let $\mathcal{H}_c = \mathcal{H}_c^1 \vee \cdots \vee \mathcal{H}_c^r$. By Definition 4.5, $\{(p_i, n_i)\} \succcurlyeq_\tau \{(p_j, n_j)\}$ iff $\mathcal{H}_c^i \succcurlyeq \mathcal{H}_c^j$, with $i, j \in [1, r]$, iff $\mathcal{H}_c = \mathcal{H}'_c \vee \mathcal{H}_c^j$ and $\mathcal{H}'_c \succcurlyeq \mathcal{H}_c^j$, where $\mathcal{H}'_c = \mathcal{H}_c^1 \vee \cdots \vee \mathcal{H}_c^{j-1} \vee \mathcal{H}_c^{j+1} \vee \cdots \vee \mathcal{H}_c^i \vee \cdots \vee \mathcal{H}_c^r$, iff $\mathcal{H}_c$ is redundant (by Definition 4.6).

(2) From point 1 above, $\mathcal{H}_c$ is minimal iff for each pair $(p_i, n_i), (p_j, n_j) \in \mathcal{T}$ neither $\{(p_i, n_i)\} \succcurlyeq_\tau \{(p_j, n_j)\}$ nor $\{(p_j, n_j)\} \succcurlyeq_\tau \{(p_i, n_i)\}$ iff neither $(p_i \leqslant p_j$ and $n_i \geqslant n_j)$ nor $(p_j \leqslant p_i$ and $n_j \geqslant n_i)$ (by Definition 4.5) iff $p_i < p_j$ and $n_i < n_j$, or vice versa.

(3) $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c^2$ only if $D(\mathcal{H}_c^1) \supseteq D(\mathcal{H}_c^2)$ (by Proposition 4.3) only if $D(\mathcal{H}_c) = D(\mathcal{H}_c^1) \cup (\mathcal{H}_c^2) = D(\mathcal{H}_c^1)$ only if $\mathcal{H}_c \equiv \mathcal{H}_c^1$ (by Proposition 4.4). □

Proof of Proposition 4.7. Next we show that any equivalence class into which is partitioned the hypothesis subspace $\mathbf{H}_\Phi(P, N)$ by the relation $\equiv$ has a unique minimal classifier. To this end, we prove that, if $\mathcal{H}_c$ and $\hat{\mathcal{H}}_c$ are two minimal classifiers such that $\mathcal{H}_c \equiv \hat{\mathcal{H}}_c$, then $\mathcal{H}_c = \hat{\mathcal{H}}_c$. From which the statement immediately follows. The proof proceeds by induction. (*Basis*) $\mathcal{H}_c$ and $\hat{\mathcal{H}}_c$ are atoms. Trivial. (*Inductive step*) $\mathcal{H}_c = \mathcal{H}_c^1 \vee \mathcal{H}_c^2$ and $\hat{\mathcal{H}}_c = \hat{\mathcal{H}}_c^1 \vee \hat{\mathcal{H}}_c^2$. Note that, from the minimality of $\mathcal{H}_c$ and $\hat{\mathcal{H}}_c$, the minimality of $\mathcal{H}_c^1, \mathcal{H}_c^2, \hat{\mathcal{H}}_c^1$ and $\hat{\mathcal{H}}_c^2$ follows. By Corollary 4.1, $\mathcal{H}_c \equiv \hat{\mathcal{H}}_c$ iff $D(\mathcal{H}_c) = D(\hat{\mathcal{H}}_c)$ iff $D(\mathcal{H}_c) \supseteq D(\hat{\mathcal{H}}_c)$ and $D(\mathcal{H}_c) \subseteq D(\hat{\mathcal{H}}_c)$. By Lemma 4.1, $D(\mathcal{H}_c) \supseteq D(\hat{\mathcal{H}}_c)$ only if $D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^1)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^1)$ and $D(\mathcal{H}_c^1) \supseteq D(\hat{\mathcal{H}}_c^2)$ or $D(\mathcal{H}_c^2) \supseteq D(\hat{\mathcal{H}}_c^2)$. Likewise, $D(\mathcal{H}_c) \subseteq D(\hat{\mathcal{H}}_c)$ only if $D(\mathcal{H}_c^1) \subseteq D(\hat{\mathcal{H}}_c^1)$ or $D(\mathcal{H}_c^1) \subseteq D(\hat{\mathcal{H}}_c^2)$ and $D(\mathcal{H}_c^2) \subseteq D(\hat{\mathcal{H}}_c^1)$ or

$D(\mathcal{H}_c^2) \subseteq D(\hat{\mathcal{H}}_c^2)$. It turns out that, because of the minimality of $\mathcal{H}_c^1, \mathcal{H}_c^2, \hat{\mathcal{H}}_c^1$ and $\hat{\mathcal{H}}_c^2$, either (1) $D(\mathcal{H}_c^1) = D(\hat{\mathcal{H}}_c^1)$ and $D(\mathcal{H}_c^2) = D(\hat{\mathcal{H}}_c^2)$ or (2) $D(\mathcal{H}_c^1) = D(\hat{\mathcal{H}}_c^2)$ and $D(\mathcal{H}_c^2) = D(\hat{\mathcal{H}}_c^1)$, only if (by Corollary 4.1) either (1) $\mathcal{H}_c^1 \equiv \hat{\mathcal{H}}_c^1$ and $\mathcal{H}_c^2 \equiv \hat{\mathcal{H}}_c^2$ or (2) $\mathcal{H}_c^1 \equiv \hat{\mathcal{H}}_c^2$ and $\mathcal{H}_c^2 \equiv \hat{\mathcal{H}}_c^1$. By the inductive hypothesis, $\mathcal{H}_c^i \equiv \hat{\mathcal{H}}_c^j$ only if $\mathcal{H}_c^i = \hat{\mathcal{H}}_c^j$, from which $\mathcal{H}_c = \hat{\mathcal{H}}_c$. □

Proof of Proposition 4.8. The statement we are going to prove is that the poset $(\mathbf{M}_\Phi(P,N), \succcurlyeq_\tau)$, where $\mathbf{M}_\Phi(P,N)$ is the set of the minimal classifiers in $\mathbf{H}_\Phi(P,N)$, is a complete lattice. To this end, let us consider two classifiers $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ in $\mathbf{M}_\Phi(P,N)$. We first show that $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$. Let $\mathcal{H}_c \in \mathbf{M}_\Phi(P,N)$ be a τ-generalization of both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$, i.e., $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}_c^1$ and $\mathcal{H}_c \succcurlyeq_\tau \mathcal{H}_c^2$. Thus, by Proposition 4.3, both $D(\mathcal{H}_c) \supseteq D(\mathcal{H}_c^1)$ and $D(\mathcal{H}_c) \supseteq D(\mathcal{H}_c^2)$ hold. On the other hand, $D(Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)) = D(\mathcal{H}_c^1) \cup D(\mathcal{H}_c^2)$, so that $D(\mathcal{H}_c) \supseteq D(Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2))$. Therefore, by Proposition 4.4, $\mathcal{H}_c \succcurlyeq_\tau Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$, from which the statement $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$ follows.

Now we prove that $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$. Let $\mathcal{H}_c$ be a τ-specialization of both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$, i.e., $\mathcal{H}_c^1 \succcurlyeq_\tau \mathcal{H}_c$ and $\mathcal{H}_c^2 \succcurlyeq_\tau \mathcal{H}_c$. Thus, by Proposition 4.3, both $D(H_c^1) \supseteq D(H_c)$ and $D(\mathcal{H}_c^2) \supseteq D(\mathcal{H}_c)$ hold. On the other hand, $D(Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))) = D(\mathcal{H}_c^1) \cap D(\mathcal{H}_c^2)$ by Proposition 4.6, so that $D(Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))) \supseteq D(\mathcal{H}_c)$. Therefore, by Proposition 4.4, $Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2)) \succcurlyeq_\tau \mathcal{H}_c$, from which the statement $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$ follows. □

Proof of Proposition 4.9. We first prove $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos, Neg, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\rangle$. By Proposition 4.8, $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\mathcal{H}_c^1 \vee \mathcal{H}_c^2)$, that is, $lub_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = Min(\langle Pos, Neg, \mathcal{T}_1 \cup \mathcal{T}_2\rangle)$. It is immediate to recognize that this classifier is $\langle Pos, Neg, \sqcup(\mathcal{T}_1, \mathcal{T}_2)\rangle$ (see Fig. 2), as function $\sqcup(\mathcal{T}_1, \mathcal{T}_2)$ simply minimizes $\mathcal{T}_1 \cup \mathcal{T}_2$ by discarding all thresholds (p_j, n_j) such that there exists (p_i, n_i) such that $\{(p_i, n_i)\} \succcurlyeq_\tau \{(p_j, n_j)\}$ holds (see Proposition 4.5 – Part 1).

Now we show that $glb_\tau(\mathcal{H}_c^1, \mathcal{H}_c^2) = \langle Pos, Neg, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\rangle$. By Proposition 4.8, we have that $glb_\tau = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$. Next we show that $Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2)) = \langle Pos, Neg, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\rangle$. To this end, we observe that lines 9–12 of Fig. 2 are the iterative version of the inductive definition of $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$ (Definition 4.7). Indeed, if both $\mathcal{H}_c^1$ and $\mathcal{H}_c^2$ are atoms, then the function computes $\sqcap(\mathcal{T}_1, \mathcal{T}_2) = \{(Max(p_1, p_2), Min(n_1, n_2))\}$, which coincides with the base step of Definition 4.7 (of course, the classifier $\langle Pos, Neg, \{(Max(p_1, p_2), Min(n_1, n_2))\}\rangle$ is minimal, being an atom). Now, let us consider the general case. It is easy to see that the inductive step of Definition 4.7 generates, for each couple of pairs $(p_1, n_1) \in \mathcal{T}_1$ and $(p_2, n_2) \in \mathcal{T}_2$, a pair $\{(Max(p_1, p_2), Min(n_1, n_2))\}$. And this is exactly what *function* $\sqcap(\mathcal{T}_1, \mathcal{T}_2)$ does at lines 9–12. Thus, after the two nested "for" have been carried out (lines 10–12), the classifier $and(\mathcal{H}_c^1, \mathcal{H}_c^2)$ is generated. However, this classifier may not be minimal (see Example 4.8), so that function *Minimize* is invoked. So, we finally get $\langle Pos, Neg, \sqcap(\mathcal{T}_1, \mathcal{T}_2)\rangle = Min(and(\mathcal{H}_c^1, \mathcal{H}_c^2))$. □

Proof of Lemma 4.2. Given threshold bounds P and N, along with $k \leqslant Min(P+1, N)$, let $T_k^+ = \{p_1, \ldots, p_k\}$ and $T_k^- = \{n_1, \ldots, n_k\}$ be sets of integers, where $\forall i \leqslant k$, $0 \leqslant p_i \leqslant P$ and $0 < n_i \leqslant N$. Next we show that there exists a unique subset $S \subseteq T_k^+ \times T_k^-$ having size k which is a minimal threshold set. Without loss of generality, we assume that $\forall p_i, p_j \in T_k^+$ and $\forall n_i, n_j \in T_k^-$, such that $i < j$, both $p_i < p_j$ and $n_i < n_j$ hold.

Existence. The set $S = \{(p_1, n_1), \ldots, (p_i, n_i), \ldots, (p_k, n_k)\}$ is a subset of $T_k^+ \times T_k^-$ of size k where, for each pair of elements (p_i, n_i) and (p_j, n_j), with $i < j$, both $p_i < p_j$ and $n_i < n_j$ hold. Thus, by Proposition 4.5 – Part 2, S is a minimal threshold set.

Uniqueness. We show that any another subset $S' \neq S$ of $T_k^+ \times T_k^-$, which is a minimal threshold set, has size lower than k. Suppose that $(p_i, n_j) \in S'$ is such that $i < j$ (the case $j < i$ is likewise). Since S' is a minimal threshold set, by Proposition 4.5 – Part 2, for any $(p_s, n_t) \in S'$, either $p_s < p_i$ and $n_t < n_j$ or $p_i < p_s$ and $n_j < n_t$. Now, it is immediate to recognize that, by the above assumption on the ordering of the elements of T_k^+ and T_k^-, the elements $p_s \in T_k^+$ such that $s < i$ (i.e., smaller than p_i) are $i-1$, while the elements $n_t \in T_k^-$ such that $j < t$ (i.e., greater than n_t) are $k-j$. It turns out that (1) there are at most $i-1$ elements $(p_s, n_t) \in S'$ such that $s < i$, and (2) there are at most $k-j$ elements $(p_s, n_t) \in S'$ such that $j < t$. That is, the size of S' is at most $i+k-j < k$. □

Proof of Proposition 4.10. *Part* 1. Every classifier in $\mathbf{M}_\Phi(P,N)$ is of order $r \leqslant Min\{P+1, N\}$. In fact, given $\mathcal{T} = \{(p_1, n_1), \ldots, (p_r, n_r)\}$, from Part 2 of Proposition 4.5 we have that $p_i \neq p_j$ and $n_i \neq n_j$ for all $i, j \in [1, r], i \neq j$. That is, in $\mathcal{T}$ there appear r different positive thresholds and r negative thresholds. Since $0 \leqslant p_i \leqslant P$ and $0 < n_i \leqslant N$, for each $i \in [1, r]$, it turns out that $r \leqslant P+1$ and $r \leqslant N$, i.e., $r \leqslant Min\{P+1, N\}$.

Part 2. Given P, N and $s = Min\{P+1, N\}$, let us consider the sets $T^+ = \{0, 1, \ldots, P\}$ and $T^- = \{1, \ldots, N\}$. T^+ (resp. T^-) is the set of possible values for the positive (resp. negative) thresholds appearing in the classifiers of $\mathbf{M}_\Phi(P,N)$. Now, $\forall r \in [1, s]$, there exist $binom(P+1, r)$ subsets $T_r^+ \subseteq T^+$ (made of r elements from T^+) and $binom(N, r)$ subsets $T_r^- \subseteq T^-$. Also, from Lemma 4.2 we know that, for each pair of sets T_r^+, T_r^-, there is a unique minimal threshold set $S \subset T_r^+ \times T_r^-$ or order r constructible from T^+ and T^-. It turns out that there are $binom(P+1, r) \times binom(N, r)$ minimal threshold sets of order r. Therefore, since $r \leqslant Min\{P+1, N\}$, the total number of threshold sets in $\mathbf{M}_\Phi(P,N)$ is that given by Eq. (1). □

Correctness of the computation of $\uparrow^\phi (\mathcal{H}_c)$. – See Section 5.1. We restrict the proof to the correctness of the computation of $\uparrow^\phi (\mathcal{H}_c)$. The proof concerning $\downarrow^\phi (\mathcal{H}_c)$ follows a similar framework.

Let us start by proving $\uparrow^\phi (\mathcal{H}_c) = \langle Pos', Neg', \mathcal{T}\rangle$, where $Pos' = Pos \cup \{t\}$, $t \in Pos^*(k)$ and $Neg' = Neg$. First of all we note that, since both $Pos' \supseteq Pos$ and $Neg' = Neg$, $\uparrow^\phi (\mathcal{H}_c) \succcurlyeq_\phi \mathcal{H}_c$ holds – see Definition 4.4. Now assume by absurd the existence of $\mathcal{H}_c'' = \langle Pos'', Neg'', \mathcal{T}\rangle$ such that $\uparrow^\phi (\mathcal{H}_c) \succcurlyeq_\tau H_c'' \succcurlyeq_\tau H_c$. Thus, both $Pos' \supseteq Pos'' \supseteq Pos$ and $Neg' \subseteq Neg'' \subseteq Neg$. However, since

Pos′ and *Pos* differ for exactly one term, either $Pos' = Pos''$ or $Pos'' = Pos$ holds. Further, since $Neg = Neg'$, $Neg' = Neg'' = Neg$ holds as well. That is, either $\mathcal{H}''_c = \mathcal{H}_c$ or $\mathcal{H}''_c = \uparrow^{\phi}(\mathcal{H}_c)$, a contradiction.

Now let us prove $\uparrow^{\phi}(\mathcal{H}_c) = \langle Pos', Neg', \mathcal{T}\rangle$, where $Pos' = Pos$, $Neg' = Neg \setminus \{t\}$ and $t \in Neg$. Since both $Pos' = Pos$ and $Neg' \subseteq Neg$, $\uparrow^{\phi}(\mathcal{H}_c) \succeq_{\phi} \mathcal{H}_c$ holds – see Definition 4.4. Now, by absurd, assume that there exists $\mathcal{H}''_c = \langle Pos'', Neg'', \mathcal{T}\rangle$ such that $\uparrow^{\phi}(\mathcal{H}_c) \succeq_{\tau} H''_c \succeq_{\tau} H_c$. Since both $Pos' \supseteq Pos'' \supseteq Pos$ and $Pos = Pos'$, it follows that $Pos'' = Pos$. Moreover, since $Neg' \subseteq Neg'' \subseteq Neg$ and, further, *Neg* and *Neg′* differ for exactly one feature, either $Neg'' = Neg$ or $Neg'' = Neg'$. That is, either $\mathcal{H}''_c = \mathcal{H}_c$ or $\mathcal{H}''_c = \uparrow^{\phi}(\mathcal{H}_c)$, a contradiction.

Proof of Proposition 5.1. Next we show that, given classifiers $\mathcal{H}^1_c$ and $\mathcal{H}^2_c$, the following holds: $\bigvee_x(\mathcal{H}^1_c, \mathcal{H}^2_c) \succeq_x \mathcal{H}^1_c$ and $\mathcal{H}^1_c \succeq_x \bigwedge_x(\mathcal{H}^1_c, \mathcal{H}^2_c)$, where $x \in \{\tau, \phi\}$. By Definition 5.2, we have that $\bigvee_x(\mathcal{H}^1_c, \mathcal{H}^2_c) = lub_x(\mathcal{H}^1_c, \mathcal{H}^{1,2}_c)$. Since $lub_x(\mathcal{H}^1_c, \mathcal{H}^{1,2}_c) \succeq_x \mathcal{H}^1_c$, it turns out that $\bigvee_x(\mathcal{H}^1_c, \mathcal{H}^2_c) \succeq_x \mathcal{H}^1_c$. Dually, from $\bigwedge_x(\mathcal{H}^1_c, \mathcal{H}^2_c) = glb_x(\mathcal{H}^1_c, \mathcal{H}^{1,2}_c)$ and $\mathcal{H}^1_c \succeq_x glb_x(\mathcal{H}^1_c, \mathcal{H}^{1,2}_c)$, $\mathcal{H}^1_c \succeq_x \bigwedge_x(\mathcal{H}^1_c, \mathcal{H}^2_c)$ follows.

Proof of Proposition 6.1. We have to prove that the decision version of the GAMoN learning problem is NP-complete. The proof is by a reduction from the Knapsack problem. Given an atom $\mathcal{H}_c = \langle Pos, Neg, \{(p,n)\}\rangle$, let $S \subseteq T$ be the set of training documents classified by $\mathcal{H}_c$ under c, i.e., $S = \{d \in D \text{ s.t. } |d \cap Pos| \geqslant p \wedge |d \cap Neg| < n\}$. *Precision* is defined as the probability that a document in S is also in the training set T_c of c, i.e.,

$$Pr(\mathcal{H}_c, T) = \frac{|S \cap T_c|}{|S|} \tag{A.1}$$

and *Recall* is defined as the probability that a document in T_c is also in S, i.e.,

$$Re(\mathcal{H}_c, T) = \frac{|S \cap T_c|}{|T_c|}. \tag{A.2}$$

By replacing Eqs. (A.1) and (A.2) into Eq. (2), after some algebra, we get the following formulation of the objective function

$$F(\mathcal{H}_c, T) = \frac{2 \cdot a}{b + |T_c|}$$

where $a = |S \cap T_c|$ and $b = |S \setminus T_c|$. Hence, to maximize $F(\mathcal{H}_c, T)$ we want a to be as large as possible, while keeping b bound to some given value (note that $|T_c|$ is a constant). Thus, the problem of learning an atomic classifier, in its recognition version, can be formulated as follows:

LEARN-ATOM-DECISION (LAD): Given the training set T, the feature space $\langle Pos^*_c(k), Neg^*_c(k)\rangle$ and two positive integers U and V, does there exist a hypothesis $\mathcal{H}_c = \langle Pos, Neg, \{(p,n)\}\rangle$ over $\langle Pos^*_c(k), Neg^*_c(k)\rangle$ such that $a \geqslant U$ and $b \leqslant V$? That is, does there exist a hypothesis which is consistent with at least a positive examples and is not consistent with at most b negative examples?

Now KNAPSACK is the following NP-complete problem: Given $2n+2$ positive integers $w_1, \ldots, w_n$, $v_1, \ldots, v_n$, W and Z, does there exist $X \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in X} w_i \leqslant W$ and $\sum_{i \in X} v_i \geqslant Z$?

We claim KNAPSACK polynomially reduces to LAD. To see this, suppose $I = (w_1, \ldots, w_n, v_1, \ldots, v_n, W, Z)$ is an instance of KNAPSACK. Make the following instance for LAD: (a) $U = Z$ and $V = W$; (b) $\langle Pos^*_c(k), Neg^*_c(k)\rangle = \langle\{t_1, \ldots, t_n\}, \emptyset\rangle$, i.e., the feature space consists of n positive candidate features and no negative candidate feature; (c) the training set T is such that:

(c.1) $\Theta(t_i) \cap \Theta(t_j) = \emptyset$, for each $t_i, t_j \in \{t_1, \ldots, t_n\}$, and
(c.2) $v_i = |\Theta(t_i) \cap T_c|$ and $w_i = |\Theta(t_i) \setminus T_c|$, for each $i \in [1,n]$

where $\Theta(t_i)$ denotes the set of examples (documents) in T where term t_i occurs. From point (c.1) above it follows that each document contains at most one positive candidate term. Further, from points (c.1) and (c.2) it turns out that, for a given $Pos \subseteq \{t_1, \ldots, t_n\}$, the following holds: $a = \sum_{t \in Pos} v_i$ and $b = \sum_{t \in Pos} w_i$. Thus, LAD turns out to be the following problem: "does there exist $\mathcal{H}_c = \langle Pos, \emptyset, \{(1,*)\}\rangle$ such that $\sum_{t \in Pos} v_i \geqslant V$ and $\sum_{t \in Pos} w_i \leqslant C$ (the symbol "*" stands for "immaterial", as $Neg = \emptyset$)?" Or, equivalently: "does there exist $X \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in X} v_i \geqslant V$ and $\sum_{i \in X} w_i \leqslant C$?" Clearly, the answer to this LAD is "yes" iff I is an instance of KNAPSACK, then proving our claim. To conclude the proof it suffices to notice that verifying a YES instance of LAD requires polynomial time. Hence, problem LAD, i.e., the problem of deciding whether there exists an atom satisfying the constraints $a \geqslant U$ and $b \leqslant V$, is NP-complete. It is immediate to realize that the decision version of the learning problem (see Definition 6.1) is NP-complete as well.

Appendix B. The non-deterministic function $\uparrow\mathcal{T}$

Next we give a description of the algorithm of Fig. 4. It creates a direct ancestor $\uparrow\mathcal{T}$ of $\mathcal{T} = \{\tau_1, \ldots, \tau_k\}$, where $\tau_i = (p_i, n_i)$, $1 \leqslant i \leqslant k$, by applying to $\mathcal{T}$ only local changes. We preliminarily recall that, by Proposition 4.5, the minimality of $\mathcal{T}$

requires that $p_i < p_j$, $n_i < n_j$, or vice versa, for each $i, j \in [1, k]$. In the following discussion we assume $p_i < p_j$, $n_i < n_j$ (read "τ_i smaller than τ_j") if $i < j$.

The algorithm starts by checking the condition $\mathcal{T} = \{0, N\}$, that is, if $\mathcal{T}$ is the top element. Clearly, in such a case no direct ancestor exists and the algorithm returns the empty set (line 12). Since the algorithm works on the "distance" between two elements of $\mathcal{T}$, in order not to exclude the first element τ_1 and the last one τ_k, two fictitious elements are defined, namely, $\tau_0 = (-1, 0)$ and $\tau_{k+1} = (P + 1, N + 1)$ (line 2). Then, an element τ_i of $\mathcal{T}$, along with one adjacent (left or right), are randomly selected at lines 14–17. Of course, if $i = 1$, i.e., $\tau_i = (p_i, n_i)$ is the smallest element of $\mathcal{T}$, and $p_i = 0$, then the adjacent of τ_i will be the right one, i.e., τ_{i+1} (line 15). Symmetrically, if $i = k$, i.e., $\tau_k = (p_k, n_k)$ is the greatest element of $\mathcal{T}$, and $n_i = N$ (recall that N is the negative threshold bound), then the adjacent of τ_i will be the left one, i.e., τ_{i-1} (line 16). Then, the function *NewElement* is invoked by passing τ_i and the selected adjacent *adj* (line 18). This function works as follows. First, it orders the element $X = (p_x, n_x)$ and its adjacent $Y = (p_y, n_y)$ in such a way that X is the smallest one (line 1). Then, the distances δ^+ and δ^- between X and Y are computed (line 2). Now, there are two ways for constructing an immediate ancestor of $\mathcal{T}$: either (1) by adding a suitable element τ to $\mathcal{T}$, or (2) by replacing an element τ_i of $\mathcal{T}$ by the most specific τ which generalizes τ_i. Which one of the two alternatives is applied depends on the distances δ^+ and δ^-. In particular, if both distances are greater than one (line 3 – intuitively, this means that there is "enough room" in between X and Y to accommodate a new element in $\mathcal{T}$), then the most specific threshold pair (p, n) such that $p_x < p < p_y$ and $n_x < n < n_y$ is computed, i.e., $p = p_y - 1$ and $n = n_x + 1$ (the most specific (p, n) is the one with the highest possible p value and the lowest possible n value). Then, $\uparrow\mathcal{T}$ is set to *Minimize*$(\mathcal{T} \cup \{(p, n)\})$ (line 18), where *Minimize* is the function sketched in Fig. 2.

As an example, let us consider the classifier $\mathcal{H}_c = \langle Pos, Neg, \mathcal{T}\rangle$, where $\mathcal{T} = \{\tau_1, \tau_2\}$, $\tau_1 = \{(0, 1)\}$ and $\tau_2 = \{(2, 3)\}$, and assume that the threshold bounds are $P = 2$ and $N = 3$. Note that τ_1 is smaller than τ_2. Now, suppose that the algorithm at line 13 selects $i = 2$ (i.e., $\tau_2 = (2, 3)$). Since τ_2 is the greatest element of $\mathcal{T}$ and $n_2 = N$, the algorithm choses the left adjacent, i.e., τ_1 (line 15). Then, the function is invoked (line 18) and the distances $\delta^+ = p_2 - p_1 = 2$ and $\delta^- = n_2 - n_1 = 2$ are computed (line 3). Since ($\delta^+ > 1$ and $\delta^- > 1$) holds (line 3), the function sets $(p, n) = (p_2 - 1, n_1 + 1) = (1, 2)$ (line 4) and returns it to the main. The resulting threshold set is $\uparrow\mathcal{T} =$ *Minimize*$(\mathcal{T} \cup \{(1, 2)\})$, that is, $\{(0, 1), (1, 2), (2, 3)\}$, which is an immediate ancestor of $\mathcal{T}$ (see Fig. 1).

If the condition on the distances at line 3 does not apply, then the most specific threshold pair (p, n) which generalizes either X or Y is computed. Again, this is done depending on the values of two distances δ^+ and δ^-. In particular, if $\delta^+ > 1$, then the algorithm generates the most specific element (p, n) which generalizes Y, i.e., $(p, n) = (p_y - 1, n_y)$. On the contrary, if $\delta^- > 1$, then the algorithm generates the most specific element (p, n) which generalizes X, i.e., $(p, n) = (p_x, n_x + 1)$. Finally, if none of the above conditions hold (line 9), the algorithm generates the most specific element (p, n) which generalizes both X and Y, i.e., $(p, n) = (p_x, n_y)$.

As an example, assume that $\mathcal{T} = \{\tau_1, \tau_2\}$, with $\tau_1 = \{(1, 1)\}$ and $\tau_2 = \{(2, 2)\}$, and let $i = 2$ (i.e., $\tau_2 = (2, 2)$). Suppose that the chosen adjacent is the left one, i.e., τ_1. Since $\delta^+ = \delta^- = 1$, none of the conditions at lines 3, 7 and 8 applies. Thus $(p, n) = (p_1, n_2) = (1, 2)$ is computed at line 9 and returned to the main. This element is then added to $\mathcal{T}$ (line 18) and, after minimization, the algorithm returns $\uparrow\mathcal{T} = \{(1, 2)\}$, which is an immediate ancestor of $\mathcal{T}$ (see Fig. 1).

References

[1] R. Setiono, Extracting *M*-of-*N* rules from trained neural networks, IEEE Trans. Neural Netw. 11 (2001) 512–519.

[2] P.M. Murphy, M.J. Pazzani, Id2-of-3: Constructive induction of *M*-of-*N* concepts for discriminators in decision trees, in: Proc. of the Eighth Int. Workshop on Machine Learning, Evanston, IL, 1991, pp. 183–187.

[3] G.G. Towell, J.W. Shavlik, Extracting refined rules from knowledge-based neural networks, Mach. Learn. 13 (1993) 71–101.

[4] Z. Zheng, Constructing x-of-n attributes for decision tree learning, Mach. Learn. 40 (1) (2000) 35–75.

[5] R. Setiono, S. Pan, M. Hsieh, A. Azcarraga, Automatic knowledge extraction from survey data: learning *M*-of-*N* constructs using a hybrid approach, J. Oper. Res. Soc. (2005) 3–14.

[6] T. Joachims, Learning to Classify Text Using Support Vector Machines, Kluwer, 2002.

[7] O. Larsen, A.A. Freitas, J.C. Nievola, Constructing *X*-of-*N* attributes with a genetic algorithm, in: Proc. of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, 2002, p. 1268.

[8] V.L. Policicchio, A. Pietramala, P. Rullo, A GA-based learning algorithm for inducing *M*-of-*N*-like text classifiers, in: Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops, ICMLA, vol. 1 2011, pp. 269–274.

[9] F. Herrera, Genetic fuzzy systems: Status, critical considerations and future directions, International Journal of Computational Intelligence Research 1 (2005) 59–67.

[10] A. Pietramala, V. Policicchio, P. Rullo, I. Sidhu, A genetic algorithm for text classification rule induction, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases – Part II, ECML PKDD'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 188–203.

[11] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers, San Francisco, CA, 2005.

[12] J. Bacardit, E.K. Burke, N. Krasnogor, Improving the scalability of rule-based evolutionary learning, Memetic Comput. 1 (1) (2009) 55–67.

[13] M. Franco, N. Krasnogor, J. Bacardit, Speeding up the evaluation of evolutionary learning systems using GPGPUs, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO'10, 2010, pp. 1039–1046.

[14] W.W. Cohen, Y. Singer, Context-sensitive learning methods for text categorization, in: ACM Transactions on Information Systems, ACM Press, 1996, pp. 307–315.

[15] J.R. Quinlan, Generating production rules from decision trees, in: Proceedings of the 10th International Joint Conference on Artificial Intelligence, vol. 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987, pp. 304–307.

[16] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Scholkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1998.

[17] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (2002) 1–47.
[18] C. Schaffer, Overfitting avoidance as bias, Mach. Learn. 10 (1993) 153–178.
[19] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of ECML-98, 10th European Conference on Machine Learning, No. 1398, Springer-Verlag, Heidelberg, 1998.
[20] A. McCallum, K. Nigam, A comparison of event models for naive Bayes text classification, in: AAAI-98 Workshop on Learning for Text Categorization, AAAI Press, 1998, pp. 41–48.
[21] J.D. Rennie, L. Shih, J. Teevan, D.R. Karger, Tackling the poor assumptions of naive Bayes text classifiers, in: ICML, 2003, pp. 616–623.
[22] J.R. Quinlan, Learning logical definitions from relations, Mach. Learn. 5 (1990) 239–266.
[23] W. Li, J. Han, J. Pei, CMAR: Accurate and efficient classification based on multiple class-association rules, in: Proceedings of the IEEE International Conference on Data Mining, 2001, pp. 369–376.
[24] X. Yin, J. Han, CPAR: Classification based on predictive association rules, in: Proceedings of the SIAM International Conference on Data Mining, 2003, pp. 331–335.
[25] F. Coenen, P. Leng, The effect of threshold values on association rule based classification accuracy, Data Knowl. Eng. 60 (2007) 345–360.
[26] A. Fernández, S. García, J. Luengo, E. Bernadó-Mansilla, F. Herrera, Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study, Trans. Evol. Comput. 14 (2010) 913–941.
[27] S.W. Wilson, Classifier fitness based on accuracy, Evol. Comput. 3 (1995) 149–175.
[28] G. Venturini, SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts, Mach. Learn. ECML-93 (1993) 280–296.
[29] J. Bacardit, D.E. Goldberg, M.V. Butz, Improving the performance of a Pittsburgh learning classifier system using a default rule, in: Proceedings of the 2003–2005 International Conference on Learning Classifier Systems, IWLCS'03–05, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 291–307.
[30] J.J. Liu, J.T. Kwok, An extended genetic rule induction algorithm, in: Proceedings of the 2000 Congress on Evolutionary Computation (CEC00), 2000, pp. 458–463.
[31] D.R. Carvalho, A.A. Freitas, A hybrid decision tree/genetic algorithm method for data mining, Inform. Sci. 163 (2004) 13–35.
[32] A. Giordana, L. Saitta, F. Zini, Learning disjunctive concept definitions using a genetic algorithm, in: ECAI, 1994, pp. 483–486.
[33] A. Giordana, C. Anglano, A. Giordana, G.L. Bello, L. Saitta, A network genetic algorithm for concept learning, in: Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 1997, pp. 436–443.
[34] F. Divina, M. Keijzer, E. Marchiori, A method for handling numerical attributes in GA-based inductive concept learners, in: GECCO, 2003, pp. 898–908.
[35] J. Bacardit, N. Krasnogor, Performance and efficiency of memetic Pittsburgh learning classifier systems, Evol. Comput. 17 (3) (2009) 307–342.
[36] E. Gabrilovich, S. Markovitch, Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5, in: ICMLí04, 2004, pp. 321–328.
[37] E. Baralis, P. Garza, Associative text categorization exploiting negated words, in: Proceedings of the 2006 ACM Symposium on Applied Computing, 2006, pp. 530–535.
[38] P. Rullo, L. Policicchio, C. Cumbo, S. Iiritano, Olex: effective rule learning for text categorization, IEEE Trans. Knowl. Data Eng. 21 (2009) 1118–1132.
[39] G. Forman, I. Guyon, A. Elisseeff, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (2003) 1289–1305.
[40] A. Tamaddoni-Nezhad, S. Muggleton, A genetic algorithms approach to ILP, in: Proceedings of the 12th International Conference on Inductive Logic Programming, ILP'02, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 285–300.
[41] S.-H. Nienhuys-Cheng, R.d. Wolf, Foundations of Inductive Logic Programming, Springer-Verlag, New York, Secaucus, NJ, USA, 1997.
[42] STOC'84: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 1984, 508840.
[43] L. Pitt, L.G. Valiant, Computational limitations on learning from examples, J. ACM 35 (1988) 965–984.
[44] Ahn, C. Wook, Advances in Evolutionary Algorithms: Theory, Design and Practice (Studies in Computational Intelligence), Springer-Verlag, New York, Secaucus, NJ, USA, 2006.
[45] T. Baick, Optimal mutation rates in genetic search, in: Proc. Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 2–9.
[46] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodalfunction optimization, in: ICGA, 1987, pp. 41–49.
[47] J. Bacardit, Pittsburgh genetics-based machine learning in the data mining era: Representations, generalization, and run-time, Ph.D. thesis, Ramon Llull University, Barcelona, Spain, 2004.
[48] D.P. Greene, S.F. Smith, Competition-based induction of decision models from examples, Mach. Learn. 13 (1993) 229–257.
[49] A.A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, New York, Secaucus, NJ, USA, 2002.
[50] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesús, S. Ventura, J.M. Garell, J. Otera, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[51] F. Debole, F. Sebastiani, An analysis of the relative difficulty of Reuters-21578 subsets, in: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), 2004, pp. 971–974.
[52] W. Hersh, C. Buckley, T. Leone, D. Hickman, Ohsumed: an interactive retrieval evaluation and new large text collection for research, in: W.B. Croft, C.J. Van Rijsbergen (Eds.), Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, Springer-Verlag, Heidelberg/Dublin, 1994, pp. 192–201.
[53] E. hong Han, G. Karypis, Centroid-based document classification: Analysis and experimental results, in: Principles of Data Mining and Knowledge Discovery, 2000, pp. 424–431.
[54] http://www.cs.umass.edu/~mccallum/data.html.
[55] http://www.cs.uic.edu/~liub/FBS/blog-gender-dataset.rar.
[56] http://web.ist.utl.pt/~acardoso/datasets/.
[57] http://www.dmoz.org/rdf.html (content.rdf.u8.gz).
[58] J. Demšar, Statistical comparison of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.
[59] Z. Zheng, R. Srihari, Optimally combining positive and negative features for text categorization, in: Workshop for Learning from Imbalanced Datasets II, Proceedings of the ICML, 2003.
[60] J. Bacardit, M. Stout, J.D. Hirst, K. Sastry, X. Llorà, N. Krasnogor, Automated alphabet reduction method with evolutionary algorithms for protein structure prediction, in: GECCO'07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM Press, 2007, pp. 346–353.
[61] S. Chua, F. Coenen, G. Malcolm, Classification inductive rule learning with negated features, in: Proceedings of the 6th International Conference on Advanced Data Mining and Applications: Part I, Springer-Verlag, 2010, pp. 125–136.
[62] M.A. Franco, N. Krasnogor, J. Bacardit, Analysing BioHEL using challenging boolean functions, Evol. Intell. 5 (2) (2012) 87–102.
[63] D.D. Lewis, Y. Yang, T. Rose, F. Li, RCV1: A new benchmark collection for text categorization research, Journal of Machine Learning Research 5 (2004) 361–397.