Using Lattice-Based Framework as a Tool for Feature Extraction*

Engelbert Mephu Nguifo and Patrick Njiwoua

C.R.I.L, I.U.T de Lens, Université d'Artois,
Rue de l'Université SP-16, 62307 Lens Cedex, FRANCE.

Abstract. Feature transformation (FT) is one of the way to preprocess data in order to improve classification efficiency. Different FT approaches was intensively studied this past years. These approaches wish to give to a learner only those attributes that are relevant to the target concept. This paper presents a process that extracts a set of new numerical features from the original set of boolean features through the use of an empirical mapping function. This mapping is based on the use of an entropical function to learn knowledge over the Galois semi-lattice construction of the initial set of objects. One advantage here is the reduction of effect of possible irrelevant features. This process allows to design an Instance-based Learning system, IGLUE which uses the Mahanalobis measure. A comparison is done with other ML systems, in terms of classification accuracy and running time on some real-world and artificial datasets.

1 Introduction

Learning to classify objects is one of the most studied areas in machine learning (ML). The most well-known ML system is ID3 [Qui86], which is a decision tree-based system. It plays an increasingly important role in ML. Meanwhile important works was done in order to illustrate the efficiency of preprocessing data in the area of statistics, pattern recognition as well as ML. ML systems use data analysis methods towards the search space to outperform both accuracy and complexity of their learning algorithm. These systems generally focus on transformation of original features [AG96],[KS96].

In this paper, we develop a novel constructive method for feature extraction that uses ML techniques to generate new relevant features. Our approach allows to translate initial binary attributes to continuous-valued ones. This translation takes into account only relevant attributes. This feature extraction is achieved for different purposes:

• It is necessary to reduce the effect of irrelevant attributes, by focusing our attention to the way objects are defined in the whole initial context. That is, the

* Financial support for this work has been made available by the Ganymede project of the Contrat de Plan Etat Nord-Pas-de-Calais. The authors thank M. Rouxel for English proofreading and the anonymous reviewers for helpful comments. An extended version of this paper can be found at **http://www.lifl.fr/~mephu-ng**.

definition of an object should be analyzed, not separately, but by considering other examples descriptions.

• In the literature, there are different kinds of discretization methods that allow to transform continuous-valued features into binary or nominal ones. Conversely we propose here a method to achieve the opposite process.

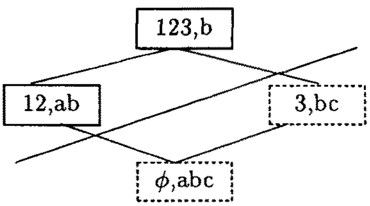
Many Instance-based learners have been developped. They have demonstrated excellent classification accuracy over a large range of domains. Instance-based learners classify an instance by comparing it to a set of pre-classified examples. There is a fundamental assumption that similar instances have similar classifications. The main difficulty is the choice of a distance metric to define *similar instance*, and of a classification function to define *similar classification.* Another difficulty arises from the lack of an integrated theoretical background. To reduce such limitations, IBL systems use a preprocess mechanism on initial data in order to select relevant features. This paper presents a new algorithm for feature extraction that has been integrated in an ML technique to design the IBL system called IGLUE [NMN97]. The feature extraction algorithm consists of two steps (see Fig.1).

First, it builds a join-semi-lattice of the initial context of objects and binary features. To do this, it uses the entropy function to select relevant nodes during the top-down lattice construction. To reduce lattice complexity, only relevant nodes at the top level are taken into account.

Second, all the initial examples are redescribed by the way they are concerned with each relevant node of the semi-lattice. The redescription transforms only relevant initial binary features into new numerical attributes, depending on how examples could appear or not in the relevant nodes. More precisely, we generate a new continuous-valued feature which corresponds to one of the remaining original features. The new feature value for an example is the number of relevant nodes containing both the example and the corresponding binary feature. Consequently, IGLUE applies a 1-nearest neighbor principle to classify unseen examples with the Mahanalobis distance between the redescriptions of unseen examples and training examples.

Experimental comparisons reported in this paper show the effectiveness of IGLUE. Section 2 describes the feature extraction and classification algorithms and Sect. 3 shows experimental comparison on different well-known data sets.

$O\backslash A$	a	b	c	Class
1	1	1		+
2	1	1		+
3		1	1	-



$O\backslash D$	d_1	d_2	Class
1	1	2	+
2	1	2	+
3	0	1	-

Fig. 1. An initial matrix context, the sup-semi Galois lattice constructed by IGLUE and the continuous-valued features obtained after the redescription process.

2 Feature extraction algorithm

This section describes the feature extraction algorithm which consists of two steps. The first step consists in building the join-semi-lattice of the initial context, and selecting the relevant nodes. The second step generates new continuous-valued attributes and redescribes the training set of examples with these new features. Finally the classification process is described.

2.1 Join-semi-lattice algorithm

The algorithm used here to generate the join-semi-lattice is somehow identical to that of LEGAL [MN94]. However their differ by the heuristics used. While LEGAL selects relevant nodes by using two user-specified learning parameters which are respectively the minimum and maximum required numbers of positive and negative instances inside a node, our method uses the entropy function to select relevant nodes at the h-first levels of the join-semi-lattice. This function has one advantage over the empirical process of LEGAL, since it allows to select all significant nodes even if they contain few training instances.

Algorithm(O, A, I, h, λ)

```
Begin
F ⟵ (O, φ), L2 ⟵ (O, φ), P2 ⟵ φ.
While ∃(Ok, Ak) ∈ F
  Remove (Ok, Ak) from F
  Construct Ck, the set of sub-nodes of (Ok, Ak)
  For each node (Oki, Aki) ∈ Ck
    If Level(Oki, Aki) < h Then
      If Entrop(Oki, Aki) ≤ λ Then
        If (Oki, Aki) ∉ L2 Then
          Begin
            P2 ⟵ P2 ⋃ Aki
            Insert(Oki, Aki, L2)
          End
        Else
          add an edge from (Ok, Ak) to (Oki, Aki) in L2
  Endfor
EndWhile
Return(P2, L2)
End
```

Fig. 2. Pseudo-code of Join-semi lattice algorithm

The algorithm (see Fig.2) uses two procedures (*Construct* and *Insert*) to construct all sub-nodes of a given node and, insert new ones in the lattice. The procedure $Level(O_1, A_1)$ restitutes the level of a node inside the lattice. The maximum level, h, of the join-semi-lattice is an input parameter. The procedure $Entrop(O_1, A_1)$ consists in using the entropy function to evaluate the relevance of the node (O_1, A_1). Let $p_1 = |O_1 \bigcap O^+|$ and $n_1 = |O_1 \bigcap O^-|$:

$$Entrop(O_1, A_1) = -\frac{p_1}{p_1+n_1} \cdot log_2(\frac{p_1}{p_1+n_1}) - \frac{n_1}{p_1+n_1} \cdot log_2(\frac{n_1}{p_1+n_1}).$$

A node (O_1, A_1) is *relevant* or *pertinent* if $Entrop(O_1, A_1) \leq \lambda$. Its associate regularity is also pertinent. λ could be an input parameter or settled by the system.

2.2 Feature generation

Here we describe the process to generate new continuous-valued features. We use the set of relevant nodes of the join-semi-lattice for this purpose. Let A^* be the set of attributes of A which appears in at least one relevant node. If an attribute a_i never appears in one of the lattice nodes, then it is not relevant[2]. All training examples are redescribed by the way they interact with the set of relevant nodes. We associate to each attribute a_k of A^*, a new feature, d_k which is defined by the relation J. Let D be the set of those new features, $d_1, \cdots, d_m$, where m is the number of attributes of A^*. Let $P_i \subseteq P$ be the set of relevant regularities which hold for the example o_i.

For an example o_i and a new feature $d_k \in D$, we define a new relation J between O and D, where $J(o_i, d_k)$ is the number of appearances of attribute a_k in all regularities $r \in P_i$, and thus, $0 \leq J(o_i, d_k) \leq |P|$. This relation is then extended to examples of the test set or unseen examples.

Each feature d_k is a quantitative variable and has a correspondence a_k in A^*. The number of new features is less equal to the number of original binary features ($|A^*| \leq |A|$, thus $|D| \leq |A|$). It is important to have a strong dependency between new features and the set of built regularities due to their future use for learning purpose (see Fig.1).

2.3 Classification algorithm

Literature reports extensive studies on nearest neighbor (NN) algorithms for learning from examples. These methods often work as well as other sophisticated ML techniques [Sal91]. Among different similarity measures proposed in the literature, we choose to implement the Mahanalobis distance [QS82] in a 1-NN technique to build the IGLUE system: $s(o_i, o_j) = \sum_{1 \leq k \leq m} \frac{|d_{ik} - d_{jk}|}{\sqrt{d_{ik} + d_{jk}}}$.

• An unseen example o_x is a *positive instance* if its most similar training example is a positive one.

• Otherwise o_x is a *negative instance.*

3 Experimental results

In this section, we present preliminary experiments designed to test our feature extraction algorithm over the new system IGLUE. We report a practical

[2] In Fig.1, as the initial feature c doesn't appear in the built semi-lattice, c is an irrelevant attribute. The new generated features d_1 and d_2 respectively correspond to features a and b.

comparison in terms of complexity and prediction accuracy between LEGAL, IGLUE, C4.5 (unpruned version) and K^*. We assembled some data sets from the classification-learning problems available in the UC/Irvine Repository [MM96]. Time and space performances comparison have been carried out using the same domain that was used for prediction tests. The discretization process applied here, for LEGAL and IGLUE, consists of creating a new attribute for each original feature value. C4.5 and K* are tested with their default parameters values within the Waikato Environment for Knowledge Analysis, WEKA [Zhe96].

For each data set of the Monk's problems, training and testing sets are given. We apply a five cross-validation method on the three other data sets to measure the accuracy of the three systems. In the case of data set with more than two classes, each class was recognized against the others. A batch process has to choose for each system the parameters values which give quite good results on the learning set. These values are then used for test. The different system codes are executed on a Pentium 166 machine with $32M_o$ of RAM.

IGLUE is really faster than LEGAL on all problems, although it is based on a more complex approach that combines different learning strategies. This is a consequence of the level threshold used when building the semi-lattice. When the level threshold increases, the time of IGLUE slightly increases but not exponentially as the theoretical complexity may indicate. This is due to the fact that the lattice construction also depends on the content of the binary table.

Results of classification accuracy obtained on the previous data sets are summarized in Table 1. This table shows that in all the problems except the small soybean, all lattice-based systems could be superior to other systems in terms of prediction accuracy. There is a significant result in the case of the Monks-2 problem. Among the two lattice-based systems, the performance of IGLUE is generally higher than that of LEGAL with a major difference in the two problems: Monks-2 and Breast-Cancer. Since lonely examples are not generally taken into account when LEGAL builds the semi-lattice, including entropy function when generating new features allows to avoid such limitation. Increasing the lattice level leads the system build additional nodes that really encapsulate the behavior of examples descriptions.

When dealing with data where there is a little correlation between examples (this is the case in Monks-2) symbolic ML systems fails to learn. The best results reported in the literature for this problem are that of neural network methods. However varying the level of our semi-lattice (h=3), IGLUE was able by varying the level threshold to obtain significant results that are comparable to neural network techniques. Although taking into account special cases do not guarantee better results, in practice this may be significant as it is the case with the Monks-2 data set.

4 Conclusion

We have presented a new model for generating continuous-valued features from originally binary features. This model is based on the use of entropy function over the Galois lattice framework to induce new features. This framework is the *concisely* largest search space when constructing regularities among examples. This model is a means of preprocessing data before using an appropriated ML system which should deal with continuous-valued features.

For analyzing the efficiency of our model, an IBL system, IGLUE, has been developed. IGLUE works with all the training set of examples which could be huge in practice. Ongoing research is dealing with this problem in order to select pertinent examples in the training set for the classification process. Our intention is to demonstrate the potential of Galois Lattice in Concept Learning.

Table 1. Accuracy results.

Data	Monks1	Monks2	Monks3	Breast C. W. S.	Soybean	Votes
LEGAL	**100**	75.6	**97.3**	79.0	96.9	95.7
IGLUE (h=1)	89.2	78.7	**97.3**	97.7	94.2	97.7
IGLUE (h=2)	**100**	73.4	**97.3**	**98.0**	94.2	**98.6**
IGLUE (h=3)	99.3	**91.0**	96.1	**98.2**	94.2	**98.6**
C4.5	97.4	67.8	93.9	92.1	96.6	90.6
K*	89.7	58.9	85.7	95.1	**100**	92.0

References

[AG96] A. Akkus and H. Gùvenir. K Nearest Neighbor Classification on Feature Projections. In *Proceedings of the Thirteenth International Conference (ICML' 96)*, pages 12–19, Bari, Italy, July 3-6 1996.

[KS96] D. Koller and M. Sahami. Toward Optimal Feature Selection. In *Proceedings of the Thirteenth International Conference (ICML' 96)*, pages 284–292, Bari, Italy, July 3-6 1996.

[MM96] C.J. Merz and P.M. Murphy. Uci repository of machine learning databases, http://www.ics.uci.edu/~mlearn/mlrepository.html, 1996.

[MN94] E. Mephu Nguifo. Galois Lattice: A framework for Concept Learning. Design, Evaluation and Refinement. In *Proc. of the sixth International Conf. on Tools with Artificial Intelligence*, pages 461–467, New Orleans, Louisiana, LA, November 6-9 1994. IEEE Press.

[NMN97] P. Njiwoua and E. Mephu Nguifo. IGLUE: An Instance-based Learning System over Lattice Theory. In *Proceedings of ICTAI'97*, pages 75–76, Newport Beach, California, 3-8 November 1997.

[QS82] J. Quinqueton and J. Sallatin. Expansion and compression on binary data to build feature data. In *Proc. of Intl. Conf. on Pattern Recognition*, 1982.

[Qui86] J. Quinlan. Induction of Decisions Trees. In *Machine Learning*, volume 1, pages 81–106. T.M. Mitchell & al. eds., 1986.

[Sal91] S. Salzberg. Distance metrics for instance-based learning. In *Proc. of 6th Intl. Symp. Methodologies for Intelligent Systems*, pages 399–408, 1991.

[Zhe96] Z. Zheng. Effects of Different Types of New Attributes on Constructive Induction. In *Proceedings of 8th Intl. Conf. on Tools with AI, TAI-96*, pages 254–257, 1996.