Inexact Field Learning Approach for Data Mining

Honghua Dai
Deakin University, Australia

INTRODUCTION

Inexact fielding learning (IFL) (Ciesieski & Dai, 1994; Dai & Ciesieski, 1994a, 1994b, 1995, 2004; Dai & Li, 2001) is a rough-set, theory-based (Pawlak, 1982) machine learning approach that derives inexact rules from fields of each attribute. In contrast to a point-learning algorithm (Quinlan, 1986, 1993), which derives rules by examining individual values of each attribute, a field learning approach (Dai, 1996) derives rules by examining the fields of each attribute. In contrast to exact rule, an inexact rule is a rule with uncertainty. The advantage of the IFL method is the capability to discover high-quality rules from low-quality data, its property of low-quality data tolerant (Dai & Ciesieski, 1994a, 2004), high efficiency in discovery, and high accuracy of the discovered rules.

BACKGROUND

Achieving high prediction accuracy rates is crucial for all learning algorithms, particularly in real applications. In the area of machine learning, a well-recognized problem is that the derived rules can fit the training data very well, but they fail to achieve a high accuracy rate on new unseen cases. This is particularly true when the learning is performed on low-quality databases. Such a problem is referred as the Low Prediction Accuracy (LPA) problem (Dai & Ciesieski, 1994b, 2004; Dai & Li, 2001), which could be caused by several factors. In particular, overfitting low-quality data and being misled by them seem to be the significant problems that can hamper a learning algorithm from achieving high accuracy. Traditional learning methods derive rules by examining individual values of instances (Quinlan, 1986, 1993). To generate classification rules, these methods always try to find cut-off points, such as in well-known decision tree algorithms (Quinlan, 1986, 1993).

What we present here is an approach to derive rough classification rules from large low-quality numerical databases that appear to be able to overcome these two problems. The algorithm works on the fields of continuous numeric variables; that is, the intervals of possible values of each attribute in the training set, rather than on individual point values. The discovered rule is in a form called β–rule and is somewhat analogous to a decision tree found by an induction algorithm. The algorithm is linear in both the number of attributes and the number of instances (Dai & Ciesieski, 1994a, 2004).

The advantage of this inexact field-learning approach is its capability of inducing high-quality classification rules from low-quality data and its high efficiency that makes it an ideal algorithm to discover reliable knowledge from large and very large low-quality databases suitable for data mining, which needs higher discovering capability.

INEXACT FIELD-LEARNING ALGORITHM

Detailed description and the applications of the algorithm can be found from the listed articles (Ciesieski & Dai, 1994a; Dai & Ciesieski, 1994a, 1994b, 1995, 2004; Dai, 1996; Dai & Li, 2001; Dai, 1996). The following is a description of the inexact field-learning algorithm, the Fish_net algorithm:

Input: The input of the algorithm is a training data set with *m* instances and *n* attributes as follows:

Instances	X_1	X_2	...	X_n	Classes
$Instance_1$	a_{11}	a_{12}	...	a_{1n}	γ_1
$Instance_2$	a_{21}	a_{22}	...	a_{2n}	γ_2
...	...	...	...	...	...
$Instance_m$	a_{m1}	a_{m2}	...	a_{mn}	γ_m

(1)

Learning Process:

- **Step 1:** Work Out Fields of each attribute $\{x_i \mid 1 \le i \le n\}$ with respect to each class.

$$h_j^{(k)} = [h_{j_l}^{(k)}, h_{j_u}^{(k)}] \quad (k = 1, 2, ..., s;\ j = 1, 2, ...n). \tag{2}$$

$$h_{j_u}^{(k)} = \max_{a_{ij} \in a_j^{(k)}} \{a_{ij} \mid i = 1,2,...,m\}$$
$$(k = 1,2,...,s;\ j = 1,2,...n) \tag{3}$$

$$h_{j_l}^{(k)} = \min_{a_{ij} \in a_j^{(k)}} \{a_{ij} \mid i = 1,2,...,m\}$$
$$(k = 1,2,...,s;\ j = 1,2,...n). \tag{4}$$

- **Step 2:** Construct Contribution Function based on the fields found in Step 1.

$$\mu_{c_k}(x_j) = \begin{cases} 0 & x_j \in \bigcup_{i \neq k}^{s} h_j^{(i)} - h_j^{(k)} \\ 1 & x_j \in h_j^{(k)} - \bigcup_{i \neq k}^{s} h_j^{(i)} \\ \dfrac{x_j - a}{b - a} & x_j \in h_j^{(k)} \bigcap (\bigcup_{i \neq k}^{s} h_j^{(i)}) \end{cases} \tag{5}$$
$$(k = 1,2,...,s)$$

The formula (5) is given on the assumption that $[a,b] = h_j^{(k)} \bigcap (\bigcup_{i \neq k}^{s} h_j^{(i)})$, and for any small number $\varepsilon > 0, b \pm \varepsilon \in h_j^{(k)}$ and $a + \varepsilon \notin \bigcup_{i \neq k}^{s} h_j^{(i)}$ or $a - \varepsilon \notin \bigcup_{i \neq k}^{s} h_j^{(i)}$. Otherwise, the formula (5) becomes,

$$\mu_{c_k}(x_j) = \begin{cases} 0 & x_j \in \bigcup_{i \neq k}^{s} h_j^{(i)} - h_j^{(k)} \\ 1 & x_j \in h_j^{(k)} - \bigcup_{i \neq k}^{s} h_j^{(i)} \\ \dfrac{x_j - b}{a - b} & x_j \in h_j^{(k)} \bigcap (\bigcup_{i \neq k}^{s} h_j^{(i)}) \end{cases} \tag{6}$$
$$(k = 1,2,...,s)$$

- **Step 3:** Work Out Contribution Fields by applying the constructed contribution functions to the training data set.
  - Calculate the contribution of each instance.

$$\alpha(I_i) = (\sum_{j=1}^{n} \mu(x_{ij}))/n$$
$$(i = 1,2,...,m) \tag{7}$$

  - Work out the contribution field for each class. $h^+ =< h_l^+, h_u^+ >$

$$h_u^{(+)} = \max_{\alpha(I_i), I_i \in +} \{\alpha(I_i) \mid i = 1,2,...,m\} \tag{8}$$
$$h_l^{(+)} = \min_{\alpha(I_i), I_i \in +} \{\alpha(I_i) \mid i = 1,2,...,m\} \tag{9}$$

Similarly we can find $h^- =< h_l^-, h_u^- >$

- **Step 4:** Construct Belief Function using the derived contribution fields.

$$B_r(C) = \begin{cases} -1 & \text{Contribution} \in \text{NegativeRegion} \\ 1 & \text{Contribution} \in \text{PositiveRegion} \\ \dfrac{c - a}{b - a} & \text{Contribution} \in \text{RoughRegion} \end{cases} \tag{10}$$

- **Step 5:** Decide Threshold. It could have 6 different cases to be considered. The simplest case is to take the threshold

$$\alpha = \text{midpoint of } h^+ \text{and } h^- \tag{11}$$

- **Step 6:** Form the Inexact Rule.

$$\text{If} \quad \bar{\alpha}(I) = \frac{1}{N} \sum_{i=1}^{N} \mu(x_i) > \alpha$$
$$\text{Then} \quad \gamma = 1(B_r(c)) \tag{12}$$

This algorithm was tested on three large real observational weather data sets containing both high-quality and low-quality data. The accuracy rates of the forecasts were 86.4%, 78%, and 76.8%. These are significantly better than the accuracy rates achieved by C4.5 (Quinlan, 1986, 1993), feed forward neural networks, discrimination analysis, K-nearest neighbor classifiers, and human weather forecasters. The fish-net algorithm exhibited significantly less overfitting than the other algorithms. The training times were shorter, in some cases by orders of magnitude (Dai & Ciesieski, 1994a, 2004; Dai 1996).

FUTURE TRENDS

The inexact field-learning approach has led to a successful algorithm in a domain where there is a high level of noise. We believe that other algorithms based on fields also can be developed. The β–rules, produced by the current FISH-NET algorithm involve linear combinations of attributes. Non-linear rules may be even more accurate.