From: KDD-95 Proceedings. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

Rough Sets Similarity-Based Learning from Databases

Xiaohua Hu, Nick Cercone
Department of Computer Science, University of Regina
Regina, SK, Canada, S4S 0A2
e-mail: {xiaohua, nick}@cs.uregina.ca

Abstract

Many data mining algorithms developed recently are based on inductive learning methods. Very few are based on similarity-based learning. However, similarity-based learning accrues advantages, such as simple representations for concept descriptions, low incremental learning costs, small storage requirements, etc. We present a similarity-based learning method from databases in the context of rough set theory. Unlike the previous similarity-based learning methods, which only consider the syntactic distance between instances and treat all attributes equally important in the similarity measure, our method can analyse the attribute in the databases by using rough set theory and identify the relevant attributes to the task attributes. We also eliminate superfluous attributes for the task attribute and assign a weight to the relevant attributes according to their significance to the task attributes. Our similarity measure takes into account the semantic information embedded in the databases.

Introduction

Many data mining algorithms developed recently are based on inductive learning methods. Very few are based on similarity-based learning (Lee 1994, Cercone & Tsuchiya 1993). The two approaches differ in their use of the database: inductive learning operates by inferring rules that reflect regularities in the data, whereas similarity-based learning works directly from the database. However, similarity-based learning accrues advantages, such as simple representations for concept descriptions, low incremental learning costs, small storage requirements, producing concept exemplars on demand (Aha & Kibler 1989). Similarity-based learning are represented in machine learning by Aha and Kibler's *instanced-based learning* (Aha & Kibler 1989), Stanfill and Waltz's *Memory-based Learning* (Stanfill & Waltz 1986), and others (Biberman 1994). Most of the similarity-based learning methods use some kind of similarity measure to classify new instances, such as a distance similarity matrix. The shortcomings of these similarity measures are that they only take into account the syntactic distance between instances, the semantic information which is usually more important is not considered. They tend to treat all attributes of the instances as equally important. Such treatment is not generally the case from any real life examples. For example, to classify the gas mileage of cars, the horsepower and weight of the car are very important factors and the number of car doors is trivial. Human beings often tend to concentrate on only a few important attributes when they compare different instances without considering all the differences between all of the attributes. Based on this consideration, we propose a new similarity measure in the context of rough sets (Pawlak 1982). Our methods can identify the relevant attributes for the learning task and eliminate superfluous attributes in the database. Our method can compute the weights of the attributes according to the importance of the attributes to the learning task, automatically.

The paper is organised as follows. The basic features of Similarity-Based Learning (SBL) is outlined in section 2. In Section 3, a new similarity measure using rough sets is introduced and an example is used to illustrate our algorithm. In Section 4 we present our conclusions and our summary.

Outlines of Similarity-Based Learning

Our similarity-based learning algorithm stores a series of training instances in its memory, and uses a similarity metric to compare new instances to those stored. New instances are classified according to those close examples stored in memory. The similarity-based learning algorithm does not generate knowledge rules, decision trees, or other types of abstractions. Instead, instance-based concept descriptions are represented solely by a set of instances. Each instance-based concept is represented by a set of attribute-value pairs (a tuple in the relational database).

More precisely, all similarity-based learning algorithms consist of the following three components (Aha & Kibler 1989):

1: Similarity function: given two normalised instances, this yields their numeric-valued similarity.

2: Classification function: given an instance i to be classified and its similarity with each saved instance, this yields a classification for i.

3: Memory updating algorithm: given the instance being classified and the results of the other two components, this updates the set of saved instances and their classification records.

The heart of the algorithm is the way in which it measures the similarity between two examples. The computation of the similarity measure is one of the main differences among different SBL algorithms. Most SBL algorithms define the similarity of two instances as the negation of their Euclidean distance in the instance space. Each instance is represented by a set of attribute-value pairs, each instance is normalised to ensure that attributes are assigned equal importance by the similarity function, and is viewed as a point in Euclidean n-dimensional space, E^n. Typically the distance is compared attribute by attribute, and then summed. The obvious measure is to give a matched score of 1 if the two individuals have the same value for a given variable, and 0 otherwise. Stanfill and Waltz, in their MBTtalk (Stanfill & Waltz 1986) system proposed two similarity measures: *Weighted Feature Metric (MFM)* based on the precise equality of the values of the predictor fields (attributes) and the *Value Difference Metric (VDM)* by taking into account the similarity of values.

The current similarity measures have the following deficiencies:

(1) The first deficiency of these methods is that they treat attributes equally important. Determining the weights of attributes by human experts are often subjective and/or sometimes it is impossible. We must distinguish important attributes from unimportant ones, moreover, what is important is usually context-sensitive. One can not assign a single value to each attribute and hop it hold for all the time.

(2) These methods are not complete because they do not take into account semantic information into their measures. They also take into account superfluous attributes in the databases to the task attributes. The similarity function is unable to capture the complexity of the problem domains, so it may not perform satisfactory.

Because of these problems, a better similarity measure is needed in learning from databases. In the next section, we propose a new similarity measure using rough set theory.

A Similarity Measure Based on Rough Sets

In many practical applications, during the data collection procedure, it is often difficult to know exactly which features are relevant and/or important for the learning task, and how they should be represented. So all features believed to be useful are collected into the database. Hence databases usually contain some attributes that are undesirable, superfluous, or unimportant to a given learning task. Focussing on a subset of attributes is now common practice. Identifying relevant fields is the most common focussing technique. Rough sets theory proposed in (Pawlak 1982) provides the tools to analyse the attributes globally in the databases.

Basic Notation of Rough Sets

By an **Information System** S (Pawlak 1982), we mean that $S = \{U, A, V\}$, where $U = \{x_1, x_2, ..., x_n\}$ is a finite set of object (n is the number of objects), A is a finite set of attributes, the attributes in A are further classified into disjoint *condition* attributes C and *task* attributes D, $A = C \cup D$, $V = \bigcup_{p \in A} V_p$ and V_p is a *domain* of attribute p. Let $IND \subset A$, x_i, x_j ($0 \leq i, j \leq n$) $\in$ U, we define a binary relation $\widetilde{IND}$, called an *indiscernibility relation*, as $\widetilde{IND} = \{(x_i, x_j) \in U \times U : for\ every\ p \in IND,\ p(x_i) = p(x_j)\}$. We say that x_i and x_j are indiscernible by set of attributes IND in S iff $p(x_i) = p(x_j)$ for every $p \in IND$. $\widetilde{IND}$ is an equivalence relation on U for every $IND \subset A$. An ordered pair $AS = (U, \widetilde{IND})$ is called an *approximation space.* For any element x_i $(0 \leq i \leq n)$ of U, the equivalence class of x_i in relation $\widetilde{IND}$ is represented as $[x_i]_{IND}$. The equivalence class of $\widetilde{IND}$ is called an *elementary set in AS* because it represents the smallest discernible group of objects. Any finite union of elementary sets in AS is called a *definable set in AS*.

Let $X \subset U$, the lower approximation of X in AS is $\underline{IND}X = \{x_i \in U | [x_i]_{IND} \subset X\}$, the upper approximation of X in AS is $\overline{IND}X = \{x_i \in U | [x_i]_{IND} \cap X \neq \emptyset\}$. $\underline{IND}$X is the union of all those elementary sets each of which is contained by X. For any $x_i \in \underline{IND}$X, it is certain that it belongs to X. $\overline{IND}$X is the union of those elementary sets each of which has a non-empty intersection with X. For any $x_i \in \overline{IND}$X, we can only say that x_i is possible to belong to X.

Let $S = \{U, A, V\}$ be an information system, $A = C \cup D$, B $\subset$ C, a positive region B in $\tilde{D}$, $POS_B(D)$, is defined as

$$POS_B(D) = \cup\{\underline{B}X : X \in \tilde{D}\}$$

The positive region $POS_B(D)$ includes all objects in U which can be classified into classes of $\tilde{D}$ without error just based on the classification information in $\tilde{B}$.

We say that the set of attributes D depends in degree k ($0 \leq k \leq 1$) on the subset R of C in S if

$$k(R, D) = card(POS_R(D))/card(U)$$

The value $k(R, D)$ provides a measure of dependency between R and D.

Elimination of Superfluous Attributes

In an information system, we describe each object by the attribute values of C. Very often some of the attributes in C may be redundant in the sense that they do not provide any additional information about the objects in S. Thus it is necessary to eliminate those superfluous attributes to improve learning efficiency and accuracy.

Definition 1. An attribute $p \in B$ is superfluous in B with respect to D if $POS_B(D) = POS_{B-\{p\}}(D)$, otherwise p is indispensable in B with respect to D.

If an attribute is superfluous in the information system, it can be removed from the information system without changing the dependency relationship of the original system. While an indispensable attribute carries essential information about objects in the information system, this attribute should be kept if you do not want to change the dependency relationship of the original system.

Let $B \subset C$ be a nonempty subset of condition attributes, then B is a *dependant* set of C with respect to D if there exists a proper subset $B' \subset B$ such that $POS_{B'}(D) = POS_B(D)$; otherwise B is an independent set with respect to D.

In order to check whether the set, $B \subset C$ is independent or not, it is enough to check for every attribute whether removing of this attribute increases the number of elementary sets or not in the system.

Property 1. If $P \subset C$ is superfluous in C and $\{p\}$ is superfluous in $C - P$, then $P \cup \{p\}$ is superfluous in C.

By this property, we can eliminate superfluous attributes, step by step, from the system.

Example 1. Suppose we have a selected collection of Japanese and America cars in our memory as shown in Table 1 with attributes **Make-model**, # of **door**, **colour** of the car, engine displacement (**displace**), compression ration (**compress**), type of the transmission (**trans**), **weight** of the car, and gas **mileage**, where **make-model**={JAPAN, USA}, **door**={2,4}, **colour**={WHITE, BLACK, RED, GREEN}, **displace**={SMALL, MEDIUM, HIGH}, **compress**={LOW, MEDIUM, HIGH}, **trans**={AUTO, MANUAL}, **Weight**={LIGHT, MEDIUM, HEAVY}, and **Mileage**= {MEDIUM, HIGH}.

The attributes, *door* and *colour* are superfluous. A refined data relation as shown in Table 2 is obtained by removing these two attributes from the relation without losing any essential information. Some tuples which are different in the initial data relation become identical in the refined relation after removing those superfluous attributes. An additional attribute, "vote" is associated with each tuple to indicate the number of tuples in the initial data refined to the current one after eliminating the superfluous attributes. Two tuples are identical if their corresponding attributes values are the same without considering their votes.

Calculating Attribute Weights.

Different attributes may play different roles in determining the dependency relationship between the condition and task attributes. The basic idea for calculating the weights of each attribute is that the more information an attribute provides to the target attributes, the more weight has the attribute. Rough sets theory provides the theoretical background for calculating both attribute weights and the value similarity. Rough sets theory supplies a variety of tools which can measure the amount of information each attribute gives to the other attributes as a form of significance.

The significance of an individual attribute a belonging to the condition attributes C with respect to the dependency between C and D is represented by a significant factor SGF, given by

$$SGF(a) = \frac{k(C, D) - k(C - \{a\}, D)}{k(C, D)}$$

$SGF(a)$ reflects the degree of decrease of significance level between C and D as a result of removing attribute a from C. In practice, the stronger the influence attribute a has on the relationship between C and D, the higher is the value of the $SGF(a)$. For example, in Table 2, if $C = \{Make\_model, displace, compress, trans, weight\}$, $D = \{mileage\}$, then $SGF(Make\_model) = 0.65$, $SGF(displace) = 0.50$, $SGF(compress) = 0.36$, $SGF(weight) = 0.615$. $SGF(trans) = 0.33$. These computational results were obtained from a KDD system DBROUGH (Hu et al. 1994).

Calculating Value-Similarity

Kullback (Kullback 1968) has proposed the following measure for the information content of a value assign-

obj#	Make_model	door	colour	displace	compress	trans	weight	mileage
u1	USA	2	WHITE	HIGH	HIGH	AUTO	HEAVY	LOW
u2	USA	4	RED	MEDIUM	HIGH	MANUAL	HEAVY	LOW
u3	USA	2	GREEN	HIGH	HIGH	AUTO	HEAVY	LOW
u4	USA	4	BLACK	MEDIUM	HIGH	AUTO	MEDIUM	LOW
u5	USA	2	RED	MEDIUM	MEDIUM	MANUAL	MEDIUM	MEDIUM
u6	USA	4	WHITE	HIGH	MEDIUM	MANUAL	MEDIUM	MEDIUM
u7	USA	4	GREEN	MEDIUM	MEDIUM	MANUAL	MEDIUM	MEDIUM
u8	JAPAN	2	BLACK	SMALL	MEDIUM	AUTO	HEAVY	MEDIUM
u9	JAPAN	2	GREEN	MEDIUM	HIGH	MANUAL	LIGHT	MEDIUM
u10	JAPAN	4	WHITE	MEDIUM	LOW	MANUAL	MEDIUM	HIGH
u11	JAPAN	4	RED	SMALL	LOW	MANUAL	MEDIUM	HIGH
u12	JAPAN	4	GREEN	SMALL	LOW	AUTO	MEDIUM	HIGH
u13	JAPAN	2	BLACK	SMALL	MEDIUM	MANUAL	LIGHT	HIGH
u14	USA	2	WHITE	SMALL	MEDIUM	MANUAL	LIGHT	HIGH

Table 1: A Car Relation

obj#	Make_model	displace	compress	trans	weight	mileage	vote
u1'	USA	HIGH	HIGH	AUTO	HEAVY	LOW	2
u2'	USA	MEDIUM	HIGH	MANUAL	HEAVY	LOW	1
u3'	USA	MEDIUM	HIGH	AUTO	MEDIUM	LOW	1
u4'	USA	MEDIUM	MEDIUM	MANUAL	MEDIUM	MEDIUM	2
u5'	USA	HIGH	MEDIUM	MANUAL	MEDIUM	MEDIUM	1
u6'	JAPAN	SMALL	MEDIUM	AUTO	HEAVY	MEDIUM	1
u7'	JAPAN	MEDIUM	HIGH	MANUAL	LIGHT	MEDIUM	1
u8'	JAPAN	MEDIUM	LOW	MANUAL	MEDIUM	HIGH	1
u9'	JAPAN	SMALL	LOW	MANUAL	MEDIUM	HIGH	1
u10'	JAPAN	SMALL	LOW	AUTO	MEDIUM	HIGH	1
u11'	JAPAN	SMALL	MEDIUM	MANUAL	LIGHT	HIGH	1
u12'	USA	SMALL	MEDIUM	MANUAL	LIGHT	HIGH	1

Table 2: A Refined Car Relation

ment

$$K(T|A_k = a) = \sum_t P(t|a)log(\frac{p(t|a)}{p(t)})$$

where a, t represent value assignments of A_k, T, respectively. The formula can be interpreted as the average mutual information between the events t and a with the expectation taken with respect to the a posterior probability distribution of T.

For example, based on table 1 (or 2), $P(mileage = HIGH) = 0.357$, $P(mileage = MEDIUM) = 0.357$, $P(mileage = LOW) = 0.286$, $P(mileage = HIGH|make\_model = JAPAN) = 0.67$, $P(mileage = HIGH|make\_model = USA) = 0.13$, $P(mileage = MEDIUM|make\_model = JAPAN) = 0.33$, $P(mileage = MEDIUM|make\_model = USA) = 0.37$, $P(mileage = LOW|make\_model = JAPAN) = 0.0$, $P(mileage = LOW|make\_model = USA) = 0.5$, then

$K(Mileage|make\_model = USA)$
$= \sum_{t \in mileage} P(t|USA)log(\frac{p(t|USA)}{p(t)}) = P(mileage = LOW|make\_model = USA)log(\frac{P(mileage=LOW|make\_model=USA)}{P(mileage=LOW)}) + P(mileage = MEDIUM|make\_model = USA)log(\frac{P(mileage=MEDIUM|make\_model=USA)}{P(mileage=MEDIUM)}) + P(mileage = HIGH|make\_model = USA)log(\frac{P(mileage=HIGH|make\_model=USA)}{P(mileage=HIGH)})$
$= 0.13 \times log(\frac{0.13}{0.357}) + 0.37 \times log(\frac{0.37}{0.357}) + 0.5 \times log(\frac{0.5}{0.286}) = 0.08$,

In the same way, we can get $K(Mileage|make\_model = JAPAN) = 0.18$

Suppose v_l, v_m are the values of attribute A_k, in our method, the value similarity $ValueSIM(v_l, v_m)$ is defined as the ratio of their absolute value $K(T|A_k = v_l) - K(T|A_k = v_m)$ to the total range of attribute A_k.

$$ValueSIM(v_l, v_m) = 1 - \frac{|K(T|A_k = v_l) - K(T|A_k = v_m)|}{SA_{max} - SA_{min}}$$

where SA_{max}, SA_{min} are the maximum and minimum K values as defined in formula of the attribute A_k.

For example, for the attribute Make-model, $ValueSIM(JAPAN, JAPAN) = 1$, $ValueSIM(JAPAN, USA) = 0$.

A New Similarity Measure

Our similarity measure is designed to take into account all of the knowledge expressed in the databases. Similarity is defined on every attribute type in the refined database after eliminating the superfluous attributes in the original databases, and each attribute is assigned a weight depending on its importance with respect to the target attribute.

Based upon this discussion, we propose our similarity-measure between instances u_i' in the refined database and new instance S, $SIM(u_i, S)$ as

$$SIM(u_i,S) = \sum_k (SGF(A_k) \times ValueSIM(u_{i,k},S_k)) \times V_i$$

where k is the number of attributes in the refined relation, $u_{i,k}$, S_k are the kth attribute value of u_i and S_k respectively. V_i is the vote of tuple i in the refined relation.

An Example

Suppose we have a new car instance S as shown in Table 3 added to our database. In our method, SBL is performed in three steps:

Step 1: Obtain the refined data relation by eliminating the superfluous attributes

Step 2: Calculate the weights of attributes and the value-similarity of each attributes

Step 3: Calculate the similarity measure of the new instance with each instance in the refined data relation and classify the instance to the corresponding category based on the similarity values.

¿From step 1, we know that we do not need to consider the attributes door and colour; then we compute the attributes significance and value-similarity as shown in section 3.3 and 3.4. Finally the similarity measure between S and u_i' is calculated in turn.

Since $SIM(u'_4,S) = (SGF(Make\_model) \times ValueSIM(USA,USA) + SGF(displace) \times ValueSIM(MEDIUM,MEDIUM) + SGF(compress) \times ValueSIM(MEDIUM,HIGH) + SGF(trans) \times ValueSIM(MANUAL,MANUAL) + SGF(weight) \times ValueSIM(MEDIUM,MEDIUM)) \times 2 = (0.65 \times 1 + 0.5 \times 1 + 0.36 \times 0.34 + 0.33 \times 1 + 0.615 \times 1) \times 2 = 4.8$ has the maximum value, so the new car is classified into the same category as u_4', namely the gas mileage of the new car instance is **MEDIUM**.

On the other hand, if we use other conventional similarity measures (e.g., treat each attribute equally important, and for each attribute value, if it matches, score 1 else 0), then the new instance is assigned to the category as instance u_2', since they have the maximal match attribute value 6 with u_2'. Based upon the data in Table 1, we find this is not a reasonable classification. The reason of the misclassification is that the semantic information in the databases is not taken into consideration.

Conclusion

We proposed a new similarity measure for instance-based learning based on rough sets and information theory. Unlike other similarity measure, which did not use the semantic information of the data in the databases, our method can analyse the data dependency relation and use all of the semantic information of the data. Our similarity measure can catch the cause-effect relationship between the attributes and eliminates superfluous attributes in the databases; the weighs of the attributes can be calculated automatically based on the significance value of the attributes and the value-similarity, thus we improve the accuracy of the classification.

The similarity-based learning algorithm proposed in this paper has been implemented as a independent part of a KDD system DBROUGH (Hu et al. 1994). We are currently testing the algorithm on some real large databases and hope to report our results in the future.

Acknowledgements

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council, and the participation of PRECARN Associates Inc.

References

Aha, D, D. Kibler, 1989. Noise-Tolerant Instant-Based Learning Algorithms. *Proceedings of the 11th Inter. Joint. Conf. On AI*, 794-799.

Biberman, Y. 1994. A Context Similarity Measure. *European Conference on Machine Learning*, Catnia, Italy

Cercone, N.; Tsuchiya, M., (eds), 1993 Special Issue on Learning and Discovery in Knowledge-Based Databases, *IEEE transaction on Knowledge and Data Engineering*, Vol. 5(6).

Hu, X, 1994. Object Aggregation and Cluster Identification: A Knowledge Discovery Approach, Applied Math. Letter. 7(4), 29-34.

Hu, X.; Shan, N; Cercone, N.; and Ziarko, W. 1994. DBROUGH: A Rough Set Based Knowledge Discovery System, *Proc. of the 8th International Symposium on Methodologies for Intelligent System*

Hu, X.; Cercone, N. 1994 Learning in Relational Databases: A Rough Set Approach, *Computational Intelligence, An International Journal*, W. Ziarko (ed). special issue on rough set and knowledge discovery (to appear)

Make_model	door	colour	displace	compress	trans	weight	mileage
USA	4	RED	MEDIUM	HIGH	MANUAL	MEDIUM	?

Table 3: A Car Instance

Lee, C., 1994 An Information Theoretic Similarity-based learning Method for Databases, *Proceeding of the 10th IEEE AI on Application Conf.*, 99-105

Kullback, S., 1968, *Information Theory and Statistics,* New York: Dover Publications

Pawlak, Z., 1982. Rough Sets, *International Journal of Information and Computer Science* 11(5), 341-356

Pawlak, Z.; Slowinski, K.; and Slowinski, R. 1986. Rouse Classification of Patients After Highly Selective Vagotomy For Duodenal Ulcer, *Int. J Man-Machine Studies.* Vol 26, 413-433

Stanfill, C.; Waltz, D., 1986. Toward Memory-Based Reasoning, *Communication. of ACM,*, 29:1213-1228

Ziarko, W., 1991. The Discovery, Analysis, and Representation of Data Dependencies in Databases, in *Knowledge Discovery in Databases* G. Piatetsky-Shapiro and W. J. Frawlwy,(eds) Menlo Park, CA: AAAI/MIT, 213-228