Multi-Relational Algebra and its Application to Unrealized Datasets used in C4.5

Ranjan Baghel
Department of Computer Science & Engineering,
National Institute of Technical Teachers Training & Research,
Chandigarh, India

Maitreyee Dutta,Ph.D
Department of Computer Science & Engineering,
National Institute of Technical Teachers Training & Research,
Chandigarh, India

ABSTRACT

Relational databases are based on the theory of relational algebra because all the operations of RDBMS draw their functioning from the operations in relational algebra. The operations of relational algebra are defined on the sets, however, In general, the datamining algorithms requires databases which adopts the multiset philosophy to give better and more accurate results. Unrealized datasets ensures confidentiality of the actual datasets in the datamining process.C4.5 is a classic algorithm which works on mixed real world datasets. This paper proposes the application of Relational algebra for multisets to find the split criterion to be used in classification by the C4.5 algorithm. The results are shown by making the changes in original C4.5 algorithm in the weka tool setting.

General Terms

Multiset, Dataset, Datamining etc.

Keywords

Multirelational algebra, relation, gain ratio, etc.

1. INTRODUCTION

Relation algebra used in RDBMS considers only one copy of the objects in the set. In RDBMS terminology, we can say that a table contains a single unique set of values for the given set of attributes in a given row. The Datamining tasks which uses large size databases requires the multiple occurrences of same set of values for a given set of attributes. The relational algebra operations does not provide the semantics required to handle multiple occurrences of all tuple values of given set of attributes. So, the extended form of relational algebra is a requirement to define the operation on multisets. We are giving a brief description of the operators of multirelational algebra by following [1], and assuming relation instances (multisets) T_1 and T_2 on schema R:

Basic Operators:

Containment ($T_1 \subseteq T_2$): This is similar to subset operator in set theory:
$T_1 \subseteq T_2$ iff $(\forall t)$ $(t \in Dom\ (R) \rightarrow (t \in\in T_1) \leq (t \in\in T_2))$
T_1 is contained in T_2 if it is defined on the same schema, and every tuple that occurs in T_1 k times occurs in T_2 at least k times.
Equality: $T_1 = T2$ is true iff $(\forall t)$ $(t \in Dom(R) \rightarrow (t \in\in T_1) = (t \in\in T_2))$

Strict Containment: It is similar to subset in set theory and $T_1 \subset T_2$ iff $T_1 \subseteq T_2$ and $T_1 \neq T_2$

Union: $T_1 \cup T_2 = \{(x, \max(x \in\in T_1, x \in\in T_2)) | x \in Dom(R)\}$
The union operation gives the smallest multiset that contains both T_1 and T_2. For instance, if $T_1 = [1,1, 2]$ and $T_2 = [1, 2, 2, 3]$, then $T_1 \cup T_2 = [1,1,2,2,3]$.

Concatenation: $T_1 \uplus T_2 = \{(x, x \in\in T_1 + x \in\in T_2) | x \in dom(R)\}$
The concatenation gives new multi-set that has a copy of each of the tuple instances from T_1 and T_2. For instance, if $T_1 = [1, 1, 2]$ and $T_2 = [1, 2, 2, 3]$, then $T_1 \cup T_2 = [1, 1, 1, 2, 2, 2, 3]$.

Difference: $T_1 - T_2 = \{(x, \max(x \in\in T_1 - x \in\in T_2, 0)) | x \in Dom(R)\}$
The multi-set difference operation results in a new multi-set that has a copy of each tuple instance from T_1, but with no tuple instance from T_2. For instance, if $T_1 = [1, 1, 2]$ and $T_2 = [1, 2, 2, 3]$, then $T_1 - T_2 = [1]$.

Selection (σ): Selection operator σ on a multiset T is a means of retrieving only those elements of T that satisfy a predicate. In particular, $\sigma_\phi (T_1)$ selects those tuples in T that satisfy predicate ϕ (which is defined on the attributes in R).The resulting multi-set is also defined on R.
$\sigma_\phi(T) = \{(x, x \in\in T) | x \in dom(R) \wedge \phi(x)\} \cup \{(x, 0) | (x \notin dom(R) \vee (x \in dom(R) \wedge \neg\phi(x))\}$

Size (|T|): The size of a set is the number of elements contained in it. Since a multi-set has duplicates, the size of a multi-set is the number of copies contained within it.
$|T| = \Sigma_{t \in R}\ t \in\in T$

Derived operators

Iterated Concatenation: One can apply a union to the same set multiple times, as follows. Let T be a multi-set on relation schema R The iterated union operator on T is defined as follows:
$T \uplus_1 T = T \uplus T$
$T \uplus_n T = T \uplus_{n-1} T$
It can be shown by induction that $T \uplus_n T = \{(x, n(x \in\in T)) | x \in Dom(R)\}$

Multiplication by a Constant: let k be a constant and T a multi-set. One can call $k(T) = T \uplus_k T$ as the k-multiple of T. Using set pair notation:
$k(T) = T \uplus_k T = \{(x, q(x \in\in T)) | x \in dom(R)\}$
Note that this definition implies that $|k (T)| = k |T|$

The Empty Multi-Set $\oplus$: A multi-set T is empty iff $(\forall x \in dom (R))$ $((x, 0) \in T)$.

2. RELATED WORK

Joseph Albert [1] has given the description of properties of bag data types where bag means for the multiset. In [2], authors have proposed a complete extended relational algebra with multi-set semantics which is closely connected to the standard relational algebra and includes constructs that extend the relational algebra to a complete sequential database manipulation language that can either be used as a formal background to other multi-set languages like SQL, or as a database manipulation language on its own as in the PRISMA/DB database project [3], where a variant of the language has been used as the primary database language.C4.5 is classic datamining algorithm proposed by Quinlan[7],[8].The unrealization approach, proposed by P.K. Fong, et al in [4] produces two different but related datasets (Multisets) from the actual input datasets.

3. PROPOSED WORK

This paper shows the application of multirelational algebra to find the split criterion and other parameters used in C4.5 algorithm [7], using the unreal and perturbed datasets but not using actual datasets. The Used Lemma and derivations are simplified form of the derivations given in [6]. We have simulated the performance of the given expressions to assess their utility in calculating the required measure by C4.5 algorithm in weka environment.

3.1 Entropy calculations on unreal datasets

The entropy calculations is a major task in the decision making process as the entropy values are essential in determining the split criterion value, e.g. information gain or gain ratio or gini index, etc. The following lemma shows how one can find the same entropy values by using unreal and perturbed datasets and not using actual datasets.

Lemma: Let U be the universal instance space, q be a positive integer, T be a training set with $T \subseteq q(U)$, and T′, T_P be unreal data sets generated by the unrealization procedure. Then H(y, T) can be calculated by using information about U, T′ and T_P alone.

Proof:

The known standard formula to calculate entropy with respect to a given class is given as below [4]:

$$H(y, T) = \sum_{v \in dom\,(y)} \frac{|\sigma_{y=v}(T)|}{|T|} \log_2 \frac{|\sigma_{y=v}(T)|}{|T|} \quad \text{(eq. 3.1)}$$

$$\frac{|\sigma_{y=v}(T)|}{|T|} = \frac{|\sigma_{y=v}(q(U)-(T' \uplus T_P))|}{|q(U)-(T' \uplus T_P)|}$$ since $T=q(U)-(T' \cup T_P)$ as proved in [5]

$$= \frac{|\sigma_{y=v}(q(U))|-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)-(T' \uplus T_P)|}$$ because Multisets satisfies the relationship $|\sigma_\phi (A - B)| = |\sigma_\phi (A)| - |\sigma_\phi (B)|$ based on the fact $|A - B| = |A| - |B|$.

$$= \frac{|\sigma_{y=v}(q(U))|-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$$

$$= \frac{|\sigma_{y=v}(q(\sigma_{y=v}(U))|-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$$ since q(U) is the set containing q copies of each object of U.

$$= \frac{q|\sigma_{y=v}(U)|-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$$

$$= \frac{q\frac{\prod_{i=1}^{n}|dom\ (a_i)|}{|dom\ (y)|}-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$$ since U is defined as the universal relation for the given attributes.

$$= \frac{q\frac{|U|}{|dom\ (y)|}-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$$

So in the end, It can be shown that H(y, T) $= \sum x \log_2 x$

Where $x = \frac{q\frac{|U|}{|dom\ (y)|}-|\sigma_{y=v}(T' \uplus T_P)|}{|q(U)|-|(T' \uplus T_P)|}$

Since this expression does not have T, thus it proves the claim.

3.2 Intermediate calculations for entropy values on random unreal subsets

Since it is known that the C4.5 algorithm recurs on smaller bags after each successive split. So, entropy calculations on these subsets of unreal datasets are a requirement.

Predicate ψ_r: ψ_r is a predicate which is true on $A' \subseteq A$, with each attribute being assigned some value in its domain: $\psi_r = \psi_1 \wedge \psi_2 \ldots . \psi_i$ where ψ_i is component predicate of compound predicate ψ_r.

Calculation of $H(y, \sigma_\psi (T))$:

For any value of ψ_r, one can calculate $H(y, \sigma_\psi (T))$ as below by substituting T by $\sigma_\psi(T)$ in eq.3.1.

$$H(y, \sigma_\psi (T)) = \sum_{v \in dom\,(y)} \frac{|\sigma_{y=v}(\sigma_\psi (T)|}{|\sigma_\psi (T)|} \log_2 \frac{|\sigma_{y=v}(\sigma_\psi (T)|}{|\sigma_\psi (T)|}$$

One can drive the expression for $H(y, \sigma_\psi (T))$ which is free from T, as is done for H(y, T). This is a requirement in intermediate calculation to find the split criterion.

4. CALCULATION OF SPLIT CRITERION

As shown in section 3, one can find the entropy of a bag with respect to the each class value and the entropy of the subsets selected by the random predicate value. Similarly, the entropy calculations with respect to a specific attributes can also be done. So, by making these adjustments in algorithmic settings the classification can be done by using the unreal and perturbed datasets produced by fong's [5] approach. The split criterion used by C4.5 can be calculated as follows:

Gain Ratio $(a_i, T) = \frac{\text{Information Gain } (a_i,T)}{H(a_i,T)}$ where Information Gain $(a_i, T) = H(y, T) - H(y, T|a_i)$, so the expression for gain ratio can be given as:

Gain Ratio $(a_i, T) = \frac{(H(y,T) - H(y,T|a_i))}{H(a_i,T)}$, the values of H(y, T), H(y, $T|a_i$) and H (a_i, T) can be calculated over unreal datasets as stated in section 3.

5. SIMULATION RESULTS IN WEKA

Weka is datamining tool which uses the java based classes to perform datamining tasks. The modified versions of some of these classes are used to simulate the results. The classes which are modified namely EntropyBasedSplitCriteria.java, GainRatioSplitCriteria.java to adjust for the changes outlined in section 3 and section 4.The information about weka tool is collected from various enumerable sources but the main

source is [9]. The Unreal datasets are produced by our java implemented version of Unrealization process of [5]. The trivial decision value is changed from most frequent value to least frequent value. The Datasets used in the simulation are not given here to save on space; however the resulted decision tree traces are shown below:

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: Query Result
Instances: 398
Attributes: 5
outlook
temperature
humidity
windy
play
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

```
------------------
outlook = rainy
|   windy = TRUE: YES (65.0/26.0)
|windy = FALSE: No (63.0/30.0)
|windy=TRUE
: YES (0.0)
|windy=FALSE
: YES (0.0)
outlook = sunny
|   windy = TRUE: YES (74.0/33.0)
|   windy = FALSE
|   |   humidity <= 85
|   |   |   humidity <= 70: YES (14.0/5.0)
|   |   |   humidity > 70
|   |   |   |   humidity <= 75: No (8.0/2.0)
|   |   |   |   humidity > 75
|   |   |   |   |   humidity <= 80: YES (3.0/1.0)
|   |   |   |   |   humidity > 80
|   |   |   |   |   |   temperature <= 75: No (6.0/2.0)
|   |   |   |   |   |   temperature > 75: YES (2.0)
|   |   humidity > 85
|   |   |   humidity <= 90: No (16.0/6.0)
|   |   |   humidity > 90
|   |   |   |   humidity <= 95
|   |   |   |   |   temperature <= 71: No (6.0/1.0)
|   |   |   |   |   temperature > 71: YES (7.0/1.0)
|   |   |   |   humidity > 95
|   |   |   |   |   temperature <= 75: YES (3.0/1.0)
|   |   |   |   |   temperature > 75: No (3.0/1.0)
|windy=TRUE
: YES (0.0)
|windy=FALSE
: YES (0.0)
outlook = overcast
|   windy = TRUE: YES (51.0/23.0)
|   windy = FALSE
|   |   humidity <= 95: YES (60.0/21.0)
|   |   humidity > 95: No (12.0/3.0)
|windy=TRUE
: YES (0.0)
|windy=FALSE
: YES (0.0)
outlook = overcast
|humidity<=65:YES
(3.0/1.0)
|humidity>65:No
(2.0)
Number of Leaves:  24
```

Size of the tree: 39

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	176	44.2211 %
Incorrectly Classified Instances	222	55.7789 %
Total Number of Instances	398	

=== Detailed Accuracy by Class ===

TPR	FPR	Precision	Recall	F-Measure	ROC Area	Class
0.7	0.838	0.49	0.7	0.576	0.395	YES
0.15	0.294	0.297	0.15	0.199	0.383	No
0	0.005	0	0	0	0.869	YES
0	0.003	0	0	0	0.697	No

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
0	0	149	64	0	0	| c = YES
0	0	153	27	0	0	| d = No
0	0	1	0	0	1	| e = YES
0	0	1	0	2	0	| f = No

Performance of Classifier on Supplied Test Set:

=== Evaluation on test set ===

=== Summary ===

Correctly Classified Instances	6	60 %
Incorrectly Classified Instances	4	40 %
Total Number of Instances	10	

=== Detailed Accuracy By Class ===

TPR	FPR	Precision	Recall	F-Measure	ROCArea	Class
0.75	0.167	0.75	0.75	0.75	0.917	YES
1	0.429	0.5	1	0.667	0.81	No

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
0	0	1	2	0	0	| a = no
0	0	0	0	0	0	| b = yes
0	0	3	1	0	0	| c = YES
0	0	0	3	0	0	| d = No

6. CONCLUSION

The Multirelational algebra is not universalized so far as the standards are not developed to the fullest, however the operations have been included here are proven and have contributed to the same or even better results in the current context. The decision tree trace shown in the results has almost same decision rules as in the decision tree generated by the original datasets. So, these results protect the original datasets and produce the same decision tree. So, this paper exemplifies the power of Multirelational algebra in the C4.5 classification process with the given results. The power of multirelational algebra is yet to be reflected in diverse application and can be considered as a potential future research direction.

7. REFERENCES

[1] Joseph Albert. 1991 "Algebraic properties of bag data types", In VLDB '91: Proceedings of the 17th International Conference on Very Large Data Bases, pages 211–219, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[2] Grefen, P.W.P.J.; de By, R.A. 1994. Data Engineering, Proceedings.10th International Conference A multi-set

extended relational algebra: a formal approach to a practical issue

[3] Apers, P.M.G. ; van den Berg, C.A. ; Flokstra, J. ;Grefen, P.W.P.J. ; Kersten,M.L. ; Wilschut,A.N.1992. PRISMA/ DB: a parallel, main memory relational DBMS Knowledge and Data Engineering, IEEE Transactions on Volume: 4 , Issue: 6.

[4] Han, J.; Kamber, M.2006. Data Mining: Concepts and Techniques, 2nd edition, Morgan Kaufmann Publishers.

[5] Fong, P.K.; and Jens H. Weber-Jahnke, Feb 2012. "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets", IEEE Transactions on Knowledge and Data engineering, vol. 24, no. 2, page no. 353

[6] Williams, J. 2010. Unrealization Approaches for Privacy Preserving Data Mining, A Thesis submitted in Department of Computer Science, University of Victoria.

[7] Quinlan, J. R., 1993." C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers.

[8] Quinlan, J. R, 1986. Induction of Decision Trees. Machine Learning, 1, 1, 81-106.

[9] Weka Primer: URL:http://weka.wikispaces.com/Primer.