4710 project report

1.Introduction

As the development of internet technologies, it has been observed that collected data is growing in terms of volume, variety, velocity, veracity, and value. This leads to the development of various new data analysis technique. Among these technique, incremental data mining is often used to discover previous unknow and potential useful knowledge from existing and growing information. Decision making based on extracted rule from growing data is one of such important application in incremental data mining. In real life, data usually with high dimension of attributes, it is more practical to select representative features before making decision. In our project, we are focusing on incremental feature selection and make decision based on selected features.

Pawlak’s Rough[cite] set model is one of such theoretical frameworks in feature selection as well as decision making based of selected feature. Feature selection in rough set is also called attribute reduction[cite]. The selected feature is called reduct[cite]. Attribute reduction is a process to reduce the number of attribute and preserve the discernibility of the original data while minimize the class separability[cite]. In the recent decades, several techniques in attribute reduction based on rough set were proposed[cite], but most of them is focus on static data. In the environment of growing data, they lack the ability to update the selected features and must recompute them every time, which is not scalable and efficient in the century of big data. To deal with dynamic data stream, there exist some research on finding reduct incrementally based on rough set[cite]. However, most of them are focusing on update the reduct regarding to the whole database without considering concept drifting. Furthermore, it has been overserved that rough set model is very sensitive to outliers[cite], most of the work listed above also does dealing outlier properly and maybe need more preprocess work before running their techniques.

In terms of decision making, Pawlak’s Rough set also have its drawbacks when dealing with real value attributes, which are usually the dominant attributes in real life. One must implement discretization or categorization before analyzing through rough set. Even with the help of discretization or categorization, traditional rough set theory tends to treat discretized real value as nominal data and ignore some important intrarelationship, such as similarity among them. Fuzzy rough set[cite] is a variant of rough set which treat the real value as a member in fuzzy set to preserve the relationship among real value. Some of our models will also adopt the concept of membership function in fuzzy rough set.

For convenience for the future discussion, here is the description of our assumption, contributions and main idea. This project assumes that the decision attribute in the decision table(dataset) is nominal (crisp), while the conditional attribute could be nominal or numeric. In this project we proposed 3 incremental feature selection and decision rule induction models for decision-making based on crisp rough set (discretize real value data before analyzing) and 3 corresponding feature selection for decision-making models based on our modified rough set model (using membership function in fuzzy rough set concept and using similarity to comparing real value attribute). The 3 proposed incremental crisp rough set models named: incremental voting rough set (IVRS), incremental sliding window rough set (ISwRS), and time fading rough set (TFRS). The 3 proposed modified rough set model called: incremental voting membership rough set (IVMRS), incremental sliding window membership rough set (ISwMRS), and time fading membership rough set (TFMRS). In fact, the 3 crisp models and 3 membership models are correspondingly built on the same basic concept with different way to handle real value attributes and different techniques in rule mining. Inspired by association rule mining in stream data[cite], our models, unlike most of works, separate the growing dataset into different batches, and preform attribute reduction within each batch then make decision based on the rule mined in each batch. We perform attribute reduction by implementing the technique of discernibility matrix [cite] and make decision based on the rules induced by LEM2 algorithm[cite] in crisp rough set and based on ruling objects in membership rough set. These proposed models are robust to outliers since outliers would only affect the process of attribute reduction in its own batch, and outliers are usually rare in comparison to normal data. 4 of our proposed models, ISwRS, TFRS, ISwMRS, TFMRS, could also handling the problem of concept drifting by time, which is a popular phenomenon in real life, by putting more attention in the recent data. The IVRS and IVRMS, on the other hand, could also be used in static data to make outlier bearable predictions. And the 6 proposed models also tend to be more efficient than other works in the process of attribute reduction since only the reduct in the latest batch will be updated.

The structure of this report is as follow: section 2 will list some relate work, section 3 will introduce the background of rough set. section 4 will be the detail of our 3 crisp models. section 5 will introduce backgrounds in fuzzy rough set. Section 6 will contain the detail of our 3 membership-based models. Section 7 will be experiment and comparison to other existing models. Section 8 will conclude our paper and outline futures works.

2.related work

Features selection refers to selecting representative attributes in high dimension data set. In the field of rough set, it is called attribute reduction. There exist many works in finding the reduct from the decision table or information system. Skowron and Rauszer [cited] used discernibility matrix and discernibility function to find reduct in rough set system. Qian and Shen[cite] used information entropy and heuristic function to calculate the reduct. Qian and Shen[cite] used distance measure to find reduct in variable precision rough set system. These works present different techniques in attribute reduction. However, their method could only work well on static information system. In the environment of growing data, these techniques need to run entirely whenever new data come in, which are not scalable and practical.

Incremental attribute reduction refers to dynamically updating reduct in the rough set system. In the field of decision rough set, new set of decision rules could be induced from the updated reduct. There are many research works on incremental attribute reduction: Qian et al in [cite] developed incremental attribute reduction based on information entropy; Hu et al in [cite] dynamically update reduct based on positive and negative region regard to decision attribute; Zheng and Wang[cite] introduce (RRIA), a tree based incremental features selection method in rough set system; Yang et al[cite] using discernibility matrix to incrementally update reduct in fuzzy rough set. Rather than updating reduct, there are also some works[cite] directly update decision rules. However, all these works only focused on updating selected features incrementally and neglected the fact that early data could lose the ability to represent the trend in recent time and fail to capture the phenomenon of concept drift. Since these early data still present in the system and get considered when updating new reduct, they could affect accuracy of the induced decision rules when the concept of the data is drifting. On the other hand, since it has been observed that rough set system is sensitive to outliers[cited], the existence of outlier could significantly affect the result of attribute reduction, thus induction of decision rules. All works above did not treat the outliers properly and require outlier detection as a preprocessing job. Our proposed model could handle the situation of concept drift well since data will be divided by batches based on arrival and could reduce the effect of outliers by using a voting mechanism as outlier could only affect the rules, thus the vote, in its batch.

Beside the works above, the most related works to our project is Leung’s 3 models for association rule mining in data stream: landmark stream mining model[cite], sliding window stream mining model[cite] and time fading stream mining model[cite]. The proposed models are the mixture of Leung’s stream mining models and the rough set system.

3.preliminaries I

In this section, we briefly review some preliminaries in crisp rough set theory[cite] that are related to our project, including lower and upper approximation, reduct, core.

3.1 rough set theory

In this project, we focus on the problem in the decision making based on rough set. Let DT = (U, A∪D) be a decision table with nominal conditional attribute A, decision value set D = {d}, and universe of objects U.

Definition 1: An equivalence relation defined by attribute subset B ⊆ A is call B-indiscernibility relation denoted by IND(B) is defined by:

IND(B) = {(x, y) ∈ U × U: a(x) = a(y), ∀ a ∈ B}

Definition 2: for any object x ∈ U, [x]B = {y ∈ U: a(x) = a(y), ∀ a ∈ B}

Definition 3: given an attribute subset B ⊆ A, the lower approximation of B, B↓, and upper approximation of B, B↑, regarding to concept set X is define as:

B↓(X) = {x: [x]B ∈ X}, those x such that all the elements in [x]B is in X

B↑ (X) = {x: [x]B ∩ X ≠ Ø}, those x such that at least one element in [x]B is in X

Definition 4: Suppose U = {x1, x2, x3..., xn}, a n × n matrix M (), is defined as a discernibility matrix of DT = (U, A∪D) if:

and

Definition 5: The core in decision table is defined as:

Definition 6: The discernibility of attribute a is defined as: , and the discernibility of A is:

Definition 7: The reduct Red of a decision table is defined as:

4. our 3 crisp models

This section will cover the detail of the 3 proposed crisp models. In subsection 4.1 will cover the algorithm we used for incremental attribute reduction, finding approximate and decision rules induction, and subsection 4.2 will introduce the detail of our three models.

4.1 Incremental Attribute reduction

We use discernibility matrix and discernibility relation as the framework and combining with heuristic function to finding one reduct in the rough set system.

Algorithm 1: Attribute reduction by discernibility matrix and hill climbing:

Compute Dis(Red+{a}) for a in A – Red

Select a0 such that Dis(Red + {a0}) is maximum

Let Red = Red + {a0}, and Dis(Red) = Dis(Red + {a0})

Base on the discernibility matrix, this algorithm uses heuristic method to find one reduct from the decision table. Step 1 construct a discernibility matrix by using definition 4.

Algorithm 2: Incremental discernibility and reduct update when new samples come in.

Input: A decision table DT = (U, A or D), Red

Dis(a) for a in A, Dis(A), Dis(red)

new observations O

Output: Dis(a) for a in A, Dis(A), Dis(red), DT = (U+x, A or D)

Step 1: For x in U, o in O and d(x) != d(o), add (x,o), (o,x) to Dis(a) and Dis(A) when a(x) != a(u); if such a in Red, also add (x,o),(o,x) to Dis(Red)

Step 2: For o, p in O such that d(o) != d(p), add (p,o), (o,p) to Dis(a) and Dis(A) when a(x) != a(u); if such a in Red, also add (p,o),(o,p) to Dis(Red)

Step 3: if Dis(Red) == Dis(A): to step 5, else to step 4

Step 4: while Dis(Red) != Dis(A):

Compute Dis(Red+{a}) for a in A – Red

Select a0 such that Dis(Red + {a0}) is maximum

Let Red = Red + {a0}, and Dis(Red) = Dis(Red + {a0})

Step 5: while Dis(Red) == Dis(A):

For a in Red, compute Dis(Red – {a})

If there is a0 such that Dis(Red – {a0}) == Dis(A)

Red = Red – {a0}

else: break

This algorithm firstly updates the discernibility of all the attribute and Red by adding new pairs that contain the object in the observations. Then it updates the reduct by the updated discernibility. Note the step 3 is the same as the one used in initial attribute reduction, using a heuristic method to choosing new candidate. Step 4 is also necessary for pruning redundant reduct since we are using a heuristic method and choosing one reduct in the initial data, there will be case that redundant attributes present in the new reduct.

Algorithm 3 finding lower approximate and upper approximate base on reduct

Algorithm 4. Lem2

4.2 Proposed models

In this subsection, we will provide the detail description of our three crisp models. Our three models work like a batches container, each batch is an individual rough set system by itself. We then apply attribute reduction in each batch, find the lower and upper approximate of decision class in each batch and use LEM2 algorithm to find the rule in each batch. When new samples arrive, we add these new observations in the last batch and update the reduct, approximate and decision rule in the last batch. The difference among the three models is presented in how much data they keep, and the mechanism used to predict or classify object. Since among the three crisp models, the IVRS is the most general one, we will introduce it first.

IVRS

The incremental vote rough set works like a container contain several rough set systems. Here we will describe how the model fit initial train data, update new train data and make prediction base on trained data.

Algorithm 5. Fit existence data in IVRS

Input: initial data U, preassigned maximum batch size N

Output: an IVRS model

Step 0: if U is not sorted by time, sort it by time

Step 1: calculate the child number |C| = |U | / N + 1

Step 2: Let Child\_list = {}, For every N objects in the initial data, create a child decision table, Child, totally will be |C| such Child. Add each Child in Child\_list

Step 3: For Child in Child\_List, use Algorithm 1 to find the reduct

Step 4: For Child in Child\_List, use Algorithm 3 to find approximate based on reduct in Child

Step 5: For Child in Child\_List, use Algorithm 4 to induce decision rules.

Reader should keep in mind that the last Child in Child\_List could be partially filled. And for each Child in the Child\_List, it has its own reduct and own decision rules. But the IVRS model itself does not any reduct.

Algorithm 6. Update new observations into IVRS

Input: current IVRS model, new observations Unew, maximum batch size N

Output: an updated IVRS model

Step 1: Let Childlast be the last Child in the Child\_List

Step 2: select the first min((N - |Childlast|), |Unew|)objects in Unew as Uadd.

Step 3: update Uadd into Childlast by algorithm 2, then apply algorithm 3 and 4 to Childlast.

Step 4: Let Unew = Unew - Uadd

Step 4: if |Unew| > 0, create a Child decision table for every N objects in Unew, apply algorithm 1, 3, 4 to each new created Child and add them to Child\_List by order

To sum up this process, we first fill the last decision table in the child list then create new decision tables for the rest object in Unew like what we do in algorithm 5.

To make prediction or classification, we have two choice: the first one is matching the predicted object with the decision rules in each batch, then the decision get most voted will be the final prediction. Another one is firstly combining all the rules in each batch with their support and confidence as totally decision rules, then make prediction on there totally rules. [Which one we choose?\_\_\_\_group member choose please.]

Algorithm 7. Prediction in IVRS

Variant of IVRS ------ ISwRS

The proposed ISwRS using the same mechanism to predict and classify object as IVRS, while it just keeps a fixed number of latest batches in the models, and the early batch will be deleted from the model. Since it is merely a modification of the IVRS, here will just a brief description of where it modifies.

Modification 1. Fit existence data in ISwRS

Input: initial data U, preassigned maximum batch size N, preassigned window size W

Output: an ISwRS model

Step 1: using algorithm 5 to initialize the model

Step 2: if |Child\_List| > W: remove the Child at the front until |Child\_List| = W

Modification 2. Update new observation in ISwRS

Input: current ISwRS model, new observations Unew, maximum batch size N, preassigned window size W

Output: an updated ISwRS model

Step 1: using algorithm 6 to update the current model

Step 2: if |Child\_List| > W: remove the Child at the front until |Child\_List| = W

This model allows the user to keep focus on the recent data and make decision based on the recent rules, and it is also more efficient in terms of space complexity than IVRS. Keeping focus on recent data tend to be more practical in the real life, especially in the case of concept drift.

Variant of IVRS--------TFRS

Even though IVRS and TFRS could handle time series data by dividing the data into different batch, in the process of prediction, they will give equal weight to decision rules in each batch. Concerning to the fact that recent data are more representative and useful in making prediction, we also proposed a variant of the IVRS by adding a time fading factor in decision rules.

[content base on the concept we decide to make prediction in IVRS]

5. preliminaries II

Additional to crisp rough set, in this project we also proposed 3 membership rough set models based on the concept of membership function in fuzzy rough set and similarity measures between real values. This section will give some reviews about relevant definition in fuzzy rough set, fuzzy discernibility matrix and similarity measure.

5.1 fuzzy rough set

With the same definition of decision table, is called a fuzzy decision table. For each condition attribute a ∈ A, one could define a fuzzy binary relation , which is called a fuzzy binary relation if is reflexive, (R(x, x) = 1), symmetric, (R(x, y) = R(y, x)) and sup-min transitive .

Definition 8: the lower and upper approximate of attribute set B, B ⊆ A, regarding the concept set X is define as:

Here, X(u) denote the membership degree of u to a fuzzy set X and RB (x, u) denote the fuzzy equivariance relation between x and u regarding to attribute set B. And RB (x, u) = , where T is a t-norm aggregation function in fuzzy set theory.

One would notice that the definitions of lower and upper approximate is different from the one in the crisp rough set. These definitions are representing the membership degree of x to such lower and upper approximate instead of representing a set of elements.

In the contest of decision table, we mostly concern able the lower and upper approximate membership degree of each x, x ∈ U, regard to its decision class. That is:

In the assumption of our models, decision attribute is always crisp (nominal), then the membership of u to [x]D becomes: . And the definition could be refined as:

Definition 9: the lower and upper approximate of attribute set B, B ⊆ A, regarding the decision class is define as:

(1)

(2)

The refined lower approximate membership function will be used in our 3 membership rough set models to detect the membership degree of object belong to its decision class’s lower approximate and to making prediction, whereas the binary fuzzy relation is replaced by similarity measures between attributes, which will be covered in next sub-section.

Definition 10: According to [cite], A fuzzy discernibility matrix M () with size n × n is defined as:

5.2 similarity relation for real value attributes

Given a decision table , and R is a similarity relation define for real value attribute a, a ∈ A, if and only if R satisfy reflexivity and symmetry. We could define many similarity relations for real value attributes, such as:

1. Min-max scale similarity: .
2. Gaussian similarity:

Here , , denote the maximum and minimum value for attribute , and denote the variance of attribute .

Our membership models will use the Gaussian similarity to calculate the similarity relation between real value attributes. For nominal attributes, we use equivariance relation as defines in crisp rough set.

For attribute subset , would be the aggregation function of . And such aggregation function could be defined as:

3.Minimum T-norm in fuzzy relations:

4.Product T-norm in fuzzy relation:

5.

Equation 5 is suitable for decision table that is a mixture of nominal and real value attribute since it could preserve some tolerance even when the objects disagree in nominal attribute and the two T-norm version is suitable in the case that all attributes are real value or ordinal.

6. Incremental membership rough set models

This section will cover the detail of the proposed: IVMRS, ISwMRS and TFMRS. All these three models find and update new reduct by adopting discernibility relation from [cite]. The motivation of these 3 models is that in crisp rough set, real value data require categorized or discretized before processed and the equivariance relation is too rigorous, resulting in lost of similarity and connectivity among real value attribute. Section 6.1 will introduce these discernibility relation techniques as our framework and the 3 proposed model will be covered in section 6.2.

6.1 Discernibility relation in fuzzy rough set for incremental attribute reduction

In our three membership rough set models, we adopt and modify the discernibility relation techniques in [cite] to find the reduct in the dataset. For prediction, we check the membership degree of the predicted object to the lower approximate of each decision class and pick the decision class with highest degree. This subsection will cover the algorithm we used to find reduct in our decision table, update reduct when new samples arriving and make prediction.

Algorithm 8. Attribute relation tensor generation

Input: A decision table DT = (U, A + D)

Output: A relation cube of DT: cube

cube= []

For a in A:

Relation\_mat = []

For id1 in U:

Sim\_row = []

For id2 in U:

Sim\_row. 🡨 sim(id1,id2)

Relation\_mat 🡨 sim\_row

cube 🡨 Relation\_mat

The generated cube is simple an array of relation matrix of each attribute. The entry eij in the matrix of attribute a is the similarity of sim(a(i), a(j)), here we use gaussian similarity for real value attributes and strict equivalence for nominal attributes

Algorithm 9. Similarity of two objects on specific attribute set

Input: Obj1, Obj2, relation Cube, compared attribute set B

Return: , the similarity of two object on B

Algorithm 10. Attribute reduction in membership rough set

Input: A decision table DT = (U, A + D)

Output: A reduct of DT: Red

Step 1: use algorithm 8 to get the relation cube, Cube

Step 2: according (2) in definition 9, using algorithm 9 to calculate the membership degree of each object to its decision class’s lower approximate

Step 3: According to definition 10, compute the discernibility matrix M.

Step 4: According to definition 6, compute Dis(A) and Dis(a) for all a in A

Compute Dis(Red+{a}) for a in A – Red

Select a0 such that Dis(Red + {a0}) is maximum

Let Red = Red + {a0}, and Dis(Red) = Dis(Red + {a0})

Careful reader might notice that step 4 to 7 is identical to the procedure in algorithm 1, this is because in our membership rough set, things changed is the discernibility of each attribute and the new concept of membership degree to lower approximate of decision class. The discernibility of attribute *a* to each pair is now depends on the membership degree of object and the similarity relation of two object regard to attribute *a* instead of strict equal.

Algorithm 11. Incremental update attribute discernibility when new samples arriving in membership rough set

Input: A fuzzy decision table DT = (U, A+D)

Similarity relation Cube

Dis(a) for a in A; Dis(A); and Dis(Red)

New samples U0

Output: Disnew(a) for a in A; Disnew(A); and Disnew(Red)

Step 1: expand and update the relation Cube by adding new samples U0

Step 2: U = U + U0

Step 3: according to definition 10, compute

Step 4: for x in U – U0:

for o in U0:

if d(x) != d(o):

if 1 – RA(x,o) < :

= 1 – RA(x,o)

Step 5: for x in U:

for y in U:

for a in A:

if (x,y) not in Dis(a) and 1 – Ra(x,y) >= :

Dis(a) 🡨 (x,y)

If (x,y) not in Dis(A):

Dis(A) 🡨(x,y)

If a in Red and (x,y) not in Dis(Red):

Dis(Red) 🡨(x,y)

Here, Step 3 compute the membership degree for all x in U0 to its decision class; Step 4 update the membership degree for all x in U - U0 by the definition, since we already compute 1 – RA(x,o) for o in U - U0 and get , we only need to check for 1 – RA(x,o) for o in U0 and decide whether to update ; Step 5 update the discernibility of all the attributes and attribute set A as well as Red. After step 5, we could use step 3, 4, 5 in algorithm 2 to find the new reduct.

Since in the membership rough set model, real value attributes will not be discretized, it is impossible to inducing specific decision rules. To make prediction on the membership rough set model, one might need to compare with every object in the decision table. However, as the dataset getting larger and larger, such comparison will be slow and not scalable in the context of stream mining. On the other hand, we could select representative objects in the decision table as ruling objects. Whenever we need to predict an unknow object, we will use these ruling objects as a rule to make predictions. The following will be some definition about ruling objects, these definitions are adopted and modified from [cited] to fitted to our membership rough set models.

Definition 11(1): (rule in membership rough set) Given an object x in U, fdx is called the rule represented by object x. fdx|A is the condition of the rule and fdx|D is the decision of the rule, and denote fdx|A 🡪 fdx|D

Here, since in rough set system, we usually find the reduct attributes and based on the reduct attribute to induce rules, the condition of the rules usually be Red, the reduct of A. In such context, we could redefine the rule in membership rough set as:

Definition 11(2): let fdx be the rule represent by object x after attribute reduction, then fdx|Red is the condition of the rule and fdx|D is the decision of the rule, and denote fdx|Red 🡪 fdx|D

Definition 12: (reequipment to match a rule fdx ) Let y be any object, we said y match rule fdx if and only if 1-RRed(x,y) < .

Here, will be the membership degree of x to its decision class. By the previous definition, equal to the inf(1-RRed(x,z)) for any d(z) != d(x). As a result, any y such that 1-RRed(x,y) < could be conclude as d(y) = d(x)

Definition 13: (rule covering) Given any object x in U and its corresponding rule fdx, for an arbitrary object y in U, and fdy, if y matches rule fdx, we will also say fdx covers fdy.

This definition is an important concept for us in the process of mining and simplifying ruling objects.

Algorithm 12. Decision Ruling Object induction in membership rough set

Input: A decision table DT, and the reduct attribute of DT

Output: A group of decision ruling objects

Step 1: Sort the objects in the decision table by their membership degree to decision class’s lower approximate and store in S

Step 2: Let all\_rule = []

Step 3: Let x = S[0], S = S – S[0], x\_cover = 0

Step 4: for y in S:

If d(y) = d(x) and 1-RRed(x,y) < : S = S – y, x\_cover ++

end for

add tuple (x,x\_cover,d(x)) to all\_rule, represents (rule id, coverage, decision)

if |S| != 0, go back to step 3

This algorithm is a heuristic algorithm, giving higher priority to object that with higher membership degree by sorting at step 1. In step 4, it try to check all y with the same decision class and see if there is any y match the rule fdx(according to definition 12), and trim fdy according to definition 13. By such procedure, this algorithm tend to return minimum set of rule.

Algorithm 13. Prediction in membership rough set

Input: Rule in decision table, predicted object

Output: predicted decision

Step1: sort the rule by coverage

Step2: return the decision in the first match rule. (matching by definition 12)

Step3: if no rule is matched, return the decision of rule object with the highest similarity to predicted object.

Here, since it could be the case that there is matching rule, we choose the most similar object as the rule.

Reader should keep in mind that using simplify rule will lead to some accuracy discount in the model, however, it could prevent overfitting as well as increasing predicting speed.

6.2 3 proposed incremental membership rough set models:

The 3 proposed incremental membership rough set models have the same basic structure as the 3 proposed crisp rough set model in initializing and updating with simple modification in finding reduct and updating reduct as describe at algorithm 10 and 11.

7. experiment

Comparison

Performance of our model

8.conclusion and future work

Future work

Conclusion