Using Case Based Reasoning to Support The Retrieval of Incident Reports

Saw Htun Naing Soe
University of Computer Studies, Mandalay
sawhtunnaingsoe@gmail.com

ABSTRACT

Case-based reasoning (CBR) solves new problems by adapting previously successful solutions to similar problems. A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems is referred to as a previous case, stored case, or retained case. Correspondingly, Incident Reporting System can be used to strengthen the defenses that lead to the detection and resolution of potential problems and to detect problems before they results in an accident. This paper intended to retrieve information about previous incident and also use for incident report classification. So, Emergency and Out Patient Department of General Hospital will have safe time and reliability to retrieval and classify incident reports by using this system.

1. INTRODUCTION

Case-based reasoning systems were developed to help people identify previous situations that match aspects of a current problem. They were also developed to provide guidance on how to solve problems and make decisions. In this system involves decision making situations and case-based reasoning method to support the retrieval of incident reports. Incident reports provide an important defense against future failures in many safety-critical industries. They provide engineers, designers, managers and operators with important guidance about potential problems in existing systems. The relative frequency of incidents, as opposed to the relative infrequency of accidents, helps to ensure that there is a continuing focus on safety issues. Incident reports are also amenable to statistical analysis in a way that accident reports are not. It can be difficult to ensure that all sections of a workforce continue to participate in the reporting of incidents [3].

There are also problems associated with maintaining trust in the anonymity or confidentiality of such systems. In spite of these difficulties, more and more industries are relying upon the insights provided by incident reporting schemes. Case-Based Reasoning is described in section2.CBR cycle is explained in section3.K-Nearest neighbor classifier is expressed in section4.Extracting classification rules from decision tree includes algorithm for decision tree and example for liver case study are illustrated in section5.System design and implementation are described in section6 and section7.Conclusion is described in section8.

2. CASE-BASED REASONING

Case-based reasoning (CBR) classifiers are instanced-based. The samples or "cases" stored by CBR are complex symbolic descriptions. CBR has also been applied to areas such as engineering and law, where cases are wither technical designs or legal rulings, respectively. When given a new case to classify, a case-based reasoner will first check an identical training case exists. If no identical case is found, then the case-based reasoner search for training cases having components that is similar to those of the new case. Once similar cases have been retrieved, the next step in CBR is adaptation, which is the process of modifying previous solutions to address the new problem [1]. Conceptually, this training case may be considered as neighbors of the new case. The case-based reasoner tries to combine the solutions of the neighboring training cases in order to propose a solution for the new case. If incompatibilities arise with the individual solutions, then backtracking to search for other solutions may be necessary. The case-based reasoner may employ background knowledge and problem-solving strategies in order to propose a feasible combined solution.

3. CBR CYCLE

At the highest level of generality, a general CBR cycle may be described by the following four processes:

1. RETRIEVE the most similar case or cases.
2. REUSE the information and knowledge in that case to solve the problem
3. REVISE the proposed solution.

4. RETAIN the parts of this experience likely to be useful for future problem solving.

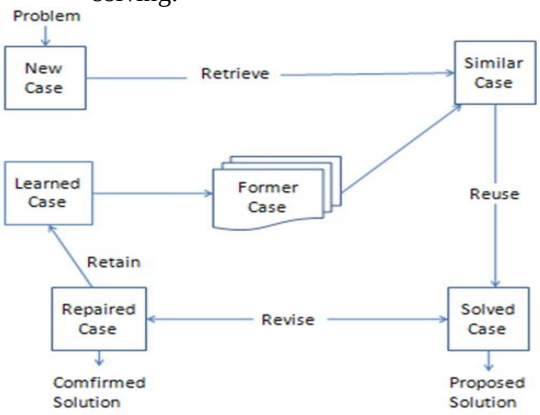


Figure1. General model of case-based reasoning

A new problem is solved by retrieving one or more previously experienced by incorporating it into the existing knowledge-base (case-bade). The four processes each involve a number of more specific steps, which will be described in the task model. In figure 1, this cycle is illustrated.

An initial description of a problem (top of figure) defines a new case. This new case is used to RETRIEVE a case form the collection of previous cases. The retrieved case is combined with the new case-through REUSE_ into a solved case, i.e., a proposed solution to the initial problem. Through the REVISE process this solution is tested for success, e.g., by being applied to the real world environment or evaluated by a teacher, and repaired if failed. During RETAIN, useful experience is retained for future reuse, and the case base is updated by a new learned case, or by modification of some existing cases [4].

4. K-NEAREST NEIGHBOR CLASSIFIER

Nearest neighbor classifiers are based on learning by analogy. The training samples are described by n-dimensional numeric attributes. Each sample represents a point in an n-dimensional space. In this way, all of the training samples are stored in an n-dimensional pattern space. When given an unknown sample, a k-nearest neighbor classifier searches the pattern space for the k-training samples are closest to the unknown sample. These k-training samples are the k "nearest neighbors" of the unknown sample. "Closeness" is defined in terms of Euclidean distance, where the Euclidean distance between tow points, X = (x_1 , x_2 , …. , x_n) and Y= (y_1 , y_2 , . . . , y_n) is

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

The unknown sample is assigned the most common class among its k nearest neighbors. When k = 1, the unknown sample is assigned the class of the training sample that is closest to it in pattern space. Nearest neighbor classifiers are instance-based or lazy learners in that they store all of the training samples and do not build a classifier until a new (unlabeled) sample needs to be classified. Analyses of the k-nearest neighbor algorithm suggest that it is a more accurate classifier than the nearest neighbor algorithm [2].

4.1 Characteristics of Nearest Neighbor Classifiers

The characteristics of the nearest-neighbor classifier are summarized below:

(1) Nearest-neighbor classification is part of a more general technique known as instance-based learning, which uses specific training instance to make predictions without having to maintain an abstraction (or mode) derived from data. Instance-based learning algorithms require a proximity measure to determine the similarity or distance between instances and a classification function that returns the predicted class of a text instance based on its proximity to other instance.

(2) Lazy learners such as nearest-neighbor classifiers do not require model building. However, classifying a test example can be quite expensive because we need to compute the proximity values individually between the test and training examples.

(3) Nearest-neighbor classifiers make their predictions based on local information.

(4) Nearest-neighbor classifiers can produce arbitrarily shaped decision boundaries. Such boundaries provide a more flexible model representation compared to decision tree and rule-based classifiers that are often constrained to rectilinear decision boundaries. The decision boundaries of nearest-neighbor classifiers also have high variability because they depend on the composition of training examples.

(5) Nearest-neighbor classifier can produce wrong prediction unless the appropriate proximity measured and data preprocessing steps are taken.

(6) A decision-tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The topmost node in a tree is the root node.

In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree, a path is traced from the root to a leaf node that holds the class prediction for the sample. Decision trees can easily be converted to classification rules.

4.2 Algorithm of the K-Nearest Neighbor Classifier

Algorithm :*K-Nearest*
Input : 1. Selected training record sets *D*,
2. Unknown record set *R*.
3. User specified *K* value
Output : *K* – Nearest record sets
Method :FindingNearestRecordsets
Create integer array *NearestIndex* as size of *K*
Create double array *MinValue* as size of *K*
For each *row* in *D.rows*
distance = *clacDistance* (*R*, *row*)
recordIfCloset (*row.Index*, *distance*)
End For
If *isExistIdentical*() Then
Display identical record.
Else
Display nearest neighbors.
EndIF
If *isExistIdentical*() Then
Display identical record.
Else
Display nearest neighbors.
EndIF

Function : *calcDistance (Unknown record set R, Training I*
Return : *Distance* of type *double* between *R* and *D.row*.
For *i*=0 through *D.columns.count* – 1
IF *R.Column.DataType* is *not Numeric* Then
Str1 = *R.Columns[i].Value*
Str2 = *D.Columns[i].Value*
IF *Str1* = *Str2* Then
weightDiff = 0
Else
weightDiff = 1
EndIf
Else
weightDiff = *R.Columns[i].Value* – *D.Columns[i].Value*
EndIF
distance += *weightDiff* * *weightDiff*
End For
distance = *Square root* of *distance*
Return *distance*

4.3 Example for Liver Case Study

There are six attributes in training sample and two class labels present and absence. They are ID(Number),PlasmaBilirubin(Number),Alonine Transaminase(Number),AspartateTransaminase(Number),AlkalineTransaminase(Number),Albumin(Number),GammaglutamylTransferase(Number),Class Label(Text).If user input threshold value(3) for similar document count. And also input values of Plasma Bilirubin 34,Alonine Transaminase45,Aspartate

Transaminase60,AlkalineTransaminase35,Albumin24,Gammaglutamyl Transferase56.In this system, user need to input threshold value for retrieves nearest similarity record when the identical result is not exist into the system.

The system retrieves nearest similarity record based on input symptoms and threshold values amount are as follow:

ID -308
PlasmaBilirubin-87
AlonineTransaminase-54
AspartateTransaminase-41
AlkalineTransaminase-29
Albumin-23
GammaglutamylTransferase-6
Class Label- Absence

ID -326
PlasmaBilirubin-91
AlonineTransaminase-52
AspartateTransaminase-76
AlkalineTransaminase-32
Albumin-24
GammaglutamylTransferase-8
Class Label- Absence

ID -328
PlasmaBilirubin-87
AlonineTransaminase-55
AspartateTransaminase-36
AlkalineTransaminase-19
Albumin-25
GammaglutamylTransferase-8
Class Label- Absence

5. EXTRACTING CLASSIFICAION RULES FROM DECISION TREE

The knowledge represented in decision trees can be extracted and represented in the form of classification. IF-THEN rules, one rule are creating for each path from the root to a leaf node. Each attribute-value pair along a given path forms a conjunction in the rule antecedent ("IF" part). The leaf node holds, the class prediction, forming the rule consequence ("THEN" part) [5].The IF-THEN rules may be easier for humans to understand, particularly if the given tree is very large.

5.1 Algorithm for Decision Tree

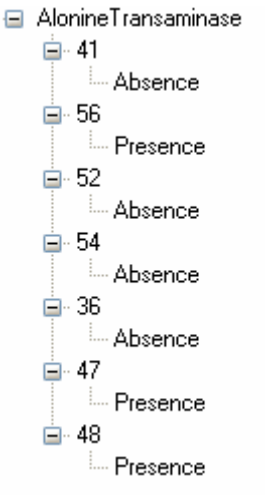
```
Algorithm : Decision Tree
Input     :1. Column count colCount of training record sets D.
           2. Attribute value rowFilter as to extract D.rows.
           3. Training record sets D.
           4. rootName as to filter attributes of D
           5. root as tree node
Output :        Decision tree
Method :        generateTree

IF colCount > 0 and rootName is not Empty Then
Create childNode as tree node
Create string array colNames as size of colCount
Initialize j to -1


  For i = 0 through colCount
   If D.Columns[i].Name <> rootName Then
   j += 1
   colNames[j] = D.Columns[i].Name
   EndIF
  EndFor
  Create data table child and set it by colNames
  child.setData (D, rootName, rowFilter)
  IF child.Rows.Count > 0 Then
  IF child.isAllSameResults () = False Then
     Compute Total Info Gain for each Result
     For I = 0 through child.Columns.Count - 2
        Compute Info Gain of child.Columns[i]
        Record max Info columns in maxInfoCol[i]
     End For
     Create new tree node tNode with max Info Gain column name
     If root is nothing Then
        Add tNode to decision tree as root node
     Else
        Add tNode to decision tree as child node
     Endif
     For each col in maxInfoCol[i]
        Recursively call generateTree
     End For
  Else
     Create a new tree node as rNode with decision result
     Add rNode to decision tree as child node
  EndIf
  EndIf
  EndIF
```

5.2 Example for Liver Case Study

The system extracts tree as follow:



The system also extract rule are as follow:

1. IF AlonineTransaminase = ''41'' THEN ''Absence''
2. ELSE IF AlonineTransaminase = ''56'' THEN ''Presence''
3. ELSE IF AlonineTransaminase = ''52'' THEN ''Absence''
4. ELSE IF AlonineTransaminase = ''54'' THEN ''Absence''
5. ELSE IF AlonineTransaminase = ''36'' THEN ''Absence''
6. ELSE IF AlonineTransaminase = ''47'' THEN ''Presence''
7. ELSE IF AlonineTransaminase = ''48'' THEN ''Presence''

The prediction results for user Unknown data as follow:

PlasmaBilirubin-34
AlonineTransaminase-45
AspartateTransaminase-60
AlkalineTransaminase-35
Albumin-24
GammaglutamylTransferase-56
Class Label- Absence

6. SYSTSEM DESIGN

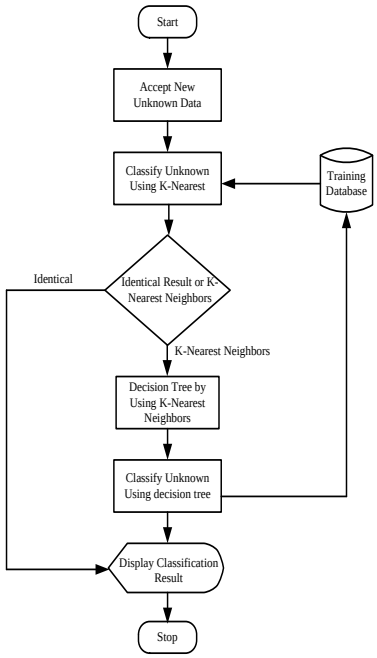


Figure2. System flow diagram

In this system, there are four processes. The first process is accepted the new unknown data. The second process is classifying unknown data by using K-Nearest Neighbor. If the result is identical result then this result is display classification result. If the result is no identical result, then this result is to calculate in the decision tree method. The third process is generated to the decision tree by using K-Nearest Neighbor algorithm. The four process is classify to the unknown data by using decision tree and then the result display classification result and the result is retained training database to be useful for future problem solving.

7. IMPLEMENTATION

In this system, user can choose the desire trainig sample of disease from the database. Sytem will be displayed the existing training database in the list box. And then,user can input symtoms to the system.

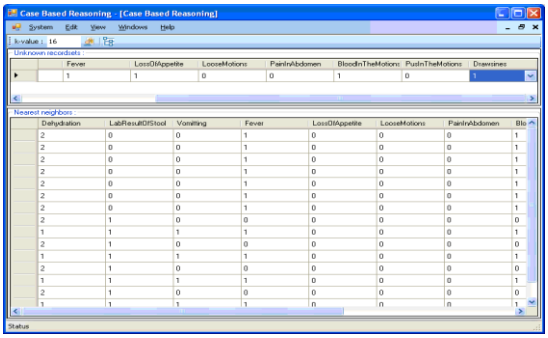


Figure3. Nearest neighbors

After that, the system classified the unknown input data. The system classified unknown disease by k-Nearest algorithm as shown in Figure3.

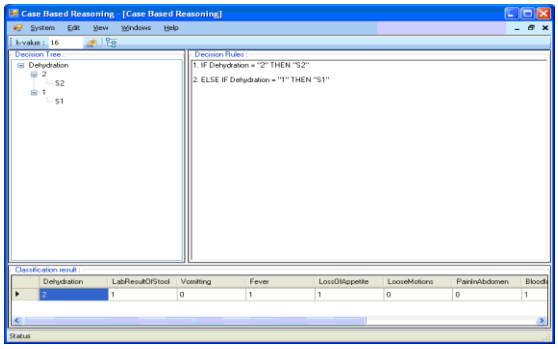


Figure4. Classification by Decision Tree

In this system, the system displayed the classification result for unknown disease using decision tree induction as shown in Figure4. User can add and modify to training data from the system.

8. CONCLUSION

In this paper we have discussed the centrality of feature-value representation of cases in the Case-based Reasoning (CBR) paradigm. We have argued that in some circumstances benefits can accrue from the use of similarity measures that work more directly on the raw data. The benefit might be that the overall design of the system is simplified because it embodies specific knowledge about the data. CBR is based on the premise that problem solving involves recalling past experiences and utilizing these experiences (cases) to solve novel problems. It is a particularly useful paradigm in domains that are not well understood and where it is difficult to come up with generalizations that can be used to model the world. Thus there are two considerations for CBR research. The first is that a broader perspective on similarity along the lines discussed in this thesis may be useful. The second is that the partiality of these computationally expensive similarity measures may depend on clever retrieval techniques such as Decision Tree to make them computationally tractable.

REFERENCES

[1] A.Aamodt."**Case-BasedReasoning: Foundational Issues,Methodlogical Variations, and System Approaches**", E.Plaza (1994).

[2] Chris Johnson, "**Using Case-Based Reasoning to Support the Indexing and Retrieval of Incident Reports**" Department of Computing Science, University of Glasgow, Glasgow,G12 8QQ.

[3] David W.Aha and Dennis Kibler"**Noise-Tolerant Instance-Based Learning Algorithms**", Department of Information and Computer Science,University of California,Invine,CA92717.

[4] Igor Jurisica and Janice Glasgow, "**Applications of Case-Based Reasoning in Molecular Biology**", AI Magazine Volume 25 Number 1 (2004).

[5] Maribel Yasmina Santos and Adriano Moreira "**Automatic Classification of Location Contexts with Decision Trees**", Department of Information Systems, University of Minho, Campus de Azurem,4800-058 Guimaraes,Portugal.