Decisional DNA with Embedded RELIEF-F and Linear Regression for Knowledge and Experience Management

Peng Wang¹, Cesar Sanin¹ and Edward Szczerbicki²

¹School of Engineering, Faculty of Engineering and Built Environment
The University of Newcastle, Australia
²Gdansk University of Technology, Gdansk, Poland
peng.wang@uon.edu.au, Cesar.Sanin@newcastle.edu.au, Edward.Szczerbicki@zie.pg.gda.pl

Abstract: The constant growth of information on the wide spread of the web is causing many difficulties in finding and extracting useful knowledge on the web. Set of Experience Knowledge Structure or Decisional DNA as knowledge representation provides features such as learning from experience, dealing with noisy and incomplete data, making precise decision and predicting. It can be a core component of a new structure with a feature selection model. This new structure is able to extract knowledge in DDNA structure by many different approaches such as web crawling and reading CVS files. Afterwards, the feature selection model is used to rank elements of set of experience structure for predicting purposes. This paper shows the application of feature selection models to enhance the Prognosis Macro process of Decisional DNA based web crawler for prediction.

Keywords. Decisional DNA: Set of Experience Knowledge Structure; data mining, prediction Analysis; Relief-f

1 Introduction

Knowledge has played a remarkable role on modern social human activities since the booming development of the internet. This advent of rapidly growing information on the internet has resulted in a huge volume of data stored in various formats as plain files, multimedia data, Markup Language(ML) format and search queries among others [13], and this irresistible trend has posed opportunities to new techniques handling variously formatted information to extract useful knowledge. In fact, the majority of data has never been reused since it was made available on the internet. If this data collected from different fields is appropriately discovered and extracted, it can produce enormous hidden knowledge. Therefore, a simple, explicit and effective method which is able to extract useful knowledge from huge repositories, and be easily reused, has been demanded [5]. On the other hand, data mining or knowledge discovery in databases (KDD) is a mature technology that has been widely used to extract patterns from data repositories. It is an inclusive technique that includes statistics, Artificial intelligent (AI), machine learning, pattern discovery and database technologies. Hence, it can be applied to study data and find any trends or patterns in order to guide users to make it reusable [6]. Furthermore, Set of Experience Structure(SOEKS) or Decisional DNA as a representation structure combines filtered information acquired from formal decision events to facilitate uncertain, implicit and incomplete data to help users make more precise decisions or predictions [12]. On this basis, this paper refines decisional DNA based web crawler [13] with data mining techniques. The RELIEF-F algorithm is a data mining technique that enhances capabilities of the SOEKS to make predictions in a more precise and efficient way. An experiment is introduced in this paper to compare with the traditional prediction method in Weka [1]. Furthermore, function of the SOEKS is tested by embedding linear regression method.

2 Background

2.1 Set of Experience Knowledge Structure (SOEKS) and Decisional DNA

Web Data Mining is currently working with different types of knowledge. The idea behind it is to store and manage knowledge in some manners. In other words, mining web data is the process of storing,
retrieving, distributing and sharing knowledge. However, web information is mostly unstructured or semi-structured in huge quantities. Thus, a technology which can be used to capture and store formal decisional events as explicit knowledge is necessary. The Set of Experience Knowledge Structure (SOEKS or shortly SOE) as a flexible and independent knowledge representation is a suitable tool for this task. Moreover, it has also been used to collect and store formal decisional events in an explicit manner [12]. Therefore, the SOEK can be a pattern based on existing and available knowledge offered by a formal decision event with dynamic structure. It can be expressed in XML or OWL as ontology in order to make it shareable and transportable [3, 8, 10].

The SOEKS is composed of variables, functions, constraints and rules [9]. The variables commonly use an attribute-value language to represent knowledge (i.e. by a vector of variables and values) [7]. It is the starting point for the SOEKS and the infrastructure of the SOE because they are the source of other components. Functions are made up of interactions of variables which include dependent variables and a set of input variables. On the other hand, according to the tasks of the decision event, functions are brought to reasoning optimal states. Therefore, this second component of the SOE establishes the relations between variables restricting experience on decision-making. Constraints are another factor of association amongst the variables. Though constraints are another way of functions, they have a different purpose. They limit the performance and possibility of a system and restrict the feasible solutions in a decision problem. Lastly, Rules are another form of expressing links among variables. They condition the relationships that operate the universe of variables. In the other words, they use the statements IF-THEN-ELSE to connect consequence with a condition.

Additionally, the SOEKS is structured in view of some important features of DNA. It imitates a gene in combining four nucleotides of DNA by integrating four components of the SOE, as a distinctive structure to adapt different needs. The components are not isolated between there four, but they connect each other. In the same way as a gene producing a phenotype, the SOE yields a value of decision with their elements. Each SOE can be categorised and acts as a gene in DNA [12]. A set of the SOE in a same category makes up of a decisional chromosome which stores decisional strategies for that category. After this, each module of chromosomes establishes an entire inference tool to offer a blue print of knowledge inside an organization [9].

A similarity metric is one of the fundamental concepts of Knowledge Discovery (KD). It provides an effective way to improve company’s strategic and operational effectiveness and efficiency. Common similarity methods are measured by computing distance between two objects. This distance or similarity is used to deal with predictions, hypothesis testing and rule discovery [8, 11]. Therefore, the SOEKS introduces a geometrical function, an effective similarity metric, which individually calculates similarities among the variables, the functions, the constraints and the rules, to further produce a joint similarity value. It provides a scalar measure of the similarity between two objects. Those series of the similarities are ultimately used for prediction purpose [9]. Usually, the weight has to be calculated by expertise because its value dramatically affects a preciseness of the prediction [8].

### 2.2 Data mining

Data Mining is defined as a process of discovering potentially useful, non-trivial, effective and ultimately meaningful patterns from existing data. It is a technique to solve problems by analysing or learning data. The data mining techniques extract useful and structured patterns. Once the patterns are found, they are used to inform future decision and make prediction. Many integrated and implicated algorithms are involved in the data mining technique. It has been comprehensively developed in many domains which include retail, Healthcare, finance, terrorism detection among others.

Weka was developed nearly two decades ago as a popular machine learning software. It is a comprehensive data mining application collecting machine-learning algorithms for data mining tasks [1]. It uses graphical user interfaces and visualization utilities for data exploration and algorithm evaluation. There are many processes included such as data pre-processing, classification, regression, clustering, association rules and visualization. Because of its package based architecture, developed packages are flexibly loaded into the system. One of the most satisfying aspects of Weka is that the software has been incorporated into many other open-source projects [1]. In our experiment, the Weka will be used to compare with our new structure.

Many searches focus on feature selection within the area of application. It is a major component of the data mining prediction procedure. Three objectives of the feature selection are improving the prediction performance of the predictor, providing faster and cost-effective predictor, and providing a better understanding of the underlying process that generated data [4]. The RELIEFF evaluation is one of the most valuable feature selection methods to evaluate the quality of the features because of its
effectiveness and simplicity. It is based on the nearest neighbour paradigm to enhance the relevant level of different feature values between different concept values of example pairs [4]. Then, the features are looped and calculated ranks for a series of weight contributions. This weight ratio can finally be used in our structure to assist DDNA for prediction as Fig. 2.

Linear Regression, in statistical word, is an approach to analyse affection of a scale of the dependent variables to one or more explanatory variables. This regression uses linear predictor functions to model data and to find unknown parameters. It is the former type of regression to be extensively used in practical applications. In other words, if the model has been found, the prediction can be done without data source. As this result, it is passable to use it to predict a query when the query is out of range of the existing data source. We attempted embedding this model to enhancing our structure to adapt to different circumstances.

2.3 Optimization approach to Decisional DNA Application

We first present a brief review of DDNA based a web crawler structure [13] (see Fig.1). This architecture proved that web information is able to be converted into the Decisional DNA structure by four Macro Processes namely, Diagnosis, Prognosis, Solution and Knowledge Macro processes. The idea of this structure based on SOEKS is to collect incomplete and noisy information by a web crawler in the diagnosis process. Afterwards, it transfers those crawled information into the DDNA structure for reusing and sharing purposes.

This paper is based on the previous research result in order to explore the functionality in the Prognosis, the Solution and the Knowledge Macro Processes in the decisional DNA. In addition, it indicates a new model to embed the RELIEF-F and the Ranking method into the Decisional DNA structure for automation and prediction procedures. This model offers many advantages. It can gain knowledge from different repositories. It stores different kind of information into Extensible Markup Language (XML) for sharing and transferring purposes. It not only share knowledge, but also assists in decision making and prediction process. This structure automatically produces a series of similarities between existing pares of experiences. Those similarities are commonly computed for predictions. It contains four main processes. They are diagnosis, prognosis, solution and knowledge macro processes.

![Fig.1. Architecture of Decisional DNA-Based Web Crawler [13]](image-url)
In the Diagnosis, information can be collected from the website by a web crawler component or any repositories such as reading CSV files. And then, those collected information is integrated by a integration component for further analysis [13]. Once the Decisional DNA has been reached, a RELIEF-F evaluator and ranking method is used to rank the attributes in the Prognosis Macro Process. It generates a series of scores related to the attributes. Equation 1 is used to calculate the percentage of the weights for the SOEKS, and this equation illustrates how the system generates the ratio \( R \) that should be allocated to the corresponding attribute. These numbers are between -1 and 1 and indicate how important the related attributes are. The vector \( O \) is the output given to the \( j^{th} \) attributes, and \( n \) is the total amount of the attributes. Therefore, according to equation 1, the outputs related to the attributes are calculated and exemplified in table 1. We designed an integration class to match those result to the weights associated with the same name of the DDNA variables.

\[
R_j = O_j \left/ \left[ 2 \sum_{j=1}^{n} O_j \right] \right.
\] (1)

The SOEKS has been designed to deal with the most valuable asset: knowledge, and it recognises patterns automatically in large multivariate data sets assisting decision making processes. As knowledge representation requirements, one of core functionalities of the SOEKS is to calculate differences between two data objects. In addition, a ranking of the result can be provided by choosing the best-matching objects [12]. It uses multidimensional scale for a similarity metric. For instance, the equation calculates distance \( d_{ik} \) between a pair of sets of Experience \( E_i \) and \( E_j \). \( E_a \) and \( E_b \) are the \( k^{th} \) attribute of the sets of \( E_i \) and \( E_j \). \( W_k \) is the weight given to the \( k^{th} \) attribute, and \( n \) is the number of variables on \( E_i \). The similarity metric example for variables takes the following equation [8]:

\[
s_v(E_i, E_j) = \sum_{k=1}^{n} w_k \left[ \frac{|E_{ik} - E_{jk}|}{\max\{|E_{ik}|, |E_{jk}|\}} \right]^{1/2} \forall k \in E_i \wedge E_j
\] (2)

The SOEKS has different elements and each of them has its own characteristics, which can be separately used in computing similarity to compare with the query objects. The value of similarity is between 0 and 1. If an element has the highest similarity or zero value, the element will be the most similar or identical object and vice versa. It can be seen from equation 2 that the result is in some extent affected by how important attributes are (weight \( W_k \)). Experts may be able to decide the percentage of the weight rate. However, subjective mistakes can be made by human beings, and it will influence decision making. This is a great challenge to find a better automatic and objective way instead of the experts. The RELIEF-F measure is one of the best methods for the feature selection and it can be designed to automatically compute and rank the attributes [4]. Therefore, it will be suggested...
to assist the DDNA in our new structure for prediction. Similar but more complicated methods among functions, constraints and rules calculate similarity as in the variables above. The similarity feature makes the SOEKS comparable and classifiable, making it available for many different systems and technologies [8].

3 Experiment Evaluation

In order to evaluate the preciseness and effectiveness of our new model, we implemented this structure in MacBook laptop with MAC system. We used the wine quality dataset [2] for testing purposes. This dataset contains 4898 records and 12 features. It was separated into two groups; one of 100 records for testing purposes and remaining 4798 records use for training of the SOEKS. The whole experiment is based on java language. In this experiment, our focus is on the performance of the SOEKS prediction.

Initially, the system demands data from various data sources such as web crawling or reading CSV files and among others. We generated a Prognosser class in the system with the responsibility of loading the data set. And then, it translates it into two types of instances. One is to evaluate attribute ranking and another is for SOEKS prediction. For the purposes, ReliefFAttributeEval class was created to implement RELIEF-F algorithm. Then, the CSV file was inputted into the system by using an array with two parameters. The parameter –k reveals number of nearest neighbours used to estimate attribute relevance. In our experiment, 3 nearest neighbours were used. Another parameter –i is used to tell the system the name of input the file. We used the runEvaluator to create the initial instance. The following code illustrates this process.

```java
ReliefFAttributeEval relieffAttributeEvaluator = new ReliefFAttributeEval();
String[] args = {"-K", "3", "-i", "C:\winequality-white.csv"};
ReliefFAttributeEval.runEvaluator(relieffAttributeEvaluator, args);
```

Afterwards, weka produces a result as seen in figure 3. There are three columns generated; they are ranked weights, position of related attribute in dataset and names of attributes.

![Attribute Selection Output](image)

According to the requirement of ranked weights discussed above, we need to collect this series of numbers to be used by SOEKS. Therefore, the system gained a series of number by invoking an array variable m_weights in the instance relieffAttributeEvaluator. After that, we used two loops to implement the equation 1 as follows:

```java
double[] rankedWeights = new double[relieffAttributeEvaluator.m_weights.length];
double amount = 0;
for (int i = 0; i < relieffAttributeEvaluator.m_weights.length; i++)
{
    if (relieffAttributeEvaluator.m_weights[i] > 0)
        amount = amount + relieffAttributeEvaluator.m_weights[i];
}
```

Fig.3. Attribute Selection Output
amount=2* amount;
for (int i = 0; i < relieffAttributeEvaluator.m_weights.length; i++)
{
    if (relieffAttributeEvaluator.m_weights[i] > 0) {
        rankedWeights[i] = relieffAttributeEvaluator.m_weights[i]/ amount;
    } else {
        rankedWeights[i] = 0;
    }
}

Eleven attributes were ranked and calculated weights according to a series of numbers produced by weka. Table 1 illustrates distributed weight for each attribute.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Output</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>volatile_acidity</td>
<td>0.00483</td>
<td>12.510%</td>
</tr>
<tr>
<td>alcohol</td>
<td>0.00451</td>
<td>11.681%</td>
</tr>
<tr>
<td>total_sulfur_dioxide</td>
<td>0.0044</td>
<td>11.396%</td>
</tr>
<tr>
<td>pH</td>
<td>0.00432</td>
<td>11.189%</td>
</tr>
<tr>
<td>fixed_acidity</td>
<td>0.00429</td>
<td>11.111%</td>
</tr>
<tr>
<td>free_sulfur_dioxide</td>
<td>0.00423</td>
<td>10.956%</td>
</tr>
<tr>
<td>sulphates</td>
<td>0.00356</td>
<td>9.220%</td>
</tr>
<tr>
<td>citric_acid</td>
<td>0.00296</td>
<td>7.666%</td>
</tr>
<tr>
<td>residual_sugar</td>
<td>0.00205</td>
<td>5.310%</td>
</tr>
<tr>
<td>density</td>
<td>0.00175</td>
<td>4.533%</td>
</tr>
<tr>
<td>chlorides</td>
<td>0.00171</td>
<td>4.429%</td>
</tr>
</tbody>
</table>

Once the weights were created, the system will read the training dataset and the testing dataset for every experience in memory. In our experiment, there are 4798 training experiences loaded and 100 testing experiences loaded. As a result, two sets of SOEKS instances are created, as follows:

Prognoser prognosis = new Prognoser();
Vector<SOEKS> setOfTrainingExperience = prognosis
       .parseFromFile("./winequality-white.csv");
Vector<SOEKS> setOfTestingExperience = prognosis
       .parseFromFile("./wine-white-test1.csv");

After those two sets of experience are loaded in memory, the generated weights will be allocated to the related variables. A loop is used to assign collected weights to the variables of each experience of the training dataset. For example, for each training experience, trainExpVars.get(0) .weight(weight) is used to assigned weight to the variable in the SOEKS. And an equation, double sim = Math.abs(trainExpVars.similarSOV(testVariables)), is used to acquire similarities of comparing test and training experiences. The minimum value was output as prediction. This process is repeatedly executed until all the predictions are found. The predictions are stored as XML files shown below.
In the XML file, it has 12 variables of the SOEKS defined [12]. Each variable has a weight related. Therefore, this prediction is easy to be reused and transported in different systems. Additionally, it also has ability to be self-described according to the functionality of XML. Hence, knowledge will be expended and shared with different users.

Moreover, the DDNA is a representation structure that can learn or be trained by existing experience [12]. The quantity of sets of experience has a great impact on it. If the scope of existing experience is too short for the system to learn, the prediction will not be precise. However, the SOEKS is able to embed algorithm as function to enhance its ability. We tested this by introducing a linear regression function in our experiment to predict data when it is out of arrange of current experience.

When the data was loaded into the SOEKS, the system executes linear regression classifier to calculate the model which has been saved as the function of the SOEKS (see below).

\[
\text{quality} = 0.061 \times \text{fixed acidity} - 1.9584 \times \text{volatile acidity} + 0.089 \times \text{residual sugar} + 0.0028 \times \text{free sulfur dioxide} - 159.7935 \times \text{density} + 0.6875 \times \text{pH} + 0.665 \times \text{sulphates} + 0.1924 \times \text{alcohol} + 159.6391
\]

Prognoseter.getFunctionFactor(quality, queriedSOE);

Afterwards, each predicted object of the DDNA is produced based upon the linear regression function. The prediction has been successfully outputted by using SOE.getSetOfFunctions().get(0).value(). When choosing this predicted set of experience for another prediction, those with better fitness value have a higher probability of being selected. In this case, the queried object is dynamically and iteratively produced to fill up different needs.

### 3.1 Performance Analysis

In this experiment, those two sets of experience, 4998 records in total, were separated into two CSV files and loading them took less than a second. Afterwards, the system spends 23 seconds to assess the percentage of variables’ weights by the means of the weka module. Ultimately, the two groups of sets of experience consumed 6 seconds to evaluate 100 predictions from 4798 training experiences. The whole process spends in total 29 seconds. This experiment also proved that this new structure has several advantages. The SOEKS provided a semantic point of view over an XML approach in which it is able to deliver interesting benefits as its Ontology model allows inferring semantically new derived queries [10, 12]. It is commonly known that quality of predictions can be measured by using Mean Absolute Error (MAE) in statistics. In our experiment, we got a MAE score of 0.64 over 100 records. It presents a better performance compare with another good prediction approach, decision stump, with a MAE score of 0.6675. We suggested that DDNA is able to be used for highly preciseness and effective prediction. Figure 4 shows the predicted and observed absolute errors of training experience and queried instances between the decision stump and the DDNA approach. Both results fluctuated in scope of [-2, 3]. However, the prediction in DDNA has a smaller range of absolute error than in the decision stump. Hence, the DDNA indicates a highly precise performance of the SOEKS.
A difference from earlier works, including the two structures mentioned above, is our focus on discovering capabilities of prognosis, solution and knowledge macro processes in the SOEKS by integrating the RELIEF-F measure and linear regression into our structure. The experiment demonstrates an implied and user-friendly measure for predictions.

4 Conclusion and Future work

The decisional DNA based data mining technique is a suitable and comprehensive tool for knowledge discovery. The enhancement of using RELIEF-F measure helps the SOEKS to automate the process of knowledge extraction. The empirical experiment also proved that this structure is an effective and efficient solution for prediction. Furthermore, as SOEKS is expressed in terms of variables, functions, constraints and rules, it is possible to integrate additional elements to adapt it to various circumstances. Therefore, this structure can be efficiently and precisely used to explore knowledge from vast data sources. And, it assists users in making their data shareable, transportable and easy understandable. Meanwhile, the highly intelligent functionality will be the key for user’s information management.

This present work provides some important steps in forecasting direction, but it makes many assumptions that need to be studied in the future. First, the precision of weight estimation still has space to be improved. In addition, we still need to carry out more experiments to apply it into different domains. More extractions and inferences need to be experienced in order to assist organizations to make better decisions.

References


