Column Vectorizing Algorithms for Support Vector Machines

Chen Zhi Yuan, Dr. Dino Isa, Dr. Peter Blanchfield.

Abstract—**In this paper we present the vectorization method for support vector machines in a hybrid Data Mining and Case-Based Reasoning system which incorporates a vector model to help transfer textual information to numerical vector in order to make the real world information more adapted to the data mining engine. The main issue of implementing this approach is two algorithms; the discrete vectorization algorithm and continuous vectorization algorithm. According to the design of the hybrid system, the input information is the text table contains different kinds of columns which are stored in the SQL server. The basic idea of the vectorization algorithm is to derive X value from the original column value and where the vector value is unavailable; the algorithm builds a vector table based on the X value by using appropriate functions. Subsequently, the vector model is classified using a support vector machine and retrieved from the case based reasoning cycle using a self organizing map. The objective of the vectorization process is to merge the difference between data mining technology and artificial intelligence tools, so that we can integrate these two techniques in the proposed hybrid system. The advantage of using discrete algorithm is that each discrete features in the whole table was assigned a vector value in an easily expression calculation. While for the continuous features we choose a relatively complicated formula that is the Hyperbolic Tangent function to achieve the vector value. The formulas in these algorithms are quite basic but the impressive part is it also provides a reasonable balance between a satisfactory result and reasonable processing time. Furthermore, due to the modular structure of the algorithm it can be adapted easily for various applications.**

Index Terms—**Support Vector Machine, Data Mining, Artificial Intelligence, Case-Based Reasoning.**

I. Introduction

The problem faced by traditional database technology developer today is lack of intelligence support, while artificial intelligence techniques [1] were limited in their capacity to supply and maintain large amount of factual data. This paper provides a method to solve this problem.

From a database point of view, there was an urgent need to address the problems caused by the limited intelligent capabilities of database systems, in particular relational database systems. Such limitations implied the impossibility of developing, in a pure database context, certain facilities for reasoning, problem solving, and question answering. From an artificial intelligence point of view, it was necessary to transcend the era of the operating on numerical signals to achieve the real information management system able to deal with large amounts of textual data. Our approach was explicitly designed to support efficient vectorization techniques by providing multiple number resources with minimum inter-dependencies and irregular constraints, yet under strict artificial intelligence considerations. It features a table in a relational database through two types of vectorizing functions, supporting to the construction of the support vector machine.

The rest of this paper is organized as follows: Section 2 presents objectives and related techniques. Section 3 describes in detail the architecture of the hybrid system. Section 4 provides the procedure of vectorization. Section 5 explains the conducted experiments. The conclusion is discussed in section 6.

II. Objectives and Foundation

Our research group works on the designing of flexible and adaptable user oriented hybrid systems which aims to combine database technology and artificial intelligence techniques. The preprocessing procedure related to data vectorization step of a classification process, going from low level data mining processes [2] to high level artificial intelligence techniques. Many domain specific system such as user modeling systems [3] or artificial intelligence hybrid systems have been described in literature [4] [5] [6]. Even when the applied strategies are designed as generic as possible, the illustration given for the system are limited to the text document and do not develop any vectorizing algorithm to quantitate the input raw textual data set into numeric data set.

Actually, to the best of our knowledge, no such complete and generic vectorization process exists because of the necessity to have an excellent know-how in the implementation of a hybrid intelligent system. Many existed systems have been developed on the basis of using artificial intelligence techniques to provide semantic support to a database system, or database techniques to aid an artificial intelligence system to deal with large amounts of information. The key factors they concerned reside in the exploitation of the equivalence between database field and the knowledge representation system of artificial intelligence.

Manuscript received May 31, 2008.

Dr. Dino Isa is with the Engineering and Computer Science Faculty, the University of Nottingham Malaysia Campus (phone: 006-03-8924-8116; fax: 006-03-8924-8017; e-mail: dino.isa@nottingham.edu.my).

Dr. Peter Blanchfield is with the Computer Science School, the University of Nottingham UK Campus, (e-mail: pxb@Cs.Nott.AC.UK).

Chen Zhi Yuan is a PhD student under the Engineering and Computer Science Faculty, the University of Nottingham Malaysia Campus, (e-mail: eyx6czy@nottingham.edu.my).

In our hybrid system, vector is the unique representation of data considering the system consistency. On the other hand, for both data mining process and case-based reasoning cycle [7], vectorization and consistency are crucial. The role of vectorization is to convert text table which stored in SQL server, into numerical vector form. Traditional vectorization method concentrates on image object into a raster vector or raw line fragments. While we focus on these table column features and describe how they can be vectorized by applied automatically approach using two kinds of vectorization functions.

In order to describe the foundation of the vectorization, the framework of our hybrid system is simply described in the following section.

III. Hybrid system architecture overview

The concepts of this project are as follows:

1) To develop a hybrid data mining and case-based reasoning user modeling system
2) To combine data mining technology and artificial intelligence pattern classifiers as a means to construct a Knowledge Base and to link this to the case-based reasoning cycle in order to provide domain specific user relevant information to the user in a timely manner.
3) To use the self organizing map [8] in the CBR cycle in order to retrieve the most relevant information for the user from the knowledge base.

Based on these concepts the architecture has been designed which is illustrated in Figure 1. The hybrid system contains five main components:

1) Individual models, comparable to the blackboard containing the user information from the real world.
2) Domain database integrated the preselected domain information [9].
3) A data mining engine which classified both user class and domain information vectors.
4) A knowledge base, containing the representation of classified user information and combined with interested domain knowledge.
5) A problem-solving life-cycle called case-based reasoning cycle, assisting in retrieve reuse revise and retain the knowledge base.

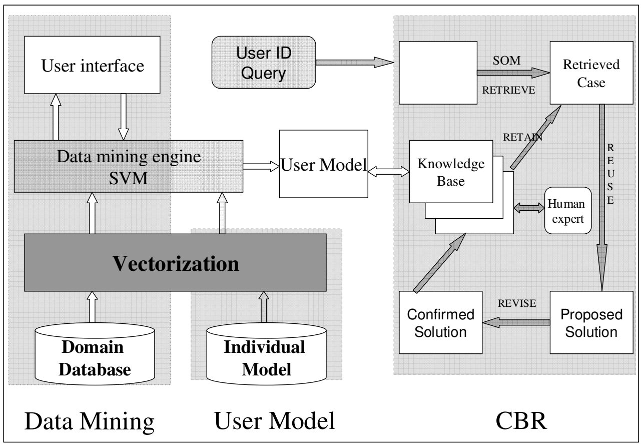


Figure 1. The architecture of the system

IV. Vectorization

As can be seen from the hybrid system architecture, in order to classify individual models and domain information into user model the support vector machine are applied. Individual models are user information which took table format and stored in the SQL server. Domain information in the database is also sorts of tables which stored the preselected user-preferred knowledge. The support vector machine [10] [11] is one of AI techniques which serve as classifier in the system. The main idea of a support vector machine is to construct a hyper plane as the decision surfaces in such a way that the margin of separation between positive and negative features is maximized. The vectorization step is the data preprocessing for the support vector machine which provides the numeric feature vector.

A. Feature type

For vectorization task to be as accurate as possible we predefined two type table columns or we called feature type; discrete columns (feature) and continuous columns (feature).

Discrete feature contains discrete values, in that the data represents a finite, counted number of categories. The values in a discrete attribute column do not imply ordered data, even if the values are numeric; the distinct character is values are clearly separated. Telephone area code is a good example of discrete data that is numeric.

Continuous feature contains values that represent a continuous set of numeric and measurement data, and it is possible for the data to contain an infinite number of fractional values. An income column is an example of a continuous column.

The numeric value is not the vital factor to determine the feature type, but if the value is a word then it must be a discrete feature.

B. Vectorization Algorithm

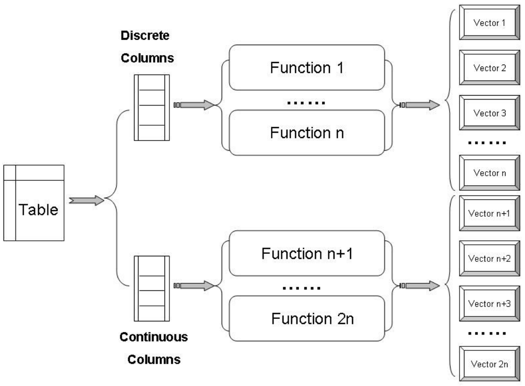


Figure 2. The schema of the vectorization algorithm

From the technology point of view, vectorization is an approach modeling relationships between the data set and the vectorizing variable. We provide a more flexible approach by allowing some of the features (columns) to be independent and some of the features to be interdependent. Constructing two parallel algorithms to avoid time consuming and save a large amount of effort.

The schema of the algorithm is specified in Figure 2 which derives the numeric vector by implementing different functions. The schema is not exhaustive and can evolve with new data, according to user need.

Furthermore, once the type of the column has been determined, adding a new record is quite straightforward. These functions are also well suited to dealing with incomplete data. Instances with missing attributes can be handled by summing or integrating the values of other attribute.

We represent each column as a data point in a dimensional space, where Z is the total number of attributes (columns). The algorithm computes the vectorizing value (or representation value) between each feature which was denoted by abscissa axis and the vector denoted by y-axis, and all the feature values determine its own vectorizing values. Once the vectorizing value list is obtained, the vector model will be classified based on the implementation of support vector machine so that the core of the hybrid system the knowledge base will be constructed completely.

The detailed vectorization algorithms are described in the Table 1 and Table 2 according to discrete columns and continuous columns.

Table 1. The discrete column vectorization algorithm

```
1: Let V be the representation of Vectors, D be the whole set of the vector model and d be the set of discrete columns.
2: FOR each data point Z DO
3:   Select $Z_d$, the discrete features of all data point,
4:   Compute $V_d = (V_{dx}, V_{dy})$, the corresponding value between Z and every vector, $(V_{dx}, V_{dy}) \in$ D.
5:   $V_{dx} = n_d$, $V_{dy} = {}^{1}\!/_{n} \times n_d$; $V_{dy} \in [0,1]$.
6: END FOR
```

Table 2. The continuous column vectorization algorithm

```
1: Let V be the representation of Vectors, D be the set of vector model and c be the set of continuous columns.
2: FOR each data point Z DO
3:   Select $Z_c$, the continuous features of all data point,
4:   Compute $V_c = (V_{cx}', V_{cy}')$, the corresponding value between Z and every vector, $(V_{cx}', V_{cy}') \in$ D.
5:   $V_{cx}' = (V_{cx} - AvgV_{cx}) / MaxV_{cx}$,
     $V_{cy}' = {}^{e^x - e^{-x}}\!/_{e^x + e^{-x}}$, $V_{cy}' \in [-1,+1]$.
6: END FOR
```

The key computation of these two algorithms is the vectorization value formula given in step 5 of the both table.

Formula 1:

$$V_{dx} = n_d$$

$$V_{dy} = {}^{1}\!/_{n} \times n_d$$

Formula 2:

$$V_{cx}' = (V_{cx} - AvgV_{cx}) \Big/ MaxV_{cx}$$

$$V_{cy}' = {}^{e^x - e^{-x}}\Big/_{e^x + e^{-x}}, \; V_{cy}' \in [-1,+1].$$

In Formula 1, n is the weight parameter associated with the discrete columns which is the sum of value type. V_{dy} is a combination of the unit value ($1/n$) multiply the sequence of the current value type (n_d). This is a regression-like expression [12]. Regression is used to make predictions for numerical targets. By far the most widely used approach for numerical prediction is regression, a statistical methodology that was developed by Sir Frances Galeton [13]. Generally speaking Regression analysis methods include Linear Regression, Nonlinear Regression. Linear Regression is widely used, owing largely to its simplicity. By applying transformations to the variables, we can convert the nonlinear model (text table column information) into a linear one according to the requirement of the support vector machine.

In order to get the negative X value and at the same time keep the same distance among original X value, in Formula 2 we minus average value to all x value and then get the proportion compare with the maximum original X value, after that get the new X value and by means of Hyperbolic Tangent function [14] to map these new value into (-1, +1) scale.

In order to explain these algorithms clearly, we show the experiment procedure in the following section.

V. Experiments

The vectorization algorithm was tested on the census-income data set extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau. The data contains 41 demographic and employment related variables. In order to explain how to apply our approach clearly, we choose 8 discrete columns and 8 continuous columns which can be found in table 3 to explain the implementation in details. In Table 4 we list the n value of the discrete columns. For example the worker class n value, because there are 9 kinds of worker class, so n is equal to 9. Parts of the experiment results implemented the proposed algorithms which contain 27 records are shown in figure 3.

The input for the algorithm was given 8 discrete features and 8 features and asked to give the vectorized value as output. The discrete attributes were decomposed into n equidistance, which yielded corresponded vector value scaling to the range of (0, 1). For the continuous attribute, firstly the raw attribute value was transferred into the whole x-axis, so that the new x value contain the negative value and by using Hyperbolic Tangent function the vector value was calculated. The Hyperbolic Tangent function make sure the vector value to be projected into the (-1, 1) scale which is required by support vector machine.

Table 3. Parameters for experiments

Discrete columns	Continuous columns
class of worker*	age*
education*	wage per hour
marital stat	capital gains
sex	capital losses
reason for unemployment	dividends from stocks
family members under 18	person for employer
live in this house 1 year ago	weeks worked in year
veterans benefits	instance weight*

Table 4. The discrete column n value

Discrete columns	n Value
class of worker	9
education	17
marital stat	7
sex	2
reason for unemployment	6
family members under 18	5
live in this house 1 year ago	3
Veterans benefits	3

The recommend vector value range is (0, 1) or (-1, 1) for support vector machine [15]. One reason for this is to avoid vector value in great numeric ranges dominates those in smaller numeric ranges. Another reason is to avoid the numerical difficulties during the calculation. Because kernel values usually depends on the inner products of feature vectors. For example the linear kernel and the polynomial kernel, large vector values may cause numerical problems [16].

Another reason why we proposed two kinds of algorithm to vectorize discrete columns and continuous columns is to preserve the character of the column for the sake of the later analysis.

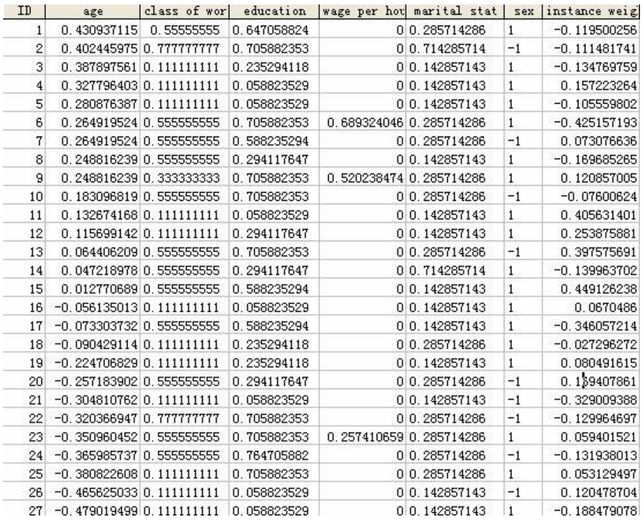
All the experiment results was created on PC computer, CPU Intel(R) Core(TM) Duo CPU T2250 @ 1.73GHz 4.6 2.3, 2GB RAM DDR2 667 MHz, with WinXP. Program was compiled with NetBeans 6.0.

VI. Conclusions

The proposed hybrid Data Mining and Case-Based Reasoning User Modeling system is a multi purpose platform and is characterized by three major processes. The vectorization processing unit communicate through the raw data set the SQL table and the output is the numeric vector, such an approach avoid the data inconsistency usually met in classifying documents chain when implement artificial intelligence tools.

In this paper we built vectorization model by applying two algorithms: The discrete vectorization algorithm and continuous vectorization algorithm. The advantage of using discrete algorithm is that each record in the whole table was assigned a vector value in an easily expression calculation. While for the continuous column we choose a relatively complicated formula that is the Hyperbolic Tangent function to achieve the vector value.

In designing the algorithm, the key consideration is to bring up easy scientific numerical transformation. Therefore, the formulas in the algorithm are quite basic but the impressive part is it also provides a reasonable balance between a satisfactory result and reasonable processing time. Secondly due to the modular structure of the algorithm it can be adapted easily for application. The results of the algorithm in the experiments labeled clean and the vector points generated by our algorithm have a standard coverage (0, 1) and (-1, 1) which is useful in fulfilling the classification task by means of support vector machine for the hybrid system.



ID	age	class of wor	education	wage per hou	marital stat	sex	instance weig
1	0.430937115	0.55555555	0.647058824	0	0.285714286	1	-0.119500256
2	0.402445975	0.777777777	0.705882353	0	0.714285714	-1	-0.111481741
3	0.387897561	0.111111111	0.235294118	0	0.142857143	1	-0.134769759
4	0.327796403	0.111111111	0.058823529	0	0.142857143	1	0.157223264
5	0.280876387	0.111111111	0.058823529	0	0.142857143	1	-0.105559802
6	0.264919524	0.555555555	0.705882353	0.689324046	0.285714286	1	-0.425157193
7	0.264919524	0.555555555	0.588235294	0	0.285714286	-1	0.073076636
8	0.248816239	0.555555555	0.294117647	0	0.142857143	1	-0.169685265
9	0.248816239	0.333333333	0.705882353	0.520238474	0.285714286	1	0.120857005
10	0.183096819	0.555555555	0.705882353	0	0.285714286	-1	-0.07600624
11	0.132674168	0.111111111	0.058823529	0	0.142857143	1	0.405631401
12	0.115699142	0.111111111	0.294117647	0	0.142857143	1	0.253875881
13	0.064406209	0.555555555	0.705882353	0	0.285714286	-1	0.397575691
14	0.047218978	0.555555555	0.294117647	0	0.714285714	1	-0.139963702
15	0.012770689	0.555555555	0.588235294	0	0.142857143	1	0.449126238
16	-0.056135013	0.111111111	0.058823529	0	0.142857143	1	0.0670486
17	-0.073303732	0.555555555	0.588235294	0	0.142857143	1	-0.346057214
18	-0.090429114	0.111111111	0.235294118	0	0.285714286	1	-0.027296272
19	-0.224706829	0.111111111	0.235294118	0	0.142857143	1	0.080491615
20	-0.257183902	0.555555555	0.294117647	0	0.285714286	-1	0.139407861
21	-0.304810762	0.111111111	0.058823529	0	0.142857143	-1	-0.329009388
22	-0.320366947	0.777777777	0.705882353	0	0.285714286	-1	-0.129964697
23	-0.350960452	0.555555555	0.705882353	0.257410659	0.285714286	1	0.059401521
24	-0.365985737	0.555555555	0.764705882	0	0.285714286	-1	-0.131938013
25	-0.380822608	0.111111111	0.705882353	0	0.285714286	1	0.053129497
26	-0.465625033	0.111111111	0.058823529	0	0.142857143	-1	0.120478704
27	-0.479019499	0.111111111	0.058823529	0	0.142857143	1	-0.188479078

Figure 3. Part of the experiment results

References

[1] Stuart J. Russell and Peter Norvig, *Artificial Intelligence A Modern Approach*, Prentice-Hall International Inc, 1995.

[2] Usama Fayyad, G. Paitetsky-Shapiro, and Padhrais Smith, “knowledge discovery and data mining: Towards a unifying framework”, *proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 82-22.

[3] Vassileva, J., "A practical architecture for user modeling in a hypermedia-based information system", *Proceedings of Fourth International Conference on User Modeling*, Hyannis, MA, August 1994, pp 15-19.

[4] Vadim I. Chepegin, Lora Aroyo, Paul De Bra, “Ontology-driven User Modeling for Modular User Adaptive Systems”, *LWA, 2004*, pp.17-19.

[5] Watson, I, Applying Case-Based Reasoning: Techniques for Enterprise Systems, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.

[6] Watson, I, *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.

[7] Aamodt, A. and Plaza, E., “Case-based reasoning: foundational issues, Methodological variations, and system approaches”, *AI communications*, 7(1), 1994, pp. 39-59.

[8] F. Murtagh. Interpreting the Kohonen, “self-organizing map using contiguity-constrained clustering”, *Pattern Recognition Letters*, 1995, pp. 399–408.

[9] Hjorland, B. & Albrechtsen, H., “Toward A New Horizon in Information Science: Domain Analysis”, *Journal of the American Society for Information Science*, 1995, 46(6), 400-425.

[10] Osuna, E., “Support Vector Machines: Training and Applications”, *Ph.D thesis*, Operations Research Center, MIT, 1998.

[11] Vapink, V.N., *Statistical Learning Theory*, New York:Wiley.

[12] Lindley, D.V., "Regression and correlation analysis," *New Palgrave: A Dictionary of Economics*, v. 4, 1987, pp. 120-23.

[13] Francis Galeton,“Typical laws of heredity", *Nature 15*,1877, pp. 492-495, 512-514, 532-533.

[14] M.A. Abdou and A.A. Soliman, “Modified extended tanh-function method and its application on nonlinear physical equations”, *Physics Letters A*, Volume 353, Issue 6, 15 May 2006, pp. 487-492

[15] E. Osuna, R. Freund, and F. Girosi, “Improved training algorithm for support vector machine”, *IEEE Neural Networks in Signal Processing 97*, 1997.

[16] Cortes, C. and V. Vapnik, “Support-vector network”, *Machine Learning*, 1995, pp. 273–297.